Technical Requirements Document for OLCF-6

Status: Draft v1.0

Change Log

Revision History		
Version	**Date**	**Description of Changes**
1.0	September 2023	Initial version

TABLE OF CONTENTS

1 Introduction 1
1.1 Program Overview and Mission Needs 1
1.2 Guidance for Offerors 4
1.3 Requirements Definitions 5
2 High Level System Requirements 6
2.1 System Architecture Requirements 6
2.2 Requested Options 9
3 Benchmarks 13
3.1 Benchmark Availability 13
3.2 Performance Measurements - Figures of Merit 13
3.3 Benchmarking Procedures 14
3.4 Allowed Modifications for Benchmark Codes 14
3.5 Benchmark Summary 15
4 I/O Subsystem 16
4.1 Parallel File System 16
4.2 AI-Optimized Storage 20
5 High Performance Interconnect 24
5.1 Network Requirements 24
5.2 Messaging Software Requirements 26
6 System Management 27
6.1 System Management Architecture 27
6.2 Workload Management 27
6.3 Operating System 28
6.4 Platform Management 29
6.5 System Software Deployment 31
6.6 System Networks 31
6.7 Data Collection and Monitoring 32
7 User Environment 34
7.1 User Environment Packaging 34
7.2 User Environment Libraries 34
7.3 AI/ML Support 35
7.4 Workflow Orchestration 36

7.5 Application Driven Runtime Power & Energy Management for Energy Efficiency 36
7.6 Baseline Languages 36
7.7 Programming Models 37
8 Tools 39
8.1 Hardware Counters (HWC) 39
8.2 Infrastructure 41
8.3 Debugging 41
8.4 Performance 43
8.5 Code Analysis 45
9 Security 47
9.1 Security Model 47
9.2 Identity Management 47
9.3 Vulnerability Disclosure & Remediation 47
9.4 Secure Software Development Framework 47
9.5 Executive Order Compliance 47
9.6 Distributed Workflow Security 47
9.7 Offeror Privileged Access 48
10 Facilities 49
10.1 Facility Requirements (Applies to On Premise, ORNL) 49
10.2 ORNL Facility – Electrical Distribution 52
10.3 ORNL Facility – Cooling Distribution 53
10.4 Power and Cooling Requirements 55
10.5 Minimal electrical and mechanical connections 56
10.6 Tolerance of Power Quality Variation 56
10.7 Power Factor and Harmonic Current Requirements 56
10.8 Electrical Safety and Power Standards 57
10.9 Mechanical 57
11 Project Management 61
11.1 Project Management 61
11.2 Project Management Documents 61
11.3 Payment Milestones 61
11.4 Technical Decision Point 61
11.5 Risk and Risk Management 62

11.6 Quality Assurance ... 62
11.7 High-Level Technical Collaboration ... 62
11.8 Working Groups ... 62
11.9 Planning Kick-off ... 63
11.10 Quarterly Reviews ... 63
11.11 Acceptance ... 63
12 Non-Recurring Engineering (NRE) ... 64
12.1 Center of Excellence ... 64
13 Maintenance and Support ... 65
13.1 24/7 Maintenance ... 65
13.2 Analyst Support ... 65
13.3 Ensuring Node Availability ... 65
13.4 Support Tracking ... 65
13.5 Non-volatile media retention ... 65
14 Glossary ... 67
14.1 AI-optimized Storage (AOS) ... 67
14.2 Cold Boot ... 67
14.3 FPP ... 67
14.4 Hero Random ... 67
14.5 Hero Sequential ... 67
14.6 JACNM ... 67
14.7 Job-shared Storage ... 67
14.8 Job-specific Storage ... 67
14.9 Node-shared Storage ... 67
14.10 PFS ... 68
14.11 System Aggregate HBM memory (SAHM) ... 68
14.12 Scheduled Availability Level ... 68
14.13 SSF ... 68
14.14 SSU ... 68
14.15 SSC ... 68
14.16 System Outage ... 68
14.17 Warm Boot ... 68
14.18 Hardware ... 68

14.19 Software 72
15 Appendix A: Workflows Context 75
16 Appendix B: I/O Use Cases 77
16.1 I/O 77

TABLE OF FIGURES

Figure 1: Workflows overlaid on supercomputers 75

Figure 2: Integrated workflow enablers 76

1 INTRODUCTION

This document contains the technical requirements for the OLCF-6 post-exascale High Performance Computing (HPC) capability to be delivered in 2027 for Oak Ridge National Laboratory (ORNL) and the Oak Ridge Leadership Computing Facility (OLCF). This capability is required to meet the mission needs of the Advanced Scientific Computing Research (ASCR) Program within the Department of Energy (DOE) Office of Science (SC). ORNL is managed by UT-Battelle, LLC, referred to in this document as Company.

Each proposed solution in response to this document should clearly describe the role of the major lower-tier technology subcontractor(s) (e.g., CPUs, accelerators, interconnect) and third-party software partners (e.g., workload manager, performance tools) and the value added that these subcontractor(s) provide(s).

This capability has maximum funding limits over its lifetime, to include all design and development, site preparation, maintenance, support, and operations. Total cost of ownership (e.g., system cost, Non-Recurring Engineering (NRE) costs, leasing costs, and power and cooling costs) will be evaluated in system selection. Offerors that do not respond with a specific configuration or with complete pricing may receive a negative evaluation.

This draft requirements document describes specific technical requirements related to both the hardware and software capabilities of the desired system as well as application benchmark requirements. Additional information on proposal preparation will be provided in the Proposal Evaluation and Proposal Preparation Instructions (PEPPI).

The delivery timing provided above represents the current outlook and alignment of programmatic requirements and funding. Company reserves the right to revise the above schedule based upon their and/or DOE's needs.

1.1 PROGRAM OVERVIEW AND MISSION NEEDS

1.1.1 Office of Science

The DOE Office of Science (SC) is the lead Federal agency supporting fundamental scientific research for energy and the nation's largest supporter of basic research in the physical sciences. The SC portfolio has two principal thrusts: direct support of scientific research and direct support of the development, construction, and operation of unique, open-access scientific user facilities. These activities have wide-reaching impact. SC supports research in all 50 States and the District of Columbia, at DOE laboratories, and at more than 300 universities and institutions of higher learning nationwide. The SC user facilities provide the Nation's researchers with state-of-the-art capabilities that are unmatched anywhere in the world.

Within SC, the mission of the Advanced Scientific Computing Research (ASCR) program is to discover, to develop, and to deploy computational and networking capabilities to analyze, to model, to simulate, and to predict complex phenomena important to the DOE. A particular challenge of this program is fulfilling the science potential of emerging computing systems and other novel computing architectures, which will require numerous significant modifications to today's tools and techniques to deliver on the promise of science in the exascale era.

1.1.2 Mission Needs

Today, high-performance computing (HPC) is used extensively in the advancement of DOE missions in science and engineering. By calendar year (CY) 2028, the Frontier supercomputer at OLCF will be nearing its end of life. To continue to meet DOE mission requirements and national priorities, the system will need to be updated and reimagined to address needs, drive innovation, and expand capabilities to sustain U.S. leadership as part of an advanced computing ecosystem.

DOE established the Leadership Computing Facility (LCF) in response to the High-End Computing Revitalization Act of 2004. The Act stated that the Secretary of Energy, acting through the Office of Science, shall (1) establish and operate Leadership Systems Facilities and (2) provide access to Leadership Systems Facilities on a competitive, merit-reviewed basis to researchers in U.S. industry, institutions of higher education, national laboratories, and other Federal agencies.

Demand for leadership-scale computing continues to grow and expand into new and challenging dimensions that require advanced capabilities. The Leadership Computing Facility's (LCF's) flagship allocation program, INCITE, is consistently oversubscribed for compute and data requests, highlighting the persistent need for leadership-class resources. The capability gap has also been identified by the U.S. science community and described in a series of recent AI for Science and multi-agency HPC workshops. LCF resources are available to users covering a breadth of science domains, each of which have a need for increased computational and data science capabilities (i.e., traditional modeling and simulation, data analytics, and AI).

Over the next few years, data generation rates at experimental and observational facilities will increase by orders of magnitude due to advances in detector technology, deployment of edge sensors, and other factors. At the same time, higher resolution simulation science will be continuing to generate data sets growing at similar rates. The post-exascale generation of LCF resources must be interoperable with an Integrated Research Infrastructure (IRI) to provide researchers with the ability to meld experimental control and analysis of large-scale experimental/observational datasets with high-resolution simulations and/or AI technologies.

With world demand for AI, data analytics and computing at all scales growing exponentially, energy utilization is both a constraint and mission driver for the LCF and for an IRI. DOE has been the global leader in driving major gains in energy efficiency, in partnership with researchers and vendors. For the past few decades, DOE's commitment to leadership-scale advances in supercomputing has fueled a robust partnership with the computer industry in software and hardware. This partnership has led to new technological developments in energy efficient HPC architectures, including AI accelerators, memory technologies, high-speed interconnects, systems software, and other innovations. There is a mission need to continue to make progress and lead in dramatically improving energy efficiency across the ecosystem.

To meet the Mission Need, the desired resource must:

- provide a significant increase in leadership computational and data science capabilities over the Frontier baseline;
- support strong and weak application scaling to the full system size;
- be expandable with new, novel architectures to provide enhanced capabilities;
- interoperate with a leadership scale integrated research infrastructure (IRI), defined as the ability to interface with and support an integrated research infrastructure connecting DOE experimental user facilities and other LCF infrastructure;
- continue to make progress and lead in dramatically improving energy efficiency across the ecosystem;
- be operational before the end of Frontier's service life;

- operate within the OLCF's utility and operations budget; and
- provide a productive programming environment to users, defined as a diverse, supported, and state-of-the-art set of compilers, debuggers, libraries, and tools.

1.1.3 Leadership Workloads

Modeling and Simulation (ModSim)

OLCF systems have traditionally supported full-system modeling and simulation workloads. This workload comprises bulk-synchronous, distributed memory applications using either strong or weak scaling. The smallest leadership job uses 20% of the nodes and applications are expected to scale up to the full system or, at least, the largest power-of-two number of processes that fits within the system. The interconnect needs to provide sufficient bandwidth for full system jobs. OLCF optimizes the parallel file system to handle large, streaming writes. Most applications (up to 90% of applications) write less than 15% of the aggregate accelerator memory per hour. On average, the system adds around 1.5x total accelerator memory to the parallel file system each day. The file size distribution is bimodal, with the vast majority of the files being small (≤32KiB) but the vast majority of the capacity is consumed by large files (≥1GiB).

While traditionally all double precision, scientists are exploring the feasibility of exploiting lower-precision data types, either natively or using iterative refinement to regain full precision, to accelerate their ModSim workloads. This is in addition to the AI workload below.

Artificial Intelligence (AI)

AI is a growing workload on OLCF systems. OLCF-6 is expected to be at the forefront in supporting domain scientists and application developers as they explore and integrate transformational AI technologies to accelerate discoveries in science, energy and security problems of national importance. We envision a wide spectrum of use cases ranging from inverse design and control of complex systems such as power grids and nuclear reactors, to generative AI and foundational models that integrate text and images that are often unstructured, high-resolution, and from multi-modal data sources. Executing AI-empowered computing campaigns and workflows will place new demands on the system architecture, possibly requiring more interconnect bandwidth and an optimized storage layer that can handle very high rates of I/O operations (IOPs) focused on random reads.

Integrated Research Infrastructure (IRI)

In addition to operating world-class computational facilities, DOE operates many other experimental facilities. Other Federal agencies also operate numerous observational facilities. DOE is leading the way to develop the ability to interconnect research facilities to decrease the time to scientific insight. OLCF-6 will need to interoperate with this new paradigm. The requirements below include OLCF's estimates for capabilities needed to support this advanced computing ecosystem, e.g., off-system bandwidth, storage capacity, and user environment.

The “Workflows Context” appendix to this document offers additional context to inform offeror responses.

1.2 GUIDANCE FOR OFFERORS

The Company has structured these requirements with the objective of allowing Offerors to propose a wide range of solutions, including an on-premises system with or without storage; an off-premises system; and/or an upgrade to the existing Frontier system. The Company also invites standalone proposals for the Parallel File System (PFS) and AI-Optimized Storage (AOS) options alone.

Specific guidance for preparing responses to each of these scenarios is given below. Offerors may provide one or more proposals for the scenarios.

All Offerors (except storage-only proposals) must respond to the Mandatory Requirements in Section 2 (2.1.1.1 System Description and 2.1.2.1 High-Level Software Model) at a minimum to be considered responsive. All other requirements are graded TR-1/2/3 to indicate the importance to the Company. Offerors should respond to as many requirements and options as are applicable to the proposal and are practicable but should not feel obligated to respond to all.

In Section 13 of this document, the Company requires maintenance for the first five years post-acceptance with options for maintenance in years 6 and 7. If offeror is bidding a service for which maintenance is included in the service price, describe the maintenance as requested in the requirements and indicate in the separate Price Proposal that the cost is included. For extended maintenance in years 6 and 7, describe the maintenance in this document and indicate in the Price Proposal what the service price would be in those years.

1.2.1 Guidance for On-Premises Proposals

ORNL has two data center location options of similar size and power delivery capability (30MW). The requirements of Section 10: Facilities provide information on both. Your response should indicate if your solution can fit in both or if it is designed to work in one specifically.

For on-premises infrastructure-as-a-service (IAAS) proposals, indicate as part of the High Level Software Model response the level of service that you are proposing and how you will provide software (e.g., base operating system, programming environment, workload manager, storage) to the Company to deploy on the system.

1.2.2 Guidance for Off-Premises Proposals

The Company intends for the language in these Technical Requirements to be inclusive of off-premises system proposals. If proposing an off-premises solution, the requirements of Section 10: Facilities do not apply and require no response.

The Company expects as much detail in the System Description regarding processors, nodes, interconnect, and storage as would be provided in an On-Premises proposal. As part of the High Level Software Model response, indicate the level of service that you are proposing and how you will provide software (e.g., base operating system, programming environment, workload manager, storage) to the Company to deploy on the system.

Any proposals for data centers outside of the state of Tennessee will require the Company to pay 9.75% sales tax on top of the system service price. This additional cost will be included in Total Cost of Ownership estimates during proposal evaluation.

1.2.3 Guidance for Upgrading Frontier Proposals

The Company assumes that any upgrade will be in-place, reusing as much of the Frontier infrastructure as possible. Storage will likely need to be upgraded as well.

1.2.4 Guidance for Storage Only Proposals

If the Offeror is only proposing storage, the Offeror needs to complete Sections 4: I/O Subsystem and Sections 10-13. The proposal should provide solutions for both options: Parallel File System (PFS) and AI-Optimized Storage (AOS). The proposal should also include options for extended maintenance for years 6 and 7, increasing and decreasing capacity for all media, and scaling bandwidth as well as any other options that the Offeror believes make sense.

Some of the storage requirements are written as a multiple of compute system memory capacity. A storage-only proposal should assume 8 PiB for this value. The proposal's options should allow the Company to adjust the storage systems up or down once the final system memory capacity is known.

Within the proposal, the Offeror should indicate which interconnect technologies are currently supported and are planned to be supported in this timeframe. The proposal's hardware and software description in 4.1.1 and 4.2.1 should include what software needs to be installed on clients (i.e., compute nodes). The proposal should indicate which Enterprise Linux distributions are supported.

The storage Offeror's proposal should define solutions for both data center options described in Section 10: Facilities.

1.3 REQUIREMENTS DEFINITIONS

Requirements in this document have priority designations, which are defined as follows:

- Mandatory Requirements (MR) are performance features that are essential to the Company's requirements, and an Offeror must satisfactorily propose all Mandatory Requirements in order to have its proposal considered responsive.
- Target Requirements (TR-1, TR-2, or TR-3) are features, components, performance characteristics, or other properties that are important to the Company, but that will not result in a nonresponsive determination if omitted from a proposal. Target Requirements are prioritized by dash number. TR-1 is most desirable to the Company, while TR-2 is more desirable than TR-3.
- Target Option Requirements (TO) are features, components, performance characteristics, or upgrades that are important to the Company, but that will not result in a nonresponsive determination if omitted from a proposal. Target Options add value to a proposal. Target Option responses will be considered as part of the proposal evaluation process; however, the Company may or may not elect to include Target Options in the resulting subcontract(s). Each proposed TO should appear as a separately identifiable item in an Offeror's proposal response.

The aggregate of MRs and TR-1s form a baseline system. TR-2s are goals that boost a baseline system, taken together as an aggregate of MRs, TR-1s and TR-2s, into a moderately useful system. TR-3s are stretch goals that boost a moderately useful system, taken together as an aggregate of MRs, TR-1s, TR-2s and TR-3s, into a highly useful system. Therefore, the ideal OLCF-6 system will meet or exceed all MRs, TR-1s, TR-2s and TR-3s. Target Requirement responses will be considered as part of the proposal evaluation process.

TOs provide alternative features, components, performance characteristics or system sizes that may be considered for technical and/or budgetary reasons. Target Options may also affect the Company's perspective of the ideal OLCF-6 system, depending on future budget considerations.

2 HIGH LEVEL SYSTEM REQUIREMENTS

2.1 SYSTEM ARCHITECTURE REQUIREMENTS

2.1.1 Hardware

2.1.1.1 System Description

The Offeror shall provide a detailed, architectural description of the proposed compute system including text, diagrams, tables, etc. as needed. Provide descriptions of each processor type, node type (which and how many processors, connectivity between processors, memory, NICs, node-local storage, if any), blade/chassis type, rack/cabinet, and interconnect. The offeror should include quantities and define any minimum scalable unit sizing to maintain optimal performance and productivity across the system.

- **Component architecture** – details of all processor(s), memory technologies, storage technologies, network interconnect(s) and any other applicable components.
- **Compute node architecture(s)** – details of how components are combined into the node architecture(s). Details should include bandwidth and latency specifications (or projections) between components. Details should be provided for each compute node type.
- **Board and/or blade architecture(s)** – details of how the node architecture(s) is integrated at the board and/or blade level. Details should include all inter-node and inter-board/blade communication paths and any additional board/blade level components.
- **Rack and/or cabinet architecture(s)** – details of how board and/or blades are organized and integrated into racks and/or cabinets. Details should include all inter rack/cabinet communication paths and any additional rack/cabinet level components.
- **Interconnect** - details of the system's high speed network topology and connectivity across all system components (compute nodes, login nodes, management system).
- **System architecture** – details of how rack or cabinets are combined to produce system architecture, including the high-speed interconnects and network topologies (if multiple) and storage systems.
- **Management node(s)** - details of hardware to support management and services to operate the OLCF-6 system. Describe the management node types (e.g., admin, leader, Slurm/resource manager, fabric manager). Multiple node types may be needed to optimize for different uses. Management nodes should be accessible even if the compute system is unavailable.
- **Front-end Environment Node(s) (FEN)** - details of hardware to support user access and potentially user-driven workflow activities. A pool of FENs will be needed to address the requirements described in Section 7. The front-end environment will collectively support 50 interactive users, 5 data analysis front-end applications, 500 batch jobs, and 20 simultaneous compilations of software of equivalent complexity to the latest GNU Compiler Suite. FEN should be accessible even if the compute system is unavailable. The "Workflows Context" appendix to this document offers additional context to inform offeror responses.

The I/O subsystem description should be in Section 4.

Priority: MR

2.1.1.2 System Performance

The Offeror will provide the system-level performance FOMs as defined in the Benchmarking section. The amount of FOM increase will contribute significantly to the total best-value assessment.

Priority: TR-1

2.1.1.3 High-Bandwidth Memory Bandwidth

The Offeror will describe the aggregate HBM bandwidth within the system. Provide both peak and estimated stream performance. It is highly desirable to have more HBM bandwidth than Frontier has (i.e., greater than 105 TiB/s measured, not peak).

Priority: TR-1

2.1.1.4 High-Bandwidth Memory Capacity

The Offeror will describe the aggregate HBM capacity within the system. It is highly desirable to have more HBM capacity than Frontier has (i.e., greater than 4.6 PiB of HBM2e).

Priority: TR-1

2.1.1.5 Compute Processor Performance

For each processor type in a compute node (e.g., CPU, GPU), provide estimated GEMM performance for each data type (e.g., FP64, FP32, FP16, BF16, INT8, INT4) in each supported data path (e.g., scalar, vector, matrix/tensor).

If a floating point data type is not IEEE 754 standard, describe the format used and denormal handling.

Priority: TR-1

2.1.1.6 Compute Processor Energy Efficiency

It is highly desirable to operate the system at the optimal point on the performance/power curve during a computationally intense phase such as a GEMM. Describe features of user-programmable processors (e.g., compute node, NIC, switch) that allow the user to control power and/or frequency.

Priority: TR-1

2.1.1.7 Projected System Power

The Offeror should describe the peak power requirement for the system and an estimate of the sustained system power when running a stress workload such as HPL.

Priority: TR-3

2.1.1.8 Maximum Process Count

Describe the expected number of processes per node and for a full system job (i.e., number of nodes times number of processes per node).

It is highly desirable to support a power-of-two processes per node.

Priority: TR-1

2.1.1.9 Reliability

Describe the projected MTBF, MTBAI/MTTAI, and MTTR.

OLCF allows Leadership Jobs (i.e., using 20% or more of the nodes) to run for 24 hours. It is highly desirable for the jobs to run without disruption for 24 hours. Each application has an I/O phase at the end of a well-defined compute phase that may be hourly or more. It is undesirable to require applications to add defensive checkpoint I/O phases in the middle of a compute phase.

Priority: TR-1

2.1.1.10 Serviceability

Describe the serviceability of the system. Describe the ability of hardware to self-report issues.

The system should have sufficient additional nodes installed and online to ensure that the node count necessary to meet the System Performance (Section 3 Benchmarks) are available.

Priority: TR-2

2.1.1.11 Accommodating Protected Information Workloads

Describe the ability of the system to handle Protected Information (e.g., HIPAA, ITAR) that requires encryption at rest, encryption in flight, unmounting of the default file systems, mounting of a protected file system, and wiping any local storage, if included.

Priority: TR-2

2.1.2 Software

2.1.2.1 High Level Software Model

The Offeror will include a high-level software architecture diagram showing all major software components included with the Offeror's system and the dependencies between them. The Offeror will include a high-level description of each component in the diagram.

Priority: MR

2.1.2.2 Software Sharing Strategy

Describe which major software components are open source, shared source (i.e., proprietary but shared with the Company), and closed source (i.e., proprietary and not shared with the Company).

The Offeror will commit to the timely sharing of open and shared source software developed by the Offeror and its subcontractors. The Offeror will describe its strategy for sharing this software with the Company before its general availability.

Priority: TR-1

2.1.2.3 Multiple Software Stacks

To the extent that it is relevant, Offeror will describe situations where provided software will include multiple "stacks" or tool chains with a similar purpose (e.g., programming environments supporting different compute devices) and their plans to handle aspects such as interoperability, integration, coordination, testing, and release practices and schedules to ensure that the resulting software suite provides the necessary quality, usability, and a positive user experience for OLCF users.

Priority: TR-1

2.1.2.4 Coordination of Open Source Software Development

To the extent that Offeror's provided software is based on open source software which includes changes or additions made by Offeror or their contractors, Offeror will describe their plans for coordination of upstream development with Offeror's development. Considerations include, but are not limited to plans to upstream changes made by Offeror or hold them private, and the level of relience on upstream development activities to deliver capabilities the Offeror considers central to their OLCF-6 offering.

Priority: TR-1

2.1.2.5 Software Licensing

The Offeror will describe their approach for software licensing, e.g., range of licenses and criteria for selection from that range of licenses for a particular package.

Priority: TR-1

2.1.2.5.1 Rebuildable Device Drivers and Kernel Modules

The Offeror should enable all provided device drivers or kernel modules to be rebuildable and manageable by the Company with the operating system proposed.

Priority: TR-1

2.1.2.5.2 Access to Source Code and Build Environment

The Offeror should provide access to source code, and necessary build environment, for all software except for firmware, compilers, and third-party products. The Offeror should provide updates of source code, and any necessary build environment, for all software over the life of the subcontract.

Priority: TR-1

2.2 REQUESTED OPTIONS

The Offeror will provide Technical Options to allow the Company to adjust various components within the system. Only provide descriptions of the options in this document; do not include the option prices. The prices should only be in the Price Proposal.

The Company will determine which options (e.g., additional racks, changes to memory capacity, changes to the interconnect) to exercise at the Go/No-Go decision point. The Offeror should describe which options remain viable after deployment (e.g., adding racks of CPU-only nodes, AI accelerator nodes, interesting architectures) and for how long after deployment.

2.2.1 Hardware Options

2.2.1.1 Scale the System Size

The Offeror should provide an option in the Price Proposal to increase and decrease the compute partition. Describe here the scalable unit (e.g., rack, N racks with CDU, row). The offeror should identify any thresholds requiring increased component infrastructure (e.g., extra spine switches, extra cables), any technical challenges foreseen with respect to scaling and any other production issues.

Priority: TO

2.2.1.2 Scale Processor Memory

For each processor type, the Offeror should provide an option to increase and decrease, if possible, the memory capacity. Describe the scaling units. The price should be per scaling unit (e.g., rack, N racks).

Priority: TO

2.2.1.3 Scale the Interconnect

The Offeror should provide options to increase and decrease the interconnect bandwidth. This option may include additional NICs per node, additional switches, additional cables, etc. Describe the expandability of the interconnect and Offeror should identify any thresholds requiring increased component infrastructure (e.g., extra spine switches, extra cables), any technical challenges foreseen with respect to scaling and any other production issues.

Priority: TO

2.2.1.4 Additional Maintenance Periods

The System Price should include five years of 24/7 maintenance. The Offeror should provide separate options for year 6 and year 7 maintenance and support for the system.

If the Offeror is a cloud provider, provide the price to continue the service for each year.

Priority: TO

2.2.1.5 Mid-Life Upgrades

The Offeror should describe and separately price any options for upgrading the proposed OLCF-6 system over its five year lifetime.

Priority: TO

2.2.1.6 Deinstallation

The offeror should provide an option to deinstall, remove and/or recycle the system and supporting infrastructure at end of life. Storage media should be wiped or destroyed to the satisfaction of the Company, and/or returned to the Company upon request.

Priority: TO

2.2.1.7 CPU-Only Nodes

In order to provide additional capabilities such as analysis and/or inference, the Offeror may propose a price for a scalable unit (e.g., rack, N racks with CDU) of CPU-only nodes. Describe the node architecture including processor architecture, number of processors, memory, connectivity within the node, connectivity to the high speed interconnect, and node-local storage, if any. The Offeror may choose to provide multiple options and prices for different architectures.

Priority: TO

2.2.1.8 Dual-Homed, CPU-Only Nodes

In order to provide additional capabilities such as analysis, inference, data-transfer, and/or workflow enablement, the Offeror may propose a price for a scalable unit (e.g., rack, N racks with CDU) of CPU-only nodes that connect to the high speed interconnect and to the Company's Ethernet fabric. Describe the node architecture including processor architecture, number of processors, memory, connectivity within the node, connectivity to the high speed interconnect, connectivity to the Company Ethernet fabric, and node-local storage, if any. The Offeror may choose to provide multiple options and prices for different architectures.

Priority: TO

2.2.1.9 Visualization Nodes

In order to provide additional visualization/rendering capabilities, the Offeror may propose a price for a scalable unit (e.g., rack, N racks with CDU) of visualization/rendering nodes that connect to the high speed interconnect. Describe the node architecture including processor architecture, number of processors, memory, connectivity within the node, connectivity to the high speed interconnect, and node-local storage, if any. The Offeror may choose to provide multiple options and prices for different architectures.

Priority: TO

2.2.1.10 Node-Local Storage

For each node type, the Offeror should provide an option to add NVMe drives with a target size of at least triple the node memory capacity. Describe the NVMe performance characteristics and system software requirements.

Priority: TO

2.2.1.11 AI and Analytics Acceleration Nodes

The Company is interested in novel AI and analytics acceleration technologies that will accelerate AI (and possibly modeling/simulation) workloads and that can integrate within an HPC ecosystem. The partition may be delivered with the OLCF-6 system or later and be made up of the scalable unit for the proposed design as determined by the Offeror. The partition should be made up of enough scalable units to achieve 1.0x AI benchmark described in Section 3.0. The description will include:

- an overall architectural diagram that shows all hardware, interconnect(s), compilation infrastructure, and Input/Output (I/O) subsystems, if applicable. The offeror should work with the Company to ensure the partition mounts the file system(s) to achieve the necessary level of performance.
- an overview of software architecture, including libraries and Software Development Kits (SDKs), support for frameworks (e.g., TensorFlow, PyTorch, etc.), usability and programmability. The description should include terms of software licensing, number of licenses included, and any support if applicable.
- available results or projections on MLPerf benchmarks, in particular the “Training”, “Training: HPC” and “Inference Datacenter” benchmark suites. Results and projections provided should specify the version of the benchmark used, whether the result was officially submitted and any required modifications to the benchmark rules required to obtain the reported results or projections.
- performance results (actual, predicted or extrapolated) for the proposed system for one or more of the benchmarks listed in Section 3.

The Offeror may choose to provide multiple options for different technologies. Each should be described and priced separately.

Priority: TO

2.2.1.12 Other Interesting Architectures

The Offeror may propose additional interesting technologies (e.g., data flow, coarse-grained reconfigurable, field programmable, neuromorphic) that can integrate within an HPC ecosystem.

The description should include:

- an overall architectural diagram that shows all hardware, interconnect(s), compilation infrastructure, and Input/Output (I/O) subsystems, if applicable. The offeror should work with the Company to ensure the partition mounts the file system(s) to achieve the necessary level of performance.

- an overview of software architecture, including libraries and Software Development Kits (SDKs), support for frameworks, usability, and programmability. The description should include terms of software licensing, number of licenses included, and any support if applicable.
- performance results (actual, predicted or extrapolated) for the proposed system for one or more of the benchmarks listed in Section 3.

The Offeror may choose to provide multiple options for different technologies. Each should be described and priced separately.

Priority: TO

2.2.2 Early Access Technology

2.2.2.1 Early Access Hardware

The Offeror should propose mechanisms to provide the Company with early access to hardware technology for hardware and software testing. Small, early-access systems with N-1 or N-2 processors are encouraged, particularly if they are sited at the Company. Access to other technologies (e.g., interconnect) are encouraged as well.

The Offeror may choose to provide multiple options for different early access systems with N-1 and N-2 technologies. Each should be described and priced separately.

Priority: TO

2.2.2.2 Early Access Software

The Offeror should propose mechanisms to provide the Company with early access to software technology and to test software releases and patches before installation on the OLCF-6 system.

Priority: TO

2.2.3 Test and Development Systems

2.2.3.1 Stand-alone Test and Development System(s)

The Offeror will provide a system configuration that consists of the minimum deployable system that mirrors the proposed OLCF-6 system. This option will include the smallest usable compute partition (e.g., rack, CDU) as well as a minimal front-end environment (e.g., management, login nodes). Options and costs for scaling the TDS up will be provided. Specifically, Offeror will describe and separately price options for the smallest usable compute partition, as well as options for any infrastructure required to add it to a larger system. Further, if the system includes different compute node types, the Offeror shall describe and separately price options to increase each compute node type independently of the other compute node types.

Priority: TO

3 BENCHMARKS

The OLCF-6 system will provide innovative solutions for hardware with a demonstrable path toward performance portability using a software stack and tools that will ease the transition without sacrificing DOE goals for continued delivery of science deliverables across modeling and simulation, artificial intelligence and machine learning, and large-scale data analysis.

The past 15-20 years of computing have provided an almost unprecedented stability in high-level system architectures and parallel programming models, with the MPI, OpenMP, C++, and Fortran standards paving the way for performance portable code. Combined with the application trends toward more coupled physics, predictive capabilities, sophisticated data management, object-oriented programming, and massive scalability – each of the applications that form the typical workload for OLCF represents tens or hundreds of person-years of effort, and *thousands* of person-years in aggregate. Thus, there is a keen interest in protecting the investment in the DOE application base by procuring systems that allow today's workhorse application codes to continue to run without radical overhauls. OLCF seeks solutions that minimize disruptive changes to software that are not part of a standard programming model likely to be available on multiple future acquisitions, while recognizing the need that the existing software base must continue to evolve.

The OLCF-6 benchmark suite has been developed to capture the programming models, programming languages, numerical motifs, fields of science, and other modalities of investigation expected to make up the bulk (e.g., more than 80% of all the consumed time on the platform) of the usage upon deployment.

Note: This section describes the OLCF-6 benchmarks list as of September 2023. Over the next few months, the list may change slightly and additional baseline data or run rules may be published.

3.1 BENCHMARK AVAILABILITY

The benchmark source codes are available via the Web at the following URL:

https://www.olcf.ornl.gov/benchmarks/

This site will be maintained with updated information throughout the proposal response period, including updates to instructions for build and execution, as well as the rare possibility of a change in the baseline Figures of Merit (FOMs) due to late discovery of a bug or issue with the baselining procedures performed by the benchmark teams.

The entire set of benchmarks have been executed on the existing ASCR Leadership Class systems at ORNL (i.e., Summit and/or Frontier) to provide baseline execution performance. The benchmark website provides the results of these runs as an aid to Offerors.

3.2 PERFORMANCE MEASUREMENTS - FIGURES OF MERIT

All performance measurements for the benchmarks are stated in terms of a FOM specific to each benchmark. Each benchmark code defines its own FOM based on the algorithm being measured in the benchmark and represents either (1) a rate of execution based on, for example, iterations per second or simulated days per day or (2) a time-to-solution. Most of the FOMs are defined so that they scale linearly (within measurement tolerances) with the delivered performance of the benchmark. For example, running a given benchmark 5x faster should result in a FOM that is ~5x larger. Likewise, running a 5x more complex problem (higher fidelity, more physics, or larger size) in the same amount of time should also result in a ~5x increase in the resulting FOM. If an application runs a 2x more complex problem and runs

4x faster than the baseline, this would be an 8x improvement overall. Note, applications with a time-to-solution FOM require strong scaling only.

The value of the FOM for each benchmark is described in its documentation. Each benchmark projected FOM is to be reported as an independent result. The Offeror simply needs to measure or to project the FOM on the target platform. The Offeror is free to additionally express the ratio of this determined FOM and the largest of the example run FOMs provided in the benchmark description. There is no further normalization across the benchmarks to be carried out, i.e., the Offeror is expected to simply report the FOM ratios for each benchmark separately.

3.3 BENCHMARKING PROCEDURES

Each benchmark includes a brief summary file and a tar file. Tar files contain source code and configuration files for carrying out each of the benchmark problems. In addition, some of the benchmarks require external data sets be used in the course of the runs, e.g., as input data. Instructions for obtaining these external files are given in the description of each benchmark. Summary files contain instructions for determining that the code has been built correctly and problems to run. RFP problems are usually characterized by a set of command line arguments that specify a problem setup and parameterization and/or input or configuration files. The benchmark website also contains output results from large scale runs of each benchmark on Summit and/or Frontier to assist the Offeror in estimating the benchmark results on the proposed OLCF-6 system.

3.4 ALLOWED MODIFICATIONS FOR BENCHMARK CODES

The source code and compile scripts downloaded from the OLCF-6 benchmark web site may be modified as necessary to get the benchmarks to compile and to run on the Offeror's system. Other allowable changes include optimizations obtained from standard compiler flags and other compiler flag hints that do not require modifications of the source code. Likewise, changes in the system software such as expected improvements to compilers, threading runtimes, and MPI implementations can be considered. Once this baseline configuration is accomplished, a full set of benchmark runs must be reported with this "as is" source code.

Beyond this, the benchmarks <u>can be optimized as desired</u> by the Offeror. Performance improvements from pragma-style guidance in C, C++, and Fortran source files are preferred. Wholesale algorithm changes or manual rewriting of loops that become strongly architecture specific are of less value. Modifications must be documented and provided in the Offeror's response. We encourage the Offeror to report both their "as is" numbers and their optimized numbers to highlight the "as is" potential of their machine and the performance that can be achieved with various amounts of effort. Please see the detailed run rules on the benchmarks site for further details.

3.5 BENCHMARK SUMMARY

3.5.1 Application Benchmark Summary Table

Benchmark	Description	Figure of merit description
MILC	Lattice quantum chromodynamics	Time to solution
LAMMPS	Molecular dynamics	Atom-steps/sec
M-PSDNS	Pseduo-spectral direct numerical simulation	
QMCPACK	Quantum Monte Carlo for materials	Monte Carlo samples/sec
FORGE	Large-language model training	Time to train large model
Spatter	Performance of sparse and irregular memory access.	Memory references/sec
Workflow	Multi-stage, dynamic workflow	Time to solution

Priority: TR-1

3.5.2 Microbenchmarks Summary Table

Provide a summary table of the microbenchmarks.

Microbenchmark	Description	Figure of merit
BabelStream	Measures accelerator bandwidth per MPI process	GB/s
py-DGEMM	Measures accelerator GEMM performance (preferably for all floating point types) per MPI process	FLOPs/sec
OSU All-to-All	Measures full system all-to-all performance for the largest possible power-of-two message size in accelerator memory.	GB/s

For each microbenchmark, assume similar number of processes per node as the benchmarks as in 3.5.1. That is, if the benchmarks use 8 processes per node, use 8 processes per node for these microbenchmarks as well.

Priority: TR-1

4 I/O SUBSYSTEM

The OLCF-6 compute system requires an I/O subsystem capable of serving two distinct storage workloads. The primary modeling/simulation workload needs a write-optimized, parallel file system (PFS) to handle application outputs and checkpoint/restart. This workload may comprise a combination of small file (e.g., 32 KiB) read and writes as well as large-block streaming writes.

The Company expects to have a growing AI workload that requires high IOPs for small, random reads. If a traditional parallel filesystem cannot adequately service both needs, the Company will entertain a second I/O subsystem tailored to the AI workload. This workload requires a shared resource capable of being read from all nodes within the job. Due to the random access pattern, this workload is not capable of using traditional file system caches (e.g., client side caches).

Both use cases need to support using a single compute node, a percentage of compute nodes (e.g., 20%), and the full system (i.e., at least 90% of compute nodes).

4.1 PARALLEL FILE SYSTEM

The Parallel File System (PFS) provides a single POSIX file system namespace designed to provide large-scale capacity and high-performance bandwidth to the OLCF-6 compute system and other compute and data resources at the center. The FS supports off-cluster connectivity to other compute systems, data transfer clusters, analysis systems, workflow nodes, etc. that may be homed on other network fabrics using diverse network technologies. The "Workflows Context" appendix to this document offers additional context to inform offeror responses.

If the Offeror is only proposing an I/O subsystem, assume 8 PiB for the SAHM. Provide options to scale the system up and down once the actual system SAHM is known.

4.1.1 PFS Architecture

The Offeror will provide a high-level architectural definition of the proposed PFS design including a clear definition all hardware and software major components, which network(s) it connects to, and subsystems for both client and server sides, as applicable.

The Offeror will provide a separately priced option for the PFS in the Price Proposal.

If the Offeror is only proposing an I/O subsystem, describe which interconnect technologies are supported.

Priority: TO

4.1.2 PFS Capacity

4.1.2.1 PFS Space

The PFS will have a usable and formatted POSIX namespace capacity to support storing 60 days of outputs, where each day adds 1.5x SAHM to the PFS. The total capacity can be calculated using the following formula:

Capacity(PFS) = SAHM * 1.5 * 60

Priority: TR-1

4.1.2.2 PFS Number of Files

The PFS POSIX namespace will have the capability of storing 100 billion file system namespace entries (files or directories).

Priority: TR-1

4.1.2.3 PFS Organization

The PFS should be built with scalable storage units (SSUs) and scalable storage clusters (SSCs). Offeror should describe data capacity, metadata capacity, and performance attributes of each SSU and SSC.

Priority: TR-2

4.1.2.4 PFS Connectivity

The PFS needs to support the OLCF-6 system as well as other resources located within the OLCF data centers. The ideal solution would have the PFS available even if the OLCF-6 compute system and interconnect are down for maintenance. The preferred approach is to dual-home the PFS on the OLCF-6 compute high-speed network (HSN) and on the Company's InfiniBand SAN.

If the PFS does not connect to the OLCF-6 HSN but does connect to a separate high-speed network, describe this high-speed network including projected, achievable bandwidth between the compute system and PFS as well as the connectivity to the Company's InfiniBand SAN.

Priority: TR-1

4.1.3 PFS Performance

4.1.3.1 PFS 15% SAHM write performance

The PFS will provide the capability of writing 15% of SAHM in 5 minutes using a sequential file-per-process workload. Performance will be reported with compression enabled on the PFS for all cases. Offeror will describe the compression algorithm used.

Offeror will describe the expected performance for the Single-client, Application Checkpoint, Application Restart, Application Cold Restart, Application Cold Reboot Restart, Hero Sequential, and Hero Random use cases.

If the Offeror's solution requires tiering to meet performance requirements, Offeror will describe the read/write performance of every tier. Additionally, the Offeror will describe the file object placement, file size, I/O request size, the tool and tool options used to measure the performance, number of CNs, and number of processes or threads per node used to derive each performance value.

Priority: TR-1

4.1.3.2 PFS small I/O transaction performance

The PFS will demonstrate at least 50,000 32KiB file create transactions per second in aggregate from multiple compute nodes, sustained for at least 20 seconds. A file create transaction consists of the following metadata operations: open (create), stat, write, read, close. Each compute node will create files in its own directory. The Offeror will specify the number of CNs required to achieve this performance. The performance of open, stat, write, read, and close operations will be reported separately.

Priority: TR-1

4.1.3.3 PFS tree walk performance

The PFS will perform a complete tree walk assuming 85% utilization of inodes of the entire PFS namespace and will purge 100 million files. The tree walk and purge operations will complete in at most 24 hours. The Offeror will describe all tools used and assumptions required to achieve this performance.

Priority: TR-1

4.1.3.4 PFS aged file system performance

At 85% utilization, PFS performance Application Restart and Hero Sequential use case performances should be the same as an empty (i.e., freshly formatted) file system. The Offeror should describe how the PFS will achieve this performance at 85% utilization.

Priority: TR-2

4.1.4 PFS Functionality/RRAS

4.1.4.1 PFS EAS and TDS

The Offeror will propose early-access (EAS) (e.g., N-1 generation) PFS hardware and software technologies for testing and development. Small additional early-access systems are encouraged, particularly if they are onsite. Offeror will provide separately priced options for each type of PFS server.

The Offeror will propose Test and Development System (TDS) identical to the final PFS configuration (e.g., device counts and capacities per server and/or chassis). The size of the TDS will be sufficiently large that all architectural features of the larger system are replicated. The TDS will support a wide variety of activities, including validation and regression testing. The Offeror will provide separately priced options for each type of PFS server.

Priority: TR-1

4.1.4.2 PFS fault resistance

The PFS architecture should optimize for field serviceability and should have no single points of failure. Offeror will describe performance and availability impacts in the event of FRU failure(s). The PFS will be capable of achieving Scheduled Availability of no less than 99.5%.

Offeror will describe FIT rates of all FRUs. Based on these FIT rates Offeror will describe MTBF, MTBI, and MTTR.

Priority: TR-1

4.1.4.3 PFS high availability

The Offeror will describe the FS's ability to minimize unscheduled downtime by using high availability and/or automatic failover.

Priority: TR-1

4.1.4.4 PFS in isolation operations

The Offeror will propose methods to operate PFS in isolation from OLCF-6 compute system and infrastructure resource(s). Describe any scenarios where a rebalance of resources is required after a reconfiguration and any scenarios where downtime for the entire PFS is required.

Priority: TR-1

4.1.4.5 PFS automatic rebuilds

The PFS will automatically initiate the rebuild processes for storage media failures. Failed storage media will be rebuilt with correct data and the data integrity and storage redundancy will be restored in at most 6 hours from the time that the failure is detected. The PFS will maintain 70% of the required bandwidth during rebuild.

The Offeror will describe the relevant data protection schemes, mechanisms for recovery, and expected performance impacts of storage rebuilds.

Priority: TR-1

4.1.4.6 PFS time to consistency

The Offeror will describe any exceptions to POSIX compliance, time to consistency, and any potential delays for reliable data consumption for the FS.

Priority: TR-1

4.1.4.7 PFS object storage interface

The PFS design should support an Object Storage interface. The Offeror should describe and quantify the read and write access performance relative to the POSIX I/O interface to the PFS.

Priority: TR-2

4.1.5 PFS System Administration

4.1.5.1 PFS power cycle performance

The PFS will complete power on and power off in a timely manner. The Offeror will describe the sequence of steps and timings for full PFS initialization and full PFS shutdown. Include any dependencies and how timings may scale with the size of the system.

Priority: TR-1

4.1.5.2 PFS system software upgrades

The Offeror's PFS solution should support firmware upgrades and system software upgrades in a timely manner. All firmware upgrades should be issued from the Linux command line. Offeror should describe the durations for both firmware flashing and system software upgrades. Include any dependencies and how timings may scale with the size of the system.

Priority: TR-2

4.1.5.3 PFS software updates

The Offeror should provide quarterly PFS software deliveries that track upstream feature releases and security vulnerability patching.

Priority: TR-2

4.1.5.4 PFS open source

The Offeror should describe open-source software components of the PFS. Patches to this software in support of PFS should be submitted upstream within 3 months.

Priority: TR-2

4.1.5.5 PFS quota support

The Offeror will provide features to enforce and report upon soft (accounting) and hard (enforcement) quotas based on uid, gid and project/fileset.

Priority: TR-1

4.1.5.6 PFS site telemetry integration

The Offeror's FS solution will support integration with site monitoring/telemetry framework.

Priority: TR-1

4.1.5.7 PFS configuration management

The Offeror will describe the configuration management and diagnostic capabilities of the PFS that address the following details of system management:

- Any effect or overhead of software management tool components on the CPU or memory available on PFS server nodes.
- Support for multiple simultaneous or alternative system software configurations.
- User activity tracking, such as audit logging and process accounting.
- Unrestricted privileged access to all hardware components delivered with the system.

Priority: TR-1

4.1.5.8 PFS site provisioning integration

The Offeror should describe:

- Integration with Company core infrastructure (RSA, LDAP, DNS, configuration management – puppet, email servers, etc.).
- Support for the use of site-developed and maintained system provisioning software – Anchor.
- Support for site-generated images based on local package repositories.
- Adherence to Company cybersecurity requirements/policies.

Priority: TR-2

4.2 AI-OPTIMIZED STORAGE

The AI-Optimized Storage (AOS) shall accommodate the I/O and storage needs of multiple workload types, particularly Distributed ML Training, Distributed Job Staging, Time-sensitive External Data Analysis, and Data-intensive Analysis workflows, and provide an on-demand capability for Object Storage. The AOS shall provide I/O behavior to compute jobs that maximizes bandwidth and IOPS, minimizes access latency, and protects user data in the event of a compute node failure during job execution via some sort of data protection scheme (e.g., erasure coding, replication). The AOS shall be accessible from the OLCF-6 compute system.

4.2.1 AI-Optimized Storage Description and Option

The Offeror will provide a high-level architectural definition of the proposed AOS design(s) including a clear definition all hardware and software major components and subsystems for both client and server sides, as applicable. The Offeror shall include one option, but Company encourages the Offeror to include multiple options. The Offeror and Company will evaluate the proposed option(s) as well as the FS to make a late-binding decision if an option should be exercised.

Priority: TO

4.2.2 AOS Capacity

The AOS in aggregate will have a usable capacity of at least 2 times the SAHM.

The pricing proposal will have options to increase/decrease the capacity. Describe the unit (e.g., node, chassis, rack) of increase/decrease here.

The Offeror may choose to provide options for different technologies. The Offeror and Company will evaluate each proposed technology and determine which, if any, option to exercise at the Go/No-Go decision.

Priority: TR-1

4.2.3 AOS Performance

The Offeror will describe the expected performance for the *Distributed ML Training, Distributed Job Staging, large-scale Application Checkpoint/Application Restart, Time-sensitive External Data Analysis, and Data-intensive Analysis,* and *Object Storage* use cases in terms of bandwidth and IOPS. The Offeror will describe the expected performance for each use case with respect to quintiles of the compute system node count.

Additionally, the Offeror will describe the file object placement, file object movement, file size, I/O request size, the tool and tool options used to measure performance, number of CNs, and number of processes or threads per node used to derive each performance value.

For the *Object Storage* interface to AOS, the Offeror will describe and quantify the read and write access performance relative to the POSIX I/O interface to AOS.

Priority: TR-1

4.2.4 AOS Functionality/RRAS

4.2.4.1 AOS EAS and TDS

The Offeror will propose mechanisms to provide early access (EAS) to AOS hardware and software technologies for testing prior to inserting the technology into the OLCF-6 system. Small additional early access systems are encouraged, particularly if they are onsite.

The Offeror will provide a Test and Development System (TDS) that is independent of the AOS. The size of the TDS will be sufficiently large that all architectural features of the larger system are replicated. The TDS will support a wide variety of activities, including validation and regression testing.

Priority: TR-1

4.2.4.2 AOS rebalancing

The Offeror will describe any scenarios where a rebalance of resources is required after a reconfiguration and any scenarios where downtime for the entire AOS is required.

Priority: TR-1

4.2.4.3 AOS fault resistance

The AOS architecture will optimize for field serviceability and the Offeror will describe any single points of failure in the AOS design. The Offeror will describe performance and availability impacts in the event of FRU failure(s).

The Offeror will describe FIT rates of all FRUs in the AOS. Based on these FIT rates, Offeror will describe MTBF, MTBI, and MTTR.

Priority: TR-1

4.2.4.4 AOS I/O Interfaces

The Offeror will describe all available I/O interfaces to O, including but not limited to POSIX I/O, Object, and other APIs. Describe any exceptions to POSIX compliance, time to consistency, and any potential delays for reliable local and remote data consumption.

Priority: TR-1

4.2.4.5 AOS open source tool support

The Offeror should describe open-source software solutions supported for the AOS. Open-source tools such as HVAC, UnifyFS, Spectral, or other similar solutions can be used to provide *Job-shared Storage.*

Priority: TR-2

4.2.4.6 AOS Site Telemetry Integration

The Offerer will support integration with site monitoring/telemetry framework.

Priority: TR-1

4.2.4.6.1 AOS Parallel Data Transfer Tool

The Offeror will describe and provide an efficient parallel data transfer tool between the FS and AOS.

Priority: TR-1

4.2.4.7 AOS Data Protection

The Offeror will describe any supported data protection schemes, mechanisms for recovery, and related performance impacts for AOS. The Offeror will describe any situation in which AOS requires erasure encoding to meet capacity, performance, or data integrity requirements.

Priority: TR-1

4.2.4.8 AOS data sanitization

The Company anticipates the need to sanitize persistent storage devices between compute jobs utilizing AOS to support HIPAA and export-controlled workloads. The Offeror should describe any data sanitization capabilities of the AOS.

Priority: TR-2

4.2.5 AOS System Administration

4.2.5.1 AOS System Software Upgrades

The Offeror's AOS solution should support firmware upgrades and software upgrades in a timely manner. All firmware upgrades should be issued from the Linux command line. The Offeror should describe the durations for both firmware flashing and software upgrades. Include any dependencies and how timings may scale with the size of the system.

Priority: TR-2

4.2.5.2 AOS Upstream Software Delivery

The Offeror should provide quarterly AOS software deliveries that track upstream feature releases and security vulnerability patching.

Priority: TR-2

4.2.5.3 AOS Open Source Upstreaming

Patches to open-source software included in the Offeror's AOS solution should be submitted upstream within 3 months.

Priority: TR-2

5 HIGH PERFORMANCE INTERCONNECT

5.1 NETWORK REQUIREMENTS

5.1.1 Network Description

The Offeror will describe the high-speed network including:

- High-level description of topology and routing protocols.
- Aggregate bandwidth and transfer rates to other networks including any bandwidth tapering across the network hierarchy.
- Expected bandwidth for all-to-all at full system scale.
- Scale of transfers and number of connections per interface and in aggregate.
- Hardware support for communication offload (e.g., GPUDirect, GPUDirectRDMA, SmartNIC)

Priority: TR-1

5.1.2 Lower Level Communication API

The Offeror will provide a lower-level communication (LLCA) API such as UCX or Libfabric. Describe any enhancements or limitations that can be expected in meeting support of the standards and the latest versions of the LLCA. Describe one and two-sided messaging support.

Priority: TR-1

5.1.3 Topology Extraction

The Offeror should provide a library or API for applications to extract topology information of their running applications.

Priority: TR-2

5.1.4 Resilience

The Offeror will describe the resilience features of the network including link and switch failure scenarios. Describe the number of links, network interfaces, and switch failures that can occur while maintaining connectivity and describe performance degradation as components fail.

Priority: TR-1

5.1.5 Management

The Offeror will describe the out-of-band management network and mechanisms to securely extend management segments through an intermediary network such as a data center network.

Priority: TR-1

5.1.6 Quality of Service

The Offeror should describe the capability and mechanisms that provide Quality of Service for the interconnect (congestion control, traffic classes, virtual channels). The Offeror should describe the configuration (mapping) of QOS for standard HPC use cases (Messaging Eager, Messaging Rendezvous, Collective Operations), and Storage system. The Offeror should describe enhancements to support complex workflows.

Priority: TR-2

5.1.7 External Connectivity

The Offeror will provide a high-bandwidth and resilient solution that allows external connectivity to and from the system to the OLCF data center network. This will support simultaneous transfers and is capable of 1 TB/s of aggregate throughput.

Priority: TR-1

5.1.8 External Connectivity IETF

The Offeror should describe how the external connectivity solution will utilize IETF standards-compliant technology for functionality that includes:

- Congestion control mechanisms across network boundaries.
- QoS required for reliable connections, with configurable buffers.
- Traffic draining capability at a link or adjacency level, with flow tracking ability using sampled flow or similar.
- Encryption and authentication.
- Integrated SDN with job management and scheduling systems.

The “Workflows Context” appendix to this document offers additional context to inform offeror responses.

Priority: TR-2

5.1.9 Off-Premise External Connectivity

If any resource is deployed off-premises, the Offeror should describe external connectivity between ORNL and the off-premises resource. Any Offeror-provided fiber connection should terminate in the Company's RDF in Building 5600.

Priority: TR-3

5.1.10 Compute Network Integration

Describe the ability to incorporate other technologies (e.g., storage, specialized racks) from other vendors into the high-performance network fabric.

The “Workflows Context” appendix to this document offers additional context to inform offeror responses.

Priority: TR-2

5.1.11 Protocol Support

For both management and HSN, the Offeror will describe support for:

- Jumbo frame, IPv6, IPv4, TCP/IP, UDP, and virtual networks support.
- The ability to control IP traffic using Access Control Lists.
- The ability to load balance and route traffic across multiple paths.

- The ability to dynamically configure routing and exchange routing information between the data center, storage, and other external networks.

Priority: TR-1

5.1.12 CXL

If CXL is offered, the Offeror should describe support for CXL 2.0 (or greater) standard in its CPU-, GPU-NIC and storage offering. Include the level of support for the intended usage models of CXL.io, CXL.mem, and CXL.cache.

Priority: TR-3

5.2 MESSAGING SOFTWARE REQUIREMENTS

5.2.1 MPI

The system will provide MPI libraries that support MPI 4.0 or higher and make accelerator-aware MPI available where this is supported by the accelerator vendor, and this will be capable of running a job at a full-system scale. The Offeror will describe any extensions or limitations to the MPI standard in the available MPI libraries.

Priority: TR-1

5.2.2 Instant-On

The Offeror will describe the instant-on capability of their solution meaning the time from application startup (e.g., srun) through MPI_Init for a full system run.

Priority: TR-1

5.2.3 PMI

The system will support the Process Management Interface (PMI). The Offeror will describe the version and supported integrations.

Priority: TR-1

5.2.4 Other PMI

The system should describe support for additional process management interfaces (PMI2, PMIX).

Priority: TR-3

5.2.5 Other Communication Libraries

The Offeror should describe any other provided communication libraries (e.g., Gasnet, SHMEM, Charm++, etc.)

Priority: TR-3

5.2.6 Optimized Collective Libraries

The Offeror should describe any optimized collective libraries (e.g., NCCL, RCCL, etc.)

Priority: TR-3

6 SYSTEM MANAGEMENT

6.1 SYSTEM MANAGEMENT ARCHITECTURE

6.1.1 System Management Hardware Design

Offeror will describe the system management hardware design of the proposed system. This will include the following descriptions:

- The hardware of the management infrastructure and what each proposed piece of hardware would be responsible for from a service perspective.
- The scalable utility storage solution for the management system to store data. This will be sized appropriately to store logs for the lifetime of the system as well as node images, kdump images, and telemetry data for 30 days.
- The out of band network management system will be designed for bandwidth and latency to support scalable boot, telemetry, and RAS.

Priority: TR-1

6.1.2 System Management Software Design

The Company values a traditional system management software design that minimizes abstraction layers and avoids a service orchestration system (e.g., Kubernetes). Describe the Offeror's proposed system management software:

- The system management software design and scalability of the proposed solution.
- How configuration and data are handled and stored separately.
- How high availability of the management software is provided and ensured by the proposed system software design.

Priority: TR-1

6.2 WORKLOAD MANAGEMENT

6.2.1 Workload Management Features

Offeror will provide a full-featured workload manager with native step management. Company currently uses Slurm and expects that the proposed solution will provide the features available in Slurm 23.02. If an older version of Slurm or an alternative workload manager is proposed, the Offeror will provide a detailed analysis of the differences between the Offeror's proposed solution and Slurm.

Offeror will ensure that the proposed workload manager supports the full functionality of the proposed system design, including process affinity, accelerator support, and high-speed network features.

Priority: TR-1

6.2.2 Workload Management Technical Requirements

The Offeror will provide the necessary scheduler to support scalable job launch, including node placement, topology-aware scheduling, rank reordering, power-aware scheduling, and node configuration and re-provisioning of nodes if supported by the hardware. The system design will not limit the scheduler's ability to support thousands of concurrent users and more than 2,000 concurrent batch jobs and 20,000 steps.

Priority: TR-1

6.2.3 Workload Management Job Startup Performance

The workload manager will support a full system run of /bin/true (or an equivalent of this command) on the full system at 1 rank per minimum allocatable unit, e.g., core, CPU, numa zone, GPU. The beginning and end time, i.e., from allocation to completion of the job, will be gathered from the scheduler logs and that time should not exceed 30 seconds.

Priority: TR-1

6.2.4 Workload Management Support Option

If the chosen workload manager is provided/maintained by a third party, Offeror should provide a level-3 support contract with the third-party company that permits Company to directly interact with the third-party for workload manager support. The Offeror should describe what value is provided by being the contract intermediary with the third party.

Priority: TO

6.3 OPERATING SYSTEM

6.3.1 Base Operating System

The system will include a commercially supported, non-proprietary base Linux operating system (BOS) environment on all visible service partitions (e.g., front-end nodes, service nodes, I/O nodes). The Offeror will describe the proposed Linux environment.

Priority: TR-1

6.3.2 BOS Distribution Requirement

The BOS provided by the Offeror should be based off either the Red Hat or SUSE Enterprise Linux distribution.

Priority: TR-2

6.3.3 BOS Distribution Consistency

The BOS provided by the Offeror should be one consistent version across all partitions of the system, i.e., compute, administrative, and service nodes.

Priority: TR-3

6.3.4 BOS Distribution Updates

The BOS provided by the Offeror should track major upstream releases by no more than 6 months and minor releases by no more than 3 months.

Priority: TR-2

6.3.5 BOS Distribution Security Updates

Security updates for the Offeror provided BOS on all partitions of the system must be provided monthly from the Offeror. Additionally, the Offeror will provide to the Company access to the Offeror's Linux distribution repository.

Priority: TR-1

6.3.6 Optimized BOS for Compute

The Offeror should describe any HPC relevant optimizations made to the compute partition operating system. Any such optimizations should be limited to opensource Linux kernel modules that can be rebuilt onsite, to provide an efficient execution environment for applications running up to full-system scale.

Priority: TR-3

6.3.7 BOS Debugging

The Linux kernel in the Offeror's BOS should function correctly when all common debugging options are enabled including those features that are enabled at compile time.

Priority: TR-2

6.3.8 BOS Kernel Dump and Crash Capability

The Offeror should provide kdump (or equivalent) that should work reliably, and dumps should work over a network, i.e., no node local storage of dumps. Crash (or other online and offline kernel debugger) should work reliably. Multiple parallel dumps across the compute partition should also be supported without overwhelming the management network or disrupting the high speed network.

Priority: TR-3

6.4 PLATFORM MANAGEMENT

6.4.1 Remote Management

The Offeror will describe remote manageability capabilities of the compute nodes, network switches, utility storage, power distribution units and servers comprising the system, including power control and console access, firmware updates, zero-touch provisioning, diagnostics, event logs, and alert capabilities. These capabilities will be accessible via documented APIs, preferably based on open standards, and a user interface.

Priority: TR-1

6.4.2 Scalable Platform Management

The Offeror will describe any features provided for scalable full-platform management software that automates the management of all hardware, provides a comprehensive overview of system operations, and automates whole-system maintenance actions. Relevant features include, but are not limited to, sequenced power up and power down of the system; summarization of temperature, power, and other sensors; automating firmware and configuration updates; maintaining an inventory of field-replaceable units over the system lifetime; collecting alert and error information from hardware.

Priority: TR-1

6.4.3 Platform Configuration Management

The Offeror will describe the system configuration management and diagnostic capabilities of the system that address the following details of system management:

- Any effect or overhead of software management tool components on the CPU or memory available on compute nodes.
- Support for multiple simultaneous or alternative system software configurations, including estimated time and effort required to install both a major and a minor system software update.
- User activity tracking, such as audit logging and process accounting.
- Unrestricted privileged access to all hardware components delivered with the system.

Priority: TR-1

6.4.4 Infrastructure as Code and Version Control

The Offeror should provide an Infrastructure as Code(IaC) tool, e.g. Ansible, Puppet, or Chef, that is stored in a version control system, e.g. git, to ensure the configuration of the system. Alternatively, the Offeror should provide the necessary APIs to manage the system without the Offeror system management tools competing for ownership of the configurations. Configurations should be stored in human readable file formats which are in the IaC tool, and not stored in a database.

Priority: TR-2

6.4.5 Single Points of Failure

The system will have no single points of failure that would cause a system outage as defined in the glossary. The system will remain in an operational or degraded state after the unexpected failure of, or planned maintenance on, any single FRU, server, or switch and during any repair or other maintenance action. The Offeror will describe RAS capabilities to mitigate single points of failure (hardware or software) and the potential effect on running applications and system availability.

Priority: TR-1

6.4.6 Platform Reliability as a Service for Jobs

The Offeror should describe the resilience, reliability, and availability mechanisms and capabilities of the system to mitigate any condition or event that can potentially cause a job interrupt and how a job maintains its resource allocation and is able to relaunch an application after an interrupt.

Priority: TR-2

6.4.7 Hardware Discovery

The Offeror will provide a systematic process or tool that discovers hardware information or describe the algorithmic process for collecting/generating items such as: Network MACs and geolocations.

Priority: TR-1

6.4.8 Compute Node Health Tests

The Offeror should describe how compute node health tests are conducted and what applications are used within the diagnostic tests. The diagnostic tests should be able to detect common hardware failures and stress the node sufficiently to reproduce hardware failure events. The diagnostic tests should also be able to run within the proposed WLM.

Priority: TR-2

6.5 SYSTEM SOFTWARE DEPLOYMENT

6.5.1 System Software Upgrade and Rollback

The system will include the ability to perform rolling upgrades and rollbacks on a subset of the system while at least half of the system remains in production operation. The Offeror will describe the mechanisms and limitations of the continuous deployment framework.

Priority: TR-1

6.5.2 Scalable Boot

The Offeror should describe the process for scalable boot, reconfiguring and rebooting of compute, server and any other node types in the system. The description should include an overview of the node boot processes (warm boot and cold boot defined in the glossary), including secure boot, stateless/stateful node provisioning, and infrastructure automation for customization and configuration of a node, the coordination, ordering and parallelism of the boot process, and techniques to provide rapid configuration and rebooting. Include how the time required to reboot scales with the number of nodes being rebooted. Cold boot should take no longer than 60 minutes and a warm boot should take no longer than 15 minutes.

Priority: TR-2

6.6 SYSTEM NETWORKS

6.6.1 Out-of-Band Management

The Offeror will describe the out-of-band management network and mechanisms to securely extend management segments through an intermediary network, such as a data center network.

Priority: TR-1

6.6.2 Dynamic Network Management System

The Offeror should provide a management platform that enables dynamic addition/removal of network segments while maintaining state tracking and recovery.

Priority: TR-2

6.6.3 Separate In-Band versus Out-of-Band Networks

The Offeror should provide an out-of-band management network to manage the platform controllers, i.e., BMCs, node controllers, etc. For the out-of-band network, the Offeror should provide a separate vlan or separate physical network from the in-band management network.

Priority: TR-3

6.7 DATA COLLECTION AND MONITORING

6.7.1 Platform Data Availability

The system will include a secure mechanism whereby all monitoring data and logs captured are available to the Company and will support an open monitoring API to facilitate lossless, scalable sampling and data collection to publish and subscribe to monitoring data. Any filtering that may need to occur will be at the option of the Company. The system will include a sampling and connection framework that allows the system manager to configure independent alternative parallel data streams to be directed off the system to site-configurable consumers.

- The Offeror's BOS will include standards-based system logging.
- The BOS will have the ability to log to local disk as well as to send log messages reliably to multiple remote systems.
- In case of network outages, the logging daemon should queue messages locally for up to 72 hours and deliver them remotely when network connectivity is restored.

Priority: TR-1

6.7.2 Monitoring Mechanisms

The system will include mechanisms to collect, provide, store, and generate alerts to monitor the status, health, and performance of the system. These mechanisms and data should adhere to available open standards (when available), be open source or provide documented APIs and data definitions if only a proprietary solution is available. The Offeror will describe these capabilities, which should include at least the following:

- Environmental measurement capabilities for all systems and peripherals and their sub-systems and supporting infrastructure, including power, energy consumption, voltage, cooling, and temperature, including sampling frequency, accuracy of the data, and timestamps of the data for individual points of measurement.
- Metrics related to memory, network and other error correction or faults.
- Metrics of both HPC protocols and TCP/IP flows. This will include switch and/or router data, load balancing, error counters, congestion state, throttling, throughput, and latency for select packets traversing the network(s).
- Resource utilization for memory, CPU, network, storage, and accelerator devices.
- The system as a whole, including all levels of integrated and attached storage, and their associated hardware performance counters, degraded components and impending failure.
- The Offeror will provide a tool, e.g., Simple Event Correlator, that monitors the system's hardware for events, and alert and take actions upon the system to notify and isolate problematic hardware.

Priority: TR-1

6.7.3 Monitoring Tools

The Offeror should provide tools for the collection, analysis, integration, and visualization of metrics and logs produced by the system (e.g., peripherals, integrated and attached storage, and environmental data, including power and energy consumption).

Priority: TR-3

6.7.4 Sharing of Monitoring Data

The Offeror should describe their internal data collection and monitoring solution and describe how that can be connected to a Company provided monitoring solution via a messaging bus. The Offeror can presume that all long term data collection from the Offeror's provided monitoring solution will be provided by the Company.

Priority: TR-3

7 USER ENVIRONMENT

7.1 USER ENVIRONMENT PACKAGING

7.1.1 Installation Mechanisms for Libraries

The Offeror should describe mechanisms that can be used to install the provided math, machine learning, and I/O libraries for each the supported compiler tool chains.

Priority: TR-2

7.1.2 Management for Multiple Versions of Packages

The Offeror should describe the mechanisms available to load specific packages and versions and the user interfaces available (e.g., environment modules) on the system.

Priority: TR-2

7.1.3 Build Support for Container Images

The Offeror will describe how non-privileged users will build container images to leverage hardware features and/or Offeror-provided software.

Priority: TR-1

7.1.4 Container Support for Process Execution

The Offeror will describe support for running processes inside containers to enable alternate userspace environments (e.g., Singularity/Apptainer, Podman) including capabilities and limitations to leverage hardware features and/or Offeror-provided software.

Priority: TR-1

7.1.5 Containerized Programming Toolchain

The Offeror should provide a containerized programming toolchain (i.e., compilers and supporting libraries including, at a minimum, MPI) to allow users to develop and build programs outside the System offering. The Offeror should describe any limitations in running such programs in the containerized environment on or outside the System.

Priority: TR-3

7.2 USER ENVIRONMENT LIBRARIES

7.2.1 Math Libraries

The Offeror will provide optimized BLAS, LAPACK, ScaLAPACK and FFT libraries for all compute devices and for all supported precisions. Describe the offering and its limitations.

Priority: TR-1

7.2.2 IO Libraries

The Offeror should describe support for optimized I/O libraries (parallel)netCDF 4.4.1.1 (or then current, (p)HDF5 1.14.0 (or then current), and ADIOS for all user-programmable compute devices. Describe any additional I/O libraries in the offering and their limitations.

Priority: TR-2

7.2.3 Data Visualization Libraries

The Offeror should describe support for data visualization libraries capable of efficient 3D rendering (e.g., OpenGL, Vulkan, ANARI), as well as the capability for hardware accelerated versions of said libraries for accelerated rendering.

Priority: TR-2

7.2.4 AOS I/O library support

The Offeror should describe I/O libraries provided that can take advantage of AI optimized storage (AOS)and any limitations (e.g., spectral, unifyFS).

Priority: TR-3

7.2.5 Library Support for Multiple Compilers

The Offeror should describe a mechanism to support the use of Offeror-supplied libraries using non-Offeror-supplied compiler toolchains.

Priority: TR-2

7.3 AI/ML SUPPORT

7.3.1 AI/ML frameworks

The Offeror should describe optimized framework libraries for execution of machine learning and AI workloads on user-programmable devices, such as those required for optimal execution of deep learning frameworks like PyTorch and Tensorflow.

Priority: TR-1

7.3.2 Distributed AI/ML

The Offeror will describe support for libraries to support efficient distributed training (data and/or model parallel), inference, hyper-parameter tuning (e.g., DeepSpeed, Ray, Horovod).

Priority: TR-1

7.3.3 AI/ML Library Interfaces

The Offeror will describe support for machine learning library interfaces with baseline languages.

Priority: TR-1

7.4 WORKFLOW ORCHESTRATION

The Offeror should describe software artifacts and any optimized capabilities of the hardware available to support orchestration of workflow tasks across nodes on the system, including support for distributed in-memory databases. The Offeror should describe any limitations regarding scalability/performance.

The "Workflows Context" appendix to this document offers additional context to inform offeror responses.

Priority: TR-2

7.5 APPLICATION DRIVEN RUNTIME POWER & ENERGY MANAGEMENT FOR ENERGY EFFICIENCY

The Offeror should describe APIs, and/or runtimes that enable non-privileged user applications to dynamically monitor and adjust their runtime power and energy usage by coordinating and controlling power management features exposed by the compute components allocated within a job allocation with one or more nodes. The power management features can incorporate compute component level capabilities provided by the system, including, but not limited to, boost and sleep states, modifications to frequency and voltage, and power caps.

Priority: TR-2

7.6 BASELINE LANGUAGES

7.6.1 Baseline Languages Support

The Offeror will provide fully supported implementations of Fortran 2023 (ISO/IEC FDIS 1539-1 - https://www.iso.org/standard/82170.html), C (ISO/IEC 9899:2018 - https://www.iso.org/standard/74528.html), and C++ (ISO/IEC 14882:2020 - https://www.iso.org/standard/79358.html) or the then current versions. Fortran, C, and C++ are referred to as the baseline languages. The Offeror will describe their timeline for implementing new standards of baseline languages.

Priority: TR-1

7.6.2 Baseline Languages Interoperability

The Offeror will fully support capability to build and execute programs from a mixture of the baseline languages (i.e., inter-language sub-procedure invocation shall be supported).

Priority: TR-1

7.6.3 Baseline Languages Parallelisms

The Offeror will describe how their implementation of parallelisms in the baseline languages enable the use of all user-programmable compute devices and any limitation it may have.

Priority: TR-1

7.6.4 Python Support for Standard Packages and Tools

The Offeror should describe their plans to provide a Python ecosystem on the platform including Offeror-provided optimized packages capable of fully utilizing user-programmable devices. Offeror should describe how optimized Python packages can be used with non-Offeror supplied Python distributions.

Priority: TR-2

7.6.5 POSIX Support

All baseline languages will support node parallelism through POSIX threads Version 2.0 (http://www.opengroup.org/onlinepubs/007908799/xsh/threads.html).

Priority: TR-1

7.6.6 DWARF support

The Offeror will describe their support for producing DWARF 5 (or then current) compliant debugging information for baseline languages and OpenMP, including for code compiled with some level of optimizations.

Priority: TR-1

7.6.7 Compiler Listings

The Offeror should provide compiler options to produce code listings showing optimizations performed and/or inhibitors to optimizations on a line-by-line, code block-by-code block, or loop-by-loop as appropriate for the baseline languages and the supported programming models.

Priority: TR-2

7.6.8 Multiple Toolchains Support

The Offeror should provide more than one programming toolchain to program all user-programmable compute devices. The Offeror should describe the interoperability between these programming toolchains.

Priority: TR-2

7.6.9 GCC Toolchain

The Offeror should provide a recent version of GCC as a programming toolchain.

Priority: TR-3

7.7 PROGRAMMING MODELS

7.7.1 OpenMP Support

The Fortran, C, and C++ compilers will support OpenMP version 5.2 (or then current). The Offeror will describe how their OpenMP implementation enable the use of all user-programmable compute devices and any limitation it may have.

The Offeror will describe their timeline for implementing new releases of the OpenMP specification.

Priority: TR-1

7.7.2 OpenACC Support

The Offeror should provide Fortran, C, and C++ compilers or interpreters that support node parallelism through OpenACC Version 3.0 (or then current) (http://openacc.org/) to support applications that currently use OpenACC.

Priority: TR-2

7.7.3 Programming Models Interoperability

The Offeror will describe capabilities and limitations of interoperability in using multiple supported programming models and runtimes within a single program. These include interoperability of memory allocations and parallelisms across all supported programing models (including baseline languages).

Priority: TR-1

7.7.4 GPU Programming Support

The Offeror will describe how they would support extant GPU-based programming models such as CUDA, HIP, and SYCL on their system offering.

Priority: TR-1

7.7.5 Abstraction Layer Functionality

The Offeror should describe the feasibility of using the supported programming models as a backend for common abstraction layer software (e.g., Kokkos) for all user-programmable compute devices.

Priority: TR-2

8 TOOLS

8.1 HARDWARE COUNTERS (HWC)

8.1.1 Compute Device HWC

The Offeror will describe hardware counters for all user-programmable compute devices in the Offered system that enable users to measure whole-device utilization, performance, and energy-efficiency characteristics of software executing on each device.

Priority: TR-1

8.1.2 Compute Unit HWC

The Offeror should describe hardware counters for all compute units/engines provided by each user-programmable compute devices in the Offered system that enable users to measure utilization, performance, and energy-efficiency characteristics at the compute unit/engine level of granularity.

Priority: TR-2

8.1.3 Network HWC

The Offeror should describe hardware counters for all network devices in the Offered system that enable users to measure network utilization and performance characteristics.

Priority: TR-2

8.1.4 Instruction-class HWC

The Offeror should describe hardware counters for each user-programmable device in the Offered system that enable users to measure the usage of varied classes of hardware architecture instructions for user software executing on each compute unit/engine provided by each device. Examples of instruction classes include floating-point and integer arithmetic, vector and matrix operations, and memory accesses. The description should differentiate across all data paths and data types/formats (e.g., differentiating FP64, FP32, and various FP16 formats).

Priority: TR-3

8.1.5 Memory Hierarchy HWC

The Offeror should describe hardware counters for each user-programmable device in the Offered system that enable users to measure quantities, latencies, and bandwidths of data accesses and movement within the entire memory and cache hierarchy for user software executing on each compute unit/engine provided by each device.

Priority: TR-3

8.1.6 Individual HWC Configuration and Measurement

The Offeror will describe how the hardware counters for each of the offered system's user-programmable compute devices will support basic configuration and measurement functionality including enabling and disabling the counter, reading the counter value, and resetting the counter to its default value. Where applicable, each counter will support configuration of the maximum counter value allowed. Upon reaching the maximum allowed value, the counter will trigger a counter overflow event and reset to the default value.

Priority: TR-1

8.1.7 Group HWC Configuration and Measurement

The Offeror should describe how hardware counters for each of the offered system's user-programmable compute device hardware counters will support group-based configuration and measurement functionality including enabling and disabling groups of counters, reading the values of counters in a group, and resetting the counters in a group.

Priority: TR-2

8.1.8 HWC Sampling

The Offeror should describe how hardware counters for each of the offered system's user-programmable compute devices will support sampling of counter values for individual counters or groups of counters. The sampling interval should be programmable, and support cycle-count-based or HWC event-based sampling (e.g., gathering a sample when a counter overflow event occurs).

Priority: TR-2

8.1.9 HWC instruction attribution

The Offeror should describe how the hardware counters for each of the offered system's user-programmable compute devices should support attribution of sampled counter values to specific, individual program instructions.

Priority: TR-3

8.1.10 HWC Sampling with Multiplexing

The Offeror should describe how the hardware counters for each of the offered system's user-programmable compute devices will support statistical profiling of counter values for individual counters or groups of counters, wherein multiple counters or counter groups are measured over time during a single program run (as opposed to having to re-run the program to collect data for all desired hardware counters) for the same device or compute unit/engine via hardware-assisted multiplexing. The multiplexing interval will be programmable.

Priority: TR-3

8.1.11 Thread-based HWC Profiling

The Offeror should describe how the hardware counters for each of the offered system's user-programmable compute devices will support execution thread-based profiling of counter values for individual counters or groups of counters, wherein counter or counter groups are measured for each thread of execution on the same device or compute unit/engine via hardware-assisted multiplexing. The multiplexing support should maintain per-thread counter value records in buffered memory that include cycle counts indicating the beginning and ending of the thread execution period and the accumulated counter value for that period.

Priority: TR-3

8.2 INFRASTRUCTURE

8.2.1 PAPI

The Offeror will provide an implementation of the Performance API (PAPI), version 7 or latest available, that allows non-privileged users to configure and read all compute and network hardware counters provided by the offered system.

Priority: TR-1

8.2.2 Hwloc

The Offeror will provide an implementation of the Portable Hardware Locality (hwloc) library that exposes the organization and attributes of the offered system's nodes to non-privileged user programs.

Priority: TR-1

8.2.3 Linaro Forge

The Offeror should provide the Linaro Forge software suite (scalable debugger, performance profiler, and performance advisor software) for the Offered system.

Priority: TO

8.3 DEBUGGING

8.3.1 Scalable debugger support

The Offeror will describe the infrastructure that enables a debugging tool operated by a non-privileged user to conduct baseline debugging of a program running across multiple nodes in the system, up to and including all nodes,that runs code on any subset (including all) of the Offered system's user-programmable compute devices and that may access data stored in any subset (including all) of the Offered system's user-accessible memory stores. If the infrastructure does not yet exist, the Offeror will commit to providing and fully documenting the infrastructure's programming interface(s) and will describe the planned approach for doing so.

For the purposes of this requirement, "baseline debugging" indicates the tool can interact with a running program ("the debuggee") under tool control in at least the following ways:

- create a debuggee
- attach to an already running program such that it becomes a debuggee
- detach a debuggee from the tool's control
- single step any thread of execution of a debuggee at the granularity of machine code and at the granularity of source code if debugging symbols are available for the program, with the ability to step into, step over, and step out of function calls
- set breakpoints in debuggee machine code and source code if symbols are available, and continue debuggee execution until any thread of execution reaches a breakpoint location
- define watchpoints in terms of the debuggee's state, and continue debuggee execution until any thread of execution satisfies a watchpoint condition
- examine and set the value of program variables if program symbols are available for the debuggee, of individual memory locations at the debuggee's finest available granularity of memory accessibility, and of compute device registers
- examine a stack back trace for each thread of execution in the program. If program symbols are

available for the debuggee, such backtraces should be correlated with line numbers and file names from the program's source code.

Priority: TR-1

8.3.2 Debugging containerized programs

If the Offered system supports running containerized programs, the Offeror will describe the capability to debug programs that run via containers on the Offered system. If the support does not yet exist, the Offeror will describe the plan for providing the support.

Priority: TR-1

8.3.3 Provide scalable debugger

The Offeror will provide a debugging tool that allows a non-privileged user to conduct baseline debugging of a debuggee running across multiple compute nodes in the system, up to and including all nodes. The tool must be implemented using the Offeror-provided debugging infrastructure (Scalable debugger support). If the tool does not yet exist, the Offeror will commit to providing such a tool and describe the planned approach for doing so.

Priority: TR-1

8.3.4 Generate core files

The Offeror should provide and fully document the capability to generate core file(s) that capture the state of a running program at the time the core file is generated. Such capability should be able to be triggered programmatically and if the running program experiences a catastrophic error such as a segmentation fault. Such state should include program state in all memory spaces used by the program including compute device registers, and program execution state sufficient for a debugging tool to generate an execution backtrace for all processes in the running program. The Offeror should document the format of the generated core file(s). If the capability does not yet exist, the Offeror should commit to providing and fully documenting the capability and should describe the planned approach for doing so.

Priority: TR-2

8.3.5 Quantitative debugger aggregation

The Offeror should describe capability of Offeror-provided scalable debugging tool to provide quantitative representations (e.g., statistics) about multivalued debuggee state, e.g., program arrays. Such state may extend across multiple, distinct memory spaces across multiple compute nodes.

Priority: TR-2

8.3.6 Qualitative debugger aggregation

The Offeror should describe capability of Offeror-provided scalable debugging tool to provide qualitative representations (e.g., graphical visualizations) of multivalued debuggee state (e.g., program arrays). Such state may extend across multiple, distinct memory spaces across multiple compute nodes.

Priority: TR-3

8.3.7 Scalable debugger GUI

The Offeror should describe the graphical user interface of Offeror-provided scalable debugging tool.

Priority: TR-3

8.4 PERFORMANCE

8.4.1 Event tracing tool support

The Offeror will describe the infrastructure that enables a performance tool operated by a non-privileged user to produce event traces that capture the performance of a program running across multiple nodes of the system, up to and including all nodes, and that runs code on any subset (including all) of the Offered system's user-programmable compute devices and that may access data stored in any subset (including all) of the Offered system's user-accessible memory stores. The event traces must capture the elapsed times and compute device(s) used to execute program events (e.g., function or kernel duration) and data transfers between distinct user-accessible memory stores. If the infrastructure does not yet exist, Offeror will commit to providing and fully documenting the infrastructure and will describe the planned approach for doing so.

Priority: TR-1

8.4.2 Profiling tool support

The Offeror will describe the infrastructure that enables a tool operated by a non-privileged user to produce profiles via sampling that capture the performance of a program running across multiple nodes of the system, up to and including all nodes, and that runs code on any subset (including all) of the Offered system's user-programmable compute devices and that may access data stored in any subset (including all) of the Offered system's user-accessible memory stores. The infrastructure must support sampling of compute device hardware counters. If the infrastructure does not yet exist, Offeror will commit to providing and fully documenting the infrastructure and will describe the planned approach for doing so.

Priority: TR-1

8.4.3 Energy usage support

The Offeror should describe the infrastructure that enables a tool operated by a non-privileged user to monitor the energy usage of a program running across multiple nodes of the system, up to and including all nodes, and that runs code on any subset (including all) of the Offered system's user-programmable compute devices and that may access data stored in any subset (including all) of the Offered system's user-accessible memory stores. If the infrastructure does not yet exist, Offeror should commit to providing and fully documenting the infrastructure and should describe the planned approach for doing so.

Priority: TR-2

8.4.4 Performance analysis of containerized programs

If the Offered system supports running containerized programs, the Offeror will describe the capability to conduct performance analysis of programs running via containers on the Offered system. If the capability does not yet exist, the Offeror will describe the plan to provide such support.

Priority: TR-1

8.4.5 Address association

The Offeror will describe how the Offeror-provided event tracing and sample-based profiling infrastructure will produce data that a non-privileged user tool can associate to the program's executable machine language instructions and data structure addresses.

Priority: TR-1

8.4.6 Symbols support

The Offeror will describe how the Offeror-provided event tracing and sample-based profiling infrastructure will produce data that a non-privileged user tool can associate to the program's executable code in terms of the symbolic names of executable constructs (e.g., functions) and data structures if such symbols were generated when the program's executable file(s) were produced.

Priority: TR-1

8.4.7 Documented performance data formats

The Offeror will fully document the format of the event traces and profile data produced by the Offeror's tracing and profiling infrastructure. If the infrastructure does not yet exist, the Offeror will commit to providing this documentation.

Priority: TR-1

8.4.8 Provide event trace reader/writer library

The Offeror should describe the planned approach for providing libraries that enable non-Offeror-provided tools to read and write event traces and profile data generated by the Offeror's tracing and profiling infrastructure.

Priority: TR-2

8.4.9 Use standard event trace format

The Offeror should document the format of event traces produced by the Offeror-provided event tracing infrastructure. OLCF has strong preference for the Open Trace Format 2 version 3.0.3 or latest available. If the infrastructure does not yet exist, Offeror should commit to providing this documentation.

Priority: TR-2

8.4.10 Provide scalable event trace analysis tool

The Offeror should provide a performance tool that allows a non-privileged user to conduct performance analysis via event tracing of the user's long-running and highly scalable program (e.g., running for multiple hours across all nodes in the system). The tool should be implemented using the Offeror-provided event tracing infrastructure. If this tool does not yet exist, Offeror should commit to providing and fully documenting the tool and should describe the plan for providing the tool.

Priority: TR-2

8.4.11 Provide scalable profiling tool

The Offeror should provide a performance tool that allows a non-privileged user to conduct performance analysis via sample-based profiling of the user's long-running and highly scalable program (e.g., running for multiple hours across all nodes in the system). The tool should be implemented using the Offeror-provided sample-based profiling infrastructure. If this tool does not yet exist, Offeror should commit to providing and fully documenting the tool and should describe the plan for providing the tool.

Priority: TR-2

8.4.12 Scalable tools provide GUI

The Offeror should describe the graphical user interface of the Offeror-provided performance tool(s).

Priority: TR-3

8.4.13 Provide automated optimization guidance tool

The Offeror should provide a performance tool, usable by a non-privileged user, that suggests changes the user might make to their program to improve its performance or scalability based on measured performance or scalability data produced by the Offeror's event tracing and/or sampling infrastructure. If the tool does not yet exist, Offeror should describe the plan for providing such a tool.

Priority: TR-3

8.5 CODE ANALYSIS

8.5.1 Memory bug detector

The Offeror should provide a tool that detects memory error bugs for programs. Such bugs include but are not limited to:

- Use after deallocation
- Accesses outside allocations
- Use after return or exit from scope
- Use before initialization
- Memory leaks

The Offeror should describe all memory bug detection capabilities and limitations of the tool, including which compute device(s) and memory store(s) are supported. If the tool does not yet exist, Offeror should commit to providing such a tool and describe the plan for providing the tool.

Priority: TR-2

8.5.2 Threading bug detector

The Offeror should provide a tool that detects data race bugs between threads of execution in programs. The Offeror should describe all thread bug detection capabilities and limitations of the tool, including which compute device(s) and memory store(s) are supported. If the tool does not yet exist, Offeror should commit to providing such a tool and describe the plan for providing the tool.

Priority: TR-3

8.5.3 Machine code analysis libraries

The Offeror should provide infrastructure that supports parsing and analysis of executable code (e.g., Dyninst's InstructionAPI and ParseAPI) for every user-programmable compute device in the system. If the infrastructure does not yet exist, the Offeror should commit to providing and fully documenting the infrastructure and should describe the plan for providing the infrastructure.

Priority: TR-3

9 SECURITY

The Company will have users who will work with protected data including HIPAA, ITAR, Export Controlled, Proprietary, etc., which will require the system to operate in a Moderate security enclave.

9.1 SECURITY MODEL

The Offeror will describe the security model(s) and/or features in place to prevent unauthorized access of information within the software, hardware, and network components of the system. The Offeror will provide a Threat Model describing points of information/session ingress and egress both external and internal to the system, as well as the types of protections in place at those points.

Priority: TR-1

9.2 IDENTITY MANAGEMENT

The Offeror will describe how their solution will interface with the Company's LDAP.

Priority: TR-1

9.3 VULNERABILITY DISCLOSURE & REMEDIATION

The Offeror should describe the process for remediating security vulnerabilities in hardware & software delivered by the Offeror, including vulnerabilities discovered by third parties, vulnerabilities in upstream components, and vulnerabilities discovered by the Company but not yet publicly known. In particular, the Offeror should describe timelines and the process for obtaining Common Vulnerability Enumeration (CVE) identifiers for vulnerabilities discovered within Offeror's products.

Priority: TR-1

9.4 SECURE SOFTWARE DEVELOPMENT FRAMEWORK

The Offeror should describe any Secure Software Development Framework (SSDF) processes in place applicable to Offeror's software products. The offeror should also describe the level of visibility the Company will have into Offeror's SSDF processes.

Priority: TR-2

9.5 EXECUTIVE ORDER COMPLIANCE

The Offeror will describe how the system software and hardware components have been designed to comply with the Presidential Executive Order on Improving the Nation's Cybersecurity (Cybersecurity Executive Order 14028). In particular, Offeror will describe to what extent Zero Trust Architecture (ZTA) and software supply chain security and integrity concepts have been built into the system software, hardware, and networking components.

Priority: TR-1

9.6 DISTRIBUTED WORKFLOW SECURITY

The Offeror should describe, if applicable, the security model for protecting long lived user-submitted workflow and other automated tasks to the system both internally and externally. In particular, Offeror should describe confidentiality and integrity related features, secret/credential management, and other user-boundary protections provided by the system components.

Priority: TR-3

9.7 OFFEROR PRIVILEGED ACCESS

The Company requires all Offeror personnel with privileged access to be US Persons. The Offeror will describe the anticipated number of personnel with privileges and their roles.

Priority: TR-1

10 FACILITIES

For off-premises proposals, this section does not apply.

For on-premises proposals, OLCF has two data centers that are available for system installations. Both facilities have a comparable amount of square footage, but the geometrical footprint is different. In addition, there are differences in floor structure and water distribution. For the remainder of this section, the centers are referred to as E102 and K100. The specific differences are broken out in the following subparts of the section.

An additional option for installation is also available: a split system. In this configuration, the compute system would be in one room and the storage system located in the other room. That option is also explained in additional subparts in this section.

The Offeror should indicate if their proposed system is specifically designed for one of these data centers.

10.1 FACILITY REQUIREMENTS (APPLIES TO ON PREMISE, ORNL)

10.1.1 Lay down Space

Lay down space for installation will be limited to the allocated installation floor space in the room. Delivery trucks will need to off load, wait for racks to be rolled into the space, peripherals to unpacked, and for packing material to be reloaded onto the truck.

Priority: TR-2

10.1.2 Delivery

The Offeror will deliver all material directly to the building 5600 dock. Deliveries will not be off-loaded at ORNL Central Receiving without consent from Company. All future part deliveries will be delivered directly to Building 5600 dock or delivered to an Offeror-provided off-site location and brought to 5600 by Offeror.

- All drivers must be US citizens. Delivery hours are restricted to 0730 to 1500 Eastern.
- Loads must be organized with a clear aisle way that allows security inspection or a walk path from the rear of the truck to the front of the truck.
- 5600 Dock is a single bay dock.
  - Loading dock: 9' x 14' (126 sq. ft.)
  - Loading dock height: 48" with -5"/+5" service range of edge of dock leveler
  - Edge of dock leveler: 72" wide w/ 20,000 lb. capacity
  - Rollup door width: 9'
  - Rollup door height: 8' 11"

Priority: TR-1

10.1.3 Installation Pathway Weight and Dimensions

The delivery path from the loading dock to the data center is concrete.

E102

- Once inside the data center aluminum plates will be required to help protect the access floor panels.
- The distance from the loading dock to the installation area is approximately 330 feet.

- The hallway dimensions leading to data center area 6' 5" wide by 8' 11" high.
- **Limiting height from floor to obstruction - 8 feet. An additional 3-4” could be obtained by removing a section of light changing the restriction to 8’3”.**

K100

- The distance from the loading dock to the storage area is approximately 400 feet. Same dock and height parameters.
- Limiting height from the floor to obstruction - 8’11”
- The hallway dimensions leading to data center area 6' 5" wide by 8' 11" high.

Priority: TR-1

10.1.4 Floor System

E102

The access floor system is a conventional 24” (609.6 mm) tile system with a top elevation of +36” above the suspended concrete floor slab below. Existing solid access floor tiles are concrete filled steel tiles with the capacity of 625 lbs/square foot or 2,500 lbs/square foot in the middle of a 24”x24” panel. The Offeror will provide shipped and operational weights of all equipment. Equipment weights will not exceed the capacity of the existing access floor tiles. Also provide uniform operational loads beneath cabinets, or individual foot loads, as applicable. The Offeror will provide dimensioned drawings of required tile cuts based on the proposed layout of the systems for coordination and approval by OLCF.

K100

The floor for the secondary location is slab on grade.

Priority: TR-1

10.1.5 Environmental Safety and Health Support

The Offeror should provide an ES&H support person to work with Company in the development of hazard assessments and as a point of contact for ES&H related concerns. Offeror should supply on-site ES&H support during major installation periods and during machine start-up/acceptance. On-site presence can be a graded approach as agreed upon by Company that matches site activities (e.g., increased support during peak receiving/installation versus less frequent visits during acceptance and periodic visits after acceptance). A graded approach can be also applied to the representative’s qualifications that matches the associated risk. Offeror representative can have collateral duties, but ES&H must be the primary function and the representative must have company authority to correct ES&H concerns, including stop work authority.

Priority: TR-2

10.1.6 Utility Locations

E102

Facility will provide water coolant from underfloor and power from the ceiling. Offeror's system will match the facility's infrastructure.

K100

Facility will provide both coolant and power from the ceiling. Current configuration has power supply located above the face of the cabinet and coolant supply located at the rear of the cabinet. Offeror's system will match the facility's infrastructure.

Priority: TR-2

10.1.7 Safety Requirements

Offeror personnel will practice safe work habits, and comply with all associated Company Environment, Safety and Health (ES&H) requirements.

Offeror will allow any component of the machine to be serviced, repaired, or replaced in a de-energized state without disabling the operation of more than 5% of the machine. Any de-energized component will completely isolate all subsidiary components, through hardware and not software (i.e., on-off switches or switch-rated circuit breakers), without any potential for re-energization. All equipment to be serviced while energized will be finger and tool safe.

Offeror's Job Hazard Analysis (SJHA) to be developed in coordination with Company. This document covers the safety steps and protections taken for all work activities that are required for the system delivery and installation.

ORNL site access training and facility access training (45 minutes combined) will be required for site and room access. Additional safety related training may be required based upon the installation process, i.e., fall protection, lock-out/tag-out, etc.

The ORNL electrical safety program that requires formal training for any person who will perform work on or near equipment under lock-out, tag-out configuration, including coordination with facility representatives and compliance with safety procedures and processes. Lock-out/tag out is an ORNL process that will be performed by ORNL facility and Offeror will overlock.

Hot or energized work is prohibited at ORNL. Energization of equipment will be made in coordination with ORNL Qualified Electrical Workers (QEWs).

All electrical equipment brought to the facility will be certified by a Nationally Recognized Testing Laboratory (NRTL).

Priority: TR-1

10.1.8 Network and Cabinet Cabling

Network cabling (e.g., system interconnect) should run above floor and be integrated into the system cabinetry.

A cable tray above the floor and below the ceiling that connects to the Company network will be provided by Company central to the machine at the end of the row.

Network cabling to an alternate storage location should be through a facility above-ceiling conduit path. Cable length is to be determined.

Power, network, and other cables should be neatly organized. Necessary cable management accessories are to be provided by the Offeror.

All network cables, wherever installed, should be source/destination labeled at both ends.

In rack power cables where both ends are not readily and easily visible should be labeled.

Where necessary, cables should be plenum rated and comply with NEC 300.22 and NEC 645.5.

Priority: TR-2

10.1.9 System Labeling

Offeror will provide and install labels for every rack, network switch, interconnect switch, node, and disk-enclosure with a unique identifier visible from the front and rear of the rack. The labels will be high-quality plastic so that they do not fall off, fade, disintegrate, or otherwise become unusable or unreadable during lifetime of the system. The Offeror will provide documentation on labeling conventions and update both labels and documentation when changes are made.

Priority: TR-3

10.1.10 Available Facility Floor Space

E102

Approximately 4060 ft^2 of gross square footage is available for the system. The contiguous space allows freight aisles and meets minimum clearance requirements for electrical distribution systems and ingress/egress. Area A in the space is approximately 54'x60'. An aisle break for utilities is in-between area A and area B with an aisle break for mechanical utilities. Area B in the space is 42'x20'. The larger space is not clear span as there is one load-bearing column (2' x 2') in that area. Other columns exist but are in the aisle break for mechanical utilities. The floor space is on a 36" raised floor over concrete slab.

K100

Approximately 4300 ft^2 of gross square footage is available for the system. The contiguous space allows freight aisles and meets minimum clearance requirements for electrical distribution systems and ingress/egress. The space is not clear span there are two load-bearing columns (2'x2' each) in that area. The floor space is slab on grade concrete slab.

Alternate Storage location or Split Systems

The Offeror must plan for all technical load to be included in one of these footprints. Optional space for storage could be in the space not used by the seller for the compute system. If the compute goes in E102, then K100 could be used for the storage option. The space for storage consists of up to 50 locations for 2' wide x 48" deep cabinets. This space also contains may contain a column depending upon positioning. Additional locations for cabinets maybe allocated by facility.

If the compute is located in K100, then E102 could be used for storage. A minimum of 50 locations could be made available in the primary location for 2' x 48" deep cabinets. Row size or number of cabinets would need to be determined.

10.2 ORNL FACILITY – ELECTRICAL DISTRIBUTION

10.2.1 Planned Power Budget

The planned power budget for the OLCF-6 system and associated I/O subsystem target is 30MW or less.

Priority: TR-2

10.2.2 Compute System Power Distribution

The electrical distribution method for the Offeror's compute solution will be based on 3-phase wye (Y) 277/480VAC. The maximum size of any circuit supplying a compute rack will be 200A. The maximum quantity of circuits supplying a compute rack will be no more than 4.

Priority: TR-1

10.2.3 I/O subsystem Power Distribution

E102

The electrical distribution method for the Offeror's I/O subsystem will be based on 3-phase wye (Y) 230/400VAC or 3-phase delta (D) 208VAC. The maximum size of any circuit supplying a subsystem rack will be 60A. The maximum quantity of circuits supplying a subsystem rack will be no more than 4.

K100

The electrical distribution method for the Offeror's I/O subsystem will be based on 3-phase delta (D) 208VAC. The maximum size of any circuit supplying a subsystem rack will be 60A. The maximum quantity of circuits supplying a subsystem rack will be no more than 4.

Priority: TR-1

10.2.4 Overhead Power Delivery

Delivery of electrical services should be assumed to be from overhead. Offeror may assume that Company will provide overhead cable tray, overhead busway, or similar. Offeror is responsible for cable management from the overhead electrical distribution system to their equipment.

Priority: TR-1

10.3 ORNL FACILITY – COOLING DISTRIBUTION

10.3.1 Compute System Cooling Budget

The planned cooling budget for the OLCF-6 compute system (or any rack that does not exchange air with the data center) is constrained to the following parameters:

Compute FWS Operating Envelope	**E102**	**K100**
Pressure drop across facility taps (PSID)	20	15
Allowable flow rate (GPM)	14,000	6,000
Minimum dT (Celsius)	8	8
ASHRAE FWS Water Class	W32	W32*
Maximum FWS Return Temp (C)	54	54

*Company can provide W21 using trim water and at higher utility cost. Company preference is a W32 cooling system.

Priority: TR-1

10.3.2 I/O System Cooling Budget

The planned cooling budget for the OLCF-6 I/O system (or any other rack that does exchange air with the data center) is constrained to the following parameters:

I/O FWS Operating Envelope	**E102**	**K100**
Pressure drop across facility taps (PSID)	20	15
Allowable flow rate (GPM)	1,000/W32; 600/W6	1,000
Minimum dT (Celsius)	8	8
ASHRAE FWS Water Class	W32*	W21
Maximum FWS Return Temp (C)	54	54

*Trim capacity must be provided by Offeror provided cooling distribution units to bring the FWS W32 down to W21 with FWS W6 cooling water.

Priority: TR-1

10.3.3 Supplementary Environmental Controls

Other facility-provided supplementary environmental control and other space specific aspects are described below. If the proposed system requires beyond described below, the Offeror is to provide as part of their contract.

Supplementary Facility Envelope Controls	**E102**	**K100**
Dewpoint < 15°C	Yes	Yes
FWS Temp < Space Dewpoint	Yes	Yes
Dewpoint > -7°C	Yes	Yes
Envelope Moist Barrier	Yes	Yes
Filtered Space Pressurization Air	Yes	Yes
General Space Sensible Cooling Capacity	Limited to < 1MW	Limited to < 150kW
FWS from below or overhead	Below	Overhead

Supplemental Air Cooling	Raised Floor Perforated Tiles	Overhead and very limited

Priority: TR-1

10.4 POWER AND COOLING REQUIREMENTS

10.4.1 Compliance with the Power Budget

The best value evaluation will include projected power and cooling costs for the five year life of the system. Without compromising performance objectives, the Offeror will minimize the power and cooling required by the proposed systems. The Offeror will describe how its proposal fits within the power budget. The Offeror will provide a detailed estimate of the total amount of power in kW (kilowatts) and kVA, and cooling in either refrigeration tons or BTU (British Thermal Units), required by the complete OLCF-6 system when running a stress workload (i.e., HPL). The estimate will describe the power and cooling loads for the individual racks (by rack type) and for each substantive component of the system. At a minimum, the Offeror will describe power and cooling estimates anticipated during special purpose / diagnostic / benchmarking efforts that may best approach the design TDP, and maximum, typical / notional, and idle (minimum) operation.

Priority: TR-1

10.4.2 Power Quality

The Offeror will describe the power factor, phase balancing, and harmonics management for their AC power systems.

Priority: TR-1

10.4.3 Fault Current and Fuse Protection

The Offeror will provide rack mounted power supplies, PDUs, or similar items that are rated to withstand 100kA available fault current. This requirement may be accomplished using fuses that are series rated with any downstream electrical devices in the racks. Fuse protection external to the racks will be provided by ORNL if the rack mounted equipment is not rated to withstand 100kA of fault current. The Offeror will provide a flexible interconnecting circuit from compute rack to an overhead termination box supplied by Company. Pin and sleeve connectors for 480VAC connections will not be used.

Priority: TR-1

10.4.4 I/O system Power Cables

E102

I/O subsystems will use no more than 60A cables with 208V or 400V rated IEC pin and sleeve watertight connectors.

K100

I/O subsystems will use no more than 60A cables with 208V rated IEC pin and sleeve watertight connectors.

Priority: TR-1

10.4.5 I/O System Redundant Feeds

I/O subsystems will utilize dual, redundant feeds with failover capability for connection to utility and UPS power.

Priority: TR-1

10.4.6 Power Terminations

Company will make direct connection of electrical service to the overhead termination box with flexible power cable supplied from compute cabinet. To avoid requiring multiple lug connections for each phase at the upstream overcurrent device, paralleling of conductors in raceway that supply compute racks will not be allowed.

Priority: TR-1

10.4.7 National Recognized Testing Laboratory Certified Equipment

All equipment will be NRTL-certified.

Priority: TR-1

10.5 MINIMAL ELECTRICAL AND MECHANICAL CONNECTIONS

10.5.1

The Offeror's solution should minimize total number of electrical and mechanical connections that are required, both inside and outside the racks.

Priority: TR-2

10.6 TOLERANCE OF POWER QUALITY VARIATION

10.6.1 Power Quality Events Design

The design of the power system for the OLCF-6 system will be tolerant of power quality events. The Offeror will describe the tolerance of their power system to power quality events, in terms of both voltage surge and sag, and in duration. All power supplies will be tolerant to voltage sag in accordance with the latest version of SEMI F47. Computer power at ORNL is reliable and clean, but not conditioned. There is no uninterruptible power available for the OLCF-6 compute system. 400/230 VAC components of the Offeror's solution, which might include the I/O subsystem, network components, and other infrastructure, may be supported by UPS systems in dual-fed configurations that can provide tolerance (short-term ride-through) to power quality events that exceed the SEMI F47 specification.

Priority: TR-1

10.7 POWER FACTOR AND HARMONIC CURRENT REQUIREMENTS

10.7.1 Power Factor

The power factor of the computer racks when operating at benchmark levels will be $\geq$ 0.98. The power factor of 230/400VAC components of the Offeror's solution, which includes the I/O subsystem, network components, and other infrastructure, will be $\geq$ 0.95.

Priority: TR-1

10.7.2 Harmonics

At benchmark power levels, the maximum total harmonic current and the maximum individual harmonic current levels will meet recommendations provided in Table 2 of IEEE STD 519-2014 for $I_{sc}/I_l \leq 20$.

Priority: TR-1

10.8 ELECTRICAL SAFETY AND POWER STANDARDS

10.8.1 Header Power Supply Equipment Standards

All equipment proposed by the Offeror will meet industry safety, and appropriate power quality and power supply standards. Equipment that is supplied by 480VAC systems will be rated for +10% nominal voltage.

Priority: TR-1

10.8.2 Zero Voltage Verification

All racks and system cabinets which are hard-wired, or which cannot be disconnected safely via plugs, will provide a means for zero-voltage verification (ZVV) or provide a documented procedure for ZVV acceptable to the Company.

Priority: TR-1

10.8.3 Electrical Design for Finger and Tool Safe

All systems will be electrically finger and tool safe, meaning that internal electrical distribution above 48V will be guarded.

Priority: TR-1

10.9 MECHANICAL

10.9.1 Scalable Unit(s)

10.9.1.1 Cooling Water Scalable Units

For each scalable unit type, water cooling loads will be provided at the worst case and idle power consumption levels. For each of those levels, the required flow rates and pressure drops will be provided for the facility water system (FWS). Supply temperature stability envelopes will be described at any FWS supply temperature fluctuation that could impact performance. The ASHRAE TC9.9 water class requirements will be identified. If no fans are used for air cooling, the heat loss to the space will be provided at the peak and idle power levels at the warmest ASHRAE WX FWS supply temperature.

Priority: TR-1

10.9.1.2 Air Cooling Loads Scalable Unit

For each scalable unit type, air cooling loads will be provided at the worst case and idle power consumption levels, as are the air flow and temperature requirements for any air cooled subsystem. The ASHRAE TC9.9 air class requirements will be identified. If air cooling requirements allow the scalable unit to have surface temperatures above the ASHRAE TC9.9 recommended air class envelope, provide the heat loss to the space at the peak and idle power levels.

Priority: TR-1

10.9.1.3 Compute Nodes Self Protection

Compute nodes will be capable of automatically reaching a safe state should any overheating occur.

Priority: TR-1

10.9.2 Compute CDUs

10.9.2.1 CDU Maintainability/Redundancy

Describe the redundancy and maintainability features of the CDU.

Routine maintenance (e.g., filter changes, sampling, pump maintenance/replacement, programming, etc.), should be accomplished without taking the associated compute racks offline.

Priority: TR-2

10.9.2.2 Primary side

10.9.2.2.1 Fail Closed Valves

Provide pressure independent flow control valves that fail closed. Time to stroke 100% shall be no less than 1 minute during controlled or failed actuation.

Priority: TR-3

10.9.2.2.2 Valve Size and Tuning

Control valves are to be sized to have the proper control authority and have PID tuning parameters accessible for optimization.

Priority: TR-1

10.9.2.2.3 Omitted Material and Minimization

All wetted materials exposed to the FWS are to be compliant with the Company's chemistry requirements. In Addition, carbon steel and aluminum are to be omitted. Brass is to be minimized.

Priority: TR-1

10.9.2.2.4 Valve Locations

The control valves are to be located directly off the Facility cooling water supply tap. Company will provide a check valve to be located on the return. The control valve shall close should a leak be detected within the cabinet(s) served by the CDU.

Priority: TR-3

10.9.2.2.5 FWS Return Temperature

FWS return temperatures should be as high as possible during all operations while staying within the Facility maximum requirements.

Priority: TR-1

10.9.2.3 Secondary side

10.9.2.3.1 Leak Protection and Alarms

A means for leak detection, shutdown, and alarm is to be provided at the cabinet level.

Priority: TR-1

10.9.2.3.2 System Design and Makeup Water

Makeup working fluid for the secondary cooling system can be provided at the cabinet, row, or system level. This system design, at whichever level, will be provided by the Offeror, and is to be accepted by and installation is to be coordinated with the Facility.

Priority: TR-2

10.9.2.3.3 Water Quality Ownership

Water quality is to be owned by Offerer for the life of the system. Water quality issues impacting the performance of the system are to be resolved by the Offerer to the extent full performance is restored.

Priority: TR-1

10.9.2.4 Controls

10.9.2.4.1 Facility Building Automation Interface Requirements

Ability to interface with the Facility's building automation system is to be BACnet/IP and physical connection is to be accessible from outside of the CDU.

Priority: TR-1

10.9.2.4.2 Controls Hardware Electrical Voltage

Controls hardware components and connections for telemetric data should be < 50V and accessible outside of panels with voltages > 50 V.

Priority: TR-2

10.9.2.5 Telemetry

Telemetric data for items such as water flows, water temperature, pressures, valve positions etc. Should be made available through BACnet/IP to ORNL. Offeror should list available telemetric data to be made available.

10.9.3 Rear Door Heat Exchangers (RDHXs)

If RDHXs are used, the following requirements apply.

10.9.3.1 RDHX Control Valves

The control valves are to be located directly off the Facility cooling water supply tap. Offeror will provide a check valve to be located on the return. The control valve shall close should a leak be detected from the cabinet served or the RDHX itself.

Priority: TR-2

10.9.3.2 Rack and Water Inlet Temperature

If air is exchanged with the data center, select RDHXs for 80F rack inlet and warmest water inlet temperature possible to keep room neutral.

Priority: TR-1

10.9.3.3 RDHX Water Connections

Connections to the bottom of any RDHX shall be a minimum of 6" above the raised floor and be as close to the hinge as possible.

Priority: TR-3

10.9.3.4 RDHX Facility Building Automation Interface Requirements

Ability to interface with the Facility's building automation system is to be BACnet/IP and physical connection is to be accessible from outside of the CDU.

Priority: TR-2

10.9.3.5 RDHX Controls Hardware Electrical Voltage

Controls hardware components and connections for telemetric data should be < 50V and accessible outside of panels with voltages > 50 V.

Priority: TR-2

11 PROJECT MANAGEMENT

11.1 PROJECT MANAGEMENT

The Offeror will describe their project management approach, including:

- A discussion of any requirements that the Offeror has for the management of the project, including any conformance to national or international standards, if applicable.
- A discussion of the proposed project management staffing structure, including key roles, responsibilities, authorities, and accountabilities, including specific provisions for Offeror Project Manager, System Architect, and Executive Liaison roles.
- A discussion of the proposed schedule with key progress milestones and the Offerors' approach to schedule development, maintenance, and progress monitoring.

Priority: TR-1

11.2 PROJECT MANAGEMENT DOCUMENTS

The successful Offeror should commit to develop, deliver, submit for approval, and maintain the following planning deliverables. Initial versions and updates of these plans are to be provided in specific agreed-upon times following award. Each of the plans and any revisions are to be submitted for comment and approval to the Company's project management leadership.

- Project Organization Chart, including specific provisions for Offeror Project Manager, System Architect, and Executive Liaison roles.
- Project Schedule
- Risk Management Plan
- Risk Register
- Quality Assurance Plan
- Change Management Plan
- Site Preparation Guide (if applicable)
- System Delivery and Installation Plan (if applicable)
- Maintenance and Support Plan
- System Administration Guide

The specific details are designed to help the Offeror successfully meet its commitments, to help the Company track the project, and to help the Company and the Offeror to understand and mitigate risks.

Priority: TR-2

11.3 PAYMENT MILESTONES

In the Price Proposal, the Offeror will propose a set of payment milestones (i.e., criteria and payment amount) applicable to Offeror's proposed development and deployment timeline and methodology. Offeror is encouraged to identify milestones for each year of the project that merit revenue that the Offeror can legally recognize in that year.

Priority: TR-1

11.4 TECHNICAL DECISION POINT

The successful Offeror and Company will hold a Technical Decision Point evaluation and joint planning meeting 9-12 months before System delivery. At the Technical Decision Point meeting the final configuration of the System will be determined based on technology status and pricing. Performance targets will be re-evaluated and converted into requirements.

Priority: TR-1

11.5 RISK AND RISK MANAGEMENT

The Offeror will describe their risk management approach for the project, including:

- A discussion of any requirements that the Offeror has for risk management, including any conformance to national or international standards, if applicable.
- A discussion of what the Offeror considers to be the major risks for the project, including planned or proposed mitigations for those risks.

Priority: TR-1

11.6 QUALITY ASSURANCE

The Offeror should describe their hardware, software, and services quality assurance approach for the project, including:

- A discussion of any requirements that the Offeror has for quality assurance, including any conformance to national or international standards, if applicable.
- A discussion of the Offeror's quality goals for the system and software.
- A discussion of the Offeror's approach to hardware and software defect detection and reduction during system design, development, deployment, and operations.

Priority: TR-2

11.7 HIGH-LEVEL TECHNICAL COLLABORATION

The Offeror and Company will have regular calls/meetings to expedite identification and mitigation/correction of critical issues. The Offeror's System Architect will be responsible for identifying Offeror's technical/engineering staff needed to address these issues and ensure their attendance at these meetings as needed.

Priority: TR-1

11.8 WORKING GROUPS

The Offeror will commit to form the relevant project Working Groups (WGs). The purpose of each WG is to ensure that the delivered system meets the requirements set forth in the Build Statement of Work (SOW). WG members from the Offeror, Offeror's subcontractors, and Company should understand the Build SOW requirements that releate to the WG's charter.

The project will take the form of a partnership that is committed to delivering the most useful system possible. Upon subcontract award, the selected Offeror and the Laboratory will assess the project for areas in which deep collaboration is necessary to ensure meeting project goals. The partnership will form WGs for these topics. Each WG will interact in regard to all details of the technical topic; the selected Offeror will not attempt to limit the scope of these interactions. The WGs will serve as a key conduit to identify, refine, and understand requirements in detail and to ensure that the delivered system meets those requirements to the greatest extent possible. WGs will establish a regular schedule for meetings. Each Quarterly face-to-face meeting may include WG breakout sessions. Project management will regularly assess WG progress and identify topics for which WGs are no longer required or additional topics for which new WGs are needed.

Priority: TR-1

11.9 PLANNING KICK-OFF

The Company and successful Offeror will schedule and complete a Project Planning Kickoff Meeting to mutually understand and agree upon project management goals, techniques, and processes for the OLCF-6 system and the NRE subcontract. The kickoff meeting will take place no later than 45 days after contract award.

Priority: TR-1

11.10 QUARTERLY REVIEWS

The selected Offeror will commit to organizing and conducting Quarterly Project Reviews, which will include executive and working group breakout sessions. Quarterly Reviews will include discussions of:

- Plan of record status
- Risk management status
- Hardware development status
- Software development status
- NRE status (as applicable)
- Quality assurance status

Quarterly reviews will commence upon contract award and continue until final acceptance.

Priority: TR-1

11.11 ACCEPTANCE

Upon delivery, a series of functionality, performance, and stability tests will be performed prior to acceptance. Acceptance testing will comprise multiple components where the overall goal is to ensure that the system is high-performance, scalable, resilient, and reliable. Acceptance testing will exercise the system infrastructure with a combination of benchmarks, forced failures, and stability tests. Any requirement described in the Technical Specification may generate a corresponding acceptance test. The Company may identify other system aspects that merit testing; Offeror shall not attempt to limit the capabilities that the Company may test. The specifics of the acceptance test plan will be determined during contract award negotiation. These acceptance requirements apply to the main system and to the IO Subsystem as well as any additional systems procured as the result of exercising any options described in Section 2.2.

Priority: TR-1

12 NON-RECURRING ENGINEERING (NRE)

The Company expects to award a Non-Recurring Engineering (NRE) subcontract, separate from the system build subcontract. It is anticipated that the NRE subcontract could be approximately 10% of the OLCF-6 system budget. The offeror is encouraged to provide proposals for areas of NRE it feels will provide substantial value. Proposed NRE areas should focus on topics that provide added value that enhance or go beyond planned roadmap activities. Proposals should not focus on one-off point solutions or gaps created by the offeror's proposed design that should be otherwise provided as part of an integrated solution.

12.1 CENTER OF EXCELLENCE

The Offeror will include in the NRE proposal a Center of Excellence task that consists of multiple milestones to support porting and improving the performance of DOE applications on the system. Activities will require the support of experts in the areas of application porting and performance optimization, who will work with Company personnel on porting and tuning of key applications, which may include benchmarks or full applications—to be determined during contract negotiation—for the proposed system. This task should be run as its own project, with a coordinator/project manager overseeing and coordinating issues as appropriate. Colocation of staff at the Company is desirable but not necessary. Base support is required from the date of subcontract execution through 2 years after final acceptance. The Laboratory may negotiate an extended period of performance or options for such an extension. This activity will reflect all terms and conditions of NRE activities including cost sharing.

Priority: TO

13 MAINTENANCE AND SUPPORT

13.1 24/7 MAINTENANCE

The Offeror will support for all systems for a period of five (5) years from the date of acceptance. If maintenance is not included in the system price, the Offeror will provide separately priced options on a per year basis, for support on a 24 hours a day, seven days a week, with a four hour response time. The Offeror will provide hardware fault diagnosis and fault determination. Redundant or non-critical components should carry 9x5 Next Business Day (NBD) support contracts.

Priority: TR-1

13.2 ANALYST SUPPORT

The Offeror will supply two dedicated system analysts. Analysts must be available travel to Company at least quarterly, though full-time location at Company is preferred. Analysts will be highly skilled in Linux systems programming and be a subject matter expert for some aspect of the system (e.g., I/O subsystem, High Performance Interconnect, User Environment software). Analysts will support Company personnel in providing solutions to the current top issues. Offeror may propose and Company may request additional on-site analysts, which will be priced separately.

Priority: TO

13.3 ENSURING NODE AVAILABILITY

For on-premises systems, the Offeror will provide an on-site parts cache of FRUs and hot spare nodes of each type proposed for the system. The size of the parts cache, based on Offeror's MTBF estimates for each component, will be sufficient to sustain necessary repair actions on all proposed hardware and keep them in fully operational status for at least one month without parts cache refresh. The required size of the parts cache will be recalculated at least every six months. The Offeror will resupply/refresh the parts cache as it is depleted for the five year hardware maintenance period. System components will be fully tested and burned in prior to delivery to minimize the number of "dead-on-arrival" components and infant mortality problems.

For an off-premises system, the Offeror will provide a SLA for the number of nodes available.

Priority: TR-1

13.4 SUPPORT TRACKING

The Offeror will provide a means for tracking and analyzing all software updates, software and hardware failures, and hardware replacements over the lifetime of the system. The Offeror will provide reports of or read-only access to this information to the Company.

Priority: TR-1

13.5 NON-VOLATILE MEDIA RETENTION

Company requires that all non-volatile media must be retained at the system site until media destruction or cryptographic erasure in compliance with Company security policies. This includes disk spindles and non-volatile flash media that are encountered during regular maintenance and at system decommission. Company will bear the cost for destroying non-volatile media for media located at the Company for the life of the subcontract. Offeror’s maintenance strategy will take the non-volatile media destruction policy into account.

Priority: TR-1

14 GLOSSARY

14.1 AI-OPTIMIZED STORAGE (AOS)

AI-optimized Storage (AOS) is defined as providing a specialized storage space accessible from the compute system designed to accelerate the AI workloads (random small read I/O).

14.2 COLD BOOT

Cold boot is defined as the booting of the system from a completely powered off state. An example of this situation is a complete power loss to the facility. The Offeror can assume that all power, cooling, and infrastructure provided by the Company is complete and there are no additional requirements for the system to start the cold boot process. Booting of the system includes all infrastructure, hardware, software, and any file systems required for the system to operate as designed, including the I/O subsystem. The system boot will progress without human intervention except for a final release to start batch jobs when all hardware and software is ready.

14.3 FPP

File-per-process is defined as an I/O workload where each process in a parallel application writes/reads data to/from its own independent file.

14.4 HERO RANDOM

The "hero random" test is to measure the best performance achievable using FPP random offset I/O from sufficient CNs with an aggregate data volume that is at least twice the memory capacity of the number of CNs used in this test.

14.5 HERO SEQUENTIAL

The "hero sequential" test is to measure the peak aggregate performance achievable using FPP sequential I/O for both writing and reading data in parallel using a sufficient number of CNs. The aggregate data volume must be at least twice the memory capacity of the number of CNs used in this test.

14.6 JACNM

Job Aggregate Compute Node Memory is defined as the total amount of HBM memory in the compute nodes allocated to a given compute job.

14.7 JOB-SHARED STORAGE

Job-shared Storage is defined as providing Job-specific Storage as a unified storage namespace shared by all CNs in a job.

14.8 JOB-SPECIFIC STORAGE

Job-specific Storage is defined as allocating ephemeral storage resources to the set of CNs assigned to a job.

14.9 NODE-SHARED STORAGE

Node-shared Storage is defined as providing Job-specific Storage as an independent storage namespace to each CN in a job.

14.10 PFS

Parallel File System Storage

14.11 SYSTEM AGGREGATE HBM MEMORY (SAHM)

System Aggregate HBM Memory (SAHM) is defined as the total amount of HBM memory in all compute system nodes.

14.12 SCHEDULED AVAILABILITY LEVEL

The "Scheduled Availability Level" is defined as the percentage of total system operating time, excluding scheduled maintenance, during which stored data is accessible and new data can be stored.

14.13 SSF

Single-shared-file is defined as an I/O workload where parallel application processes concurrently write/read data to/from a single shared file.

14.14 SSU

A "scalable storage unit" is defined as a storage system building block representing the minimum fault domain for a network storage target.

14.15 SSC

A "scalable storage cluster" is defined as the smallest grouping of SSUs and supporting infrastructure required to operate as an independent POSIX file system namespace.

14.16 SYSTEM OUTAGE

A System Outage is defined as being more than 5% of the total compute nodes being unavailable to run jobs from the WLM. This could be due to hardware failures or a system service being unavailable that would prevent a compute node from being healthy to run a job from the WLM.

14.17 WARM BOOT

Warm Boot is defined as the time it takes for only the compute nodes to power on from a powered off state.

14.18 HARDWARE

14.18.1 CN

System compute nodes. Compute Nodes (CN) are nodes in the system on which user jobs execute.

14.18.2 Core

Portion of processor that contains execution units (e.g., instruction dispatch, integer, branch, load/store, and floating-point), registers and typically at least L1 data and instruction caches. Typical cores implement multiple hardware threads of execution and interface with other cores in a processor through the memory hierarchy and possibly other specialized synchronization and interrupt hardware.

14.18.3 DDR DIMM

Double data rate dual in-line memory module.

14.18.4 FEN

Front End Nodes. Front End Nodes are nodes where users and administrators can login in and interact with the system.

14.18.5 FLOP

Floating Point Operation.

14.18.6 FLOPS

Floating Point Operation per second.

14.18.7 FMA

Fused Multiply Add (FMA) is a single 64b or 32b floating-point instruction that operates on three inputs by multiplying one pair of the inputs together and adding the third input to the multiply result.

14.18.8 FPE

Floating Point Exception.

14.18.9 GB

gigaByte. gigaByte is a billion base 10 bytes. This is typically used in every context except for Random Access Memory size and is 109 (or 1,000,000,000) bytes.

14.18.10 GFLOPS or GOP/s

gigaFLOPS. Billion (109 = 1,000,000,000) 64-bit floating point operations per second.

14.18.11 GiB

gibiByte. gibiByte is 1,073,741,824 base 10 bytes (i.e., 10243 bytes).

14.18.12 HBM

High Bandwidth Memory.

14.18.13 IBA

InfiniBand™ Architecture (IBA) http://www.infinibandta.org/specs

14.18.14 ISA

Instruction Set Architecture.

14.18.15 MB

megaByte. megaByte is a million base 10 bytes. This is typically used in every context except for Random Access Memory size and is 106 (or 1,000,000) bytes.

14.18.16 MFLOPS or MOP/s

megaFLOPS. Million (106 = 1,000,000) 64-bit floating point operations per second.

14.18.17 MIB

Management Information Base is a database used for managing the entities in a communication network, typically used with SMNP.

14.18.18 MiB

mebiByte. mebiByte is 1,048,576 base 10 bytes (i.e., 10242 bytes).

14.18.19 MTBAF

Mean Time Between (Hardware) Application Failure. A measurement of the expected hardware reliability of the system or component as seen from an application perspective. The MTBAF figure can be developed as the result of intensive testing, based on actual product experience, or predicted by analyzing known factors. Hardware failures of or transient errors in redundant components such as correctable single bit memory errors or the failure of an N+1 redundant power supply and do not cause an application to abnormally terminate do not count against this statistic. Thus, MTBAF ≥ MTBF.

14.18.20 MTBF

Mean Time Between (Hardware) Failure. A measurement of the expected hardware reliability of the system or component. The MTBF figure can be developed as the result of intensive testing, based on actual product experience, or predicted by analyzing known factors. See URL: http://www.t-cubed.com/faq_mtbf.htm

14.18.21 MTTR

Mean Time to Repair. A measurement of the average time it takes for a failed component to be returned to the system.

14.18.22 NCORE

The number of cores in the CN allocatable to and directly programmable by user MPI processes. If the peak petaFLOPS system characteristic requires multiple threads per core to be issuing floating-point instructions, then NCORE is the number of allocatable cores times that number of threads.

14.18.23 Node

An instance of hardware, compute, or storage, with its own operating system. A node may have one or more processors (and processor types), memory, other storage, and NICs. A node may be physically distinct (i.e., a 2U server) or multiple nodes may be on a blade.

14.18.24 Non-Volatile

Non-volatile memory, nonvolatile memory, NVM or non-volatile storage, is computer memory that can retain the stored information even when not powered.

14.18.25 NUMA

Non-Uniform Memory Access architecture.

14.18.26 NVRAM

Non-Volatile Random Access Memory.

14.18.27 PB

petaByte. petaByte is a quadrillion base 10 bytes. This is typically used in every context except for Random Access Memory size and is 1015 (or 1,000,000,000,000) bytes.

14.18.28 Peak FLOPS Rate

The maximum number of 64-bit floating point instructions (add, subtract, multiply or divide) or operations (instructions) per second that could conceivably be retired by the system.

14.18.29 Peta-Scale

The environment required to fully support production-level, realized petaFLOPS performance.

14.18.30 PiB

pebiByte. pebiByte is 1,125,899,906,842,624 base 10 bytes (i.e., 10245 bytes).

14.18.31 Processor

The computer ASIC die and package.

14.18.32 Scalable

A system attribute that increases in performance or size as some function of the peak rating of the system.

14.18.33 SECDED

Single Error Correction Double Error Detection. Storage and data transfer protection mechanism that can detect parity errors (single bit errors) and detect storage or data transfer errors with multiple bits in them.

14.18.34 SIMD

Single Instruction, Multiple Data (SIMD) instructions are processor instructions that operate on more than one set of input 64b or 32b floating-point values and produce more than one 64b or 32b floating-point value. Fused Multiply-Add (FMA) instructions are not SIMD. Examples of this are x86-64 SSE2 and Power VMX instructions.

14.18.35 SNMP

Simple Network Management Protocol is a popular protocol for network management. It is used for collecting information from, and configuring, network devices, such as servers, printers, hubs, switches, and routers on an Internet Protocol (IP) network.

14.18.36 Thread

Hardware threads are typically exposed through the operating system as independently schedulable sequences of instructions. A hardware thread executes a software thread within a Linux (or other) OS process.

14.18.37 TB

TeraByte. TeraByte is a trillion base 10 bytes. This is typically used in every context except for Random Access Memory size and is 1012 (or 1,000,000,000,000) bytes.

14.18.38 TFLOPS

teraFLOPS. Trillion (1012 = 1,000,000,000,000) 64-bit floating point operations per second.

14.18.39 TiB

tibiByte. tibiByte is 1,099,511,627,776 base 10 bytes (i.e., 10244 bytes).

14.18.40 TLB

Translation Look-aside Buffer (TLB) is a set of content addressable hardware registers on the processor that allows fast translation of virtual memory addresses into real memory addresses for virtual addresses that have an active TLB enetry.

14.18.41 UMA

Uniform Memory Access architecture. The distance in core clocks between core registers and every element of node memory is the same. That is, load/store operations that are serviced by the node memory have the same latency to/from every core, no matter where the target physical location is in the node memory assuming no contention.

14.19 SOFTWARE

14.19.1 32b executable

Executable binaries (user applications) with 32b (4B) virtual memory addressing.

14.19.2 64b executable

Executable binaries (user applications) with 64b (8B) virtual memory addressing.

14.19.3 API

Application Programming Interface: Syntax and semantics for invoking services from within an executing application.

14.19.4 Baseline Languages

The Baseline Languages are Fortran08, C, C++, and Python.

14.19.5 BIOS

Basic Input-Output System (BIOS) is low level (typically assembly language) code usually held in flash memory on the node that tests and functions the hardware upon power-up or reset or reboot and loads the operating system.

14.19.6 BOS

Base Operating System (BOS). Linux (LSB 3.1) compliant Operating System run on the FEN.

14.19.7 CDTI

The hierarchal Code Development Tools Infrastructure (CDTI) components are distributed throughout the CORAL system. Individual code development tool “front-end” components that interact with the user execute on the FEN (although the display may be remoted via an X-Window). Code development tool communications mechanisms interface the tool “front ends” running on the FEN with the user application running on the CN through a single level fan-out hierarchy.

14.19.8 CNOS

Light-Weight Kernel providing operating system functions to user applications running on CN.

14.19.9 Current standard

Term applied when an API is not “frozen” on a particular version of a standard but will be upgraded automatically by Offeror as new specifications are released.

14.19.10 Fully supported

A software product-quality implementation, documented and maintained by the HPC machine supplier or an affiliated software supplier.

14.19.11 Job

An allocation of resources to a user for a specified period of time. The user should be given control over which resources can be allocated to a job.

14.19.12 OS

Operating System.

14.19.13 Published (as applied to APIs):

Where an API is not required to be consistent across platforms, the capability lists it as “published,” referring to the fact that it will be documented and supported, although it will be Offeror- or even platform-specific.

14.19.14 RPCTI

Remote process control code development tools interface that allows code development tools to interface from the FEN to the CNOS on the CN and operate on user processes and threads on the CN.

14.19.15 Single-point control

Refers to the ability to control or acquire information on all processes/PEs using a single command or operation.

14.19.16 Standard (as applied to APIs)

Where an API is required to be consistent across platforms, the reference standard is named as part of the capability.

14.19.17 Task

A process launched as a job step component, typically an MPI process.

15 APPENDIX A: WORKFLOWS CONTEXT

The OLCF-6 system will run leadership-class scientific campaigns that are among the largest-scale open science computational runs conceivable. These campaigns will both generate large amounts of data and incorporate data from observational sources external to the facility. The computational campaigns may also involve bidirectional interaction with an external facility (e.g., for data and control or model updates). Each of these campaigns consists of a series of phases of interactions between the user, the software executing on the computing, networking, and data components, the orchestration software and systems, and data generated and ingested. These "phases of the campaign" that come together aimed at a goal are commonly called workflows. Leadership-scale workflows include phases of HPC simulation, high-bandwidth communication, interleaved AI training with simulation and inferencing, and data processing. The user and automation processes that direct the computational platform may include additional workflow steps of data movement and streaming, preprocessing, computation staging, and data postprocessing, visualization, transfer, and archiving. As the campaigns begin operating across more than one facility, they take the form of Integrated Research Infrastructure (IRI) [1] (discussed earlier in the introduction to this document) workflows. It is commonly accepted that effecting these workflows require a combination of technical innovations and new policy guidelines [2].

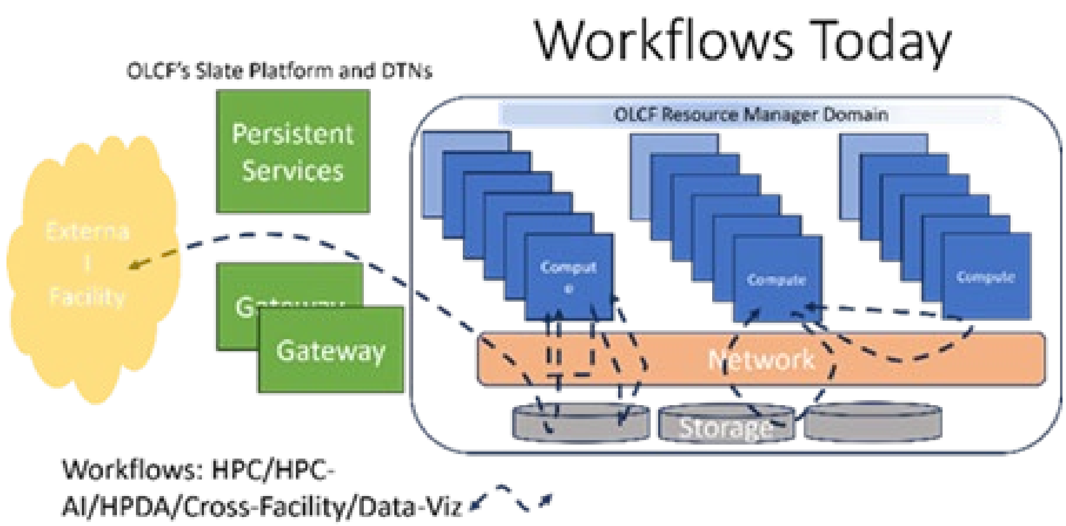

Figure 1: Workflows overlaid on supercomputers

Today, much of the execution of workflows is arranged in user-space by users who set up orchestration software within their campaign to facilitate workflows. Figure 1 shows sample workflows operating across the platform and making connections outside the supercomputer's perimeter – i.e., outside the OLCF Resource Manager Domain.

Structures and configurations of applications encoded as workflows, their associated challenges and responses have been widely discussed in the literature [e.g., in 3,4,5]. In a recent white paper [6], the NERSC facility outlines a set of candidate workflow archetypes. Workflow campaigns the white paper calls out include high-performance simulation and modeling, high-performance AI (HPAI) workflow, cross-facility workflow, hybrid HPC and HPAI and high-performance data analytics workflow, scientific data lifecycle workflow, and external event-triggered and API-driven workflows. As outlined in Figure 2, we anticipate that OLCF-6's Front End Nodes, network connectivity, AI-optimized storage, and software services target requirements in OLCF-6 will simplify the deployment and execution of such workflows as the OLCF Resource Manager Domain expands to include them.

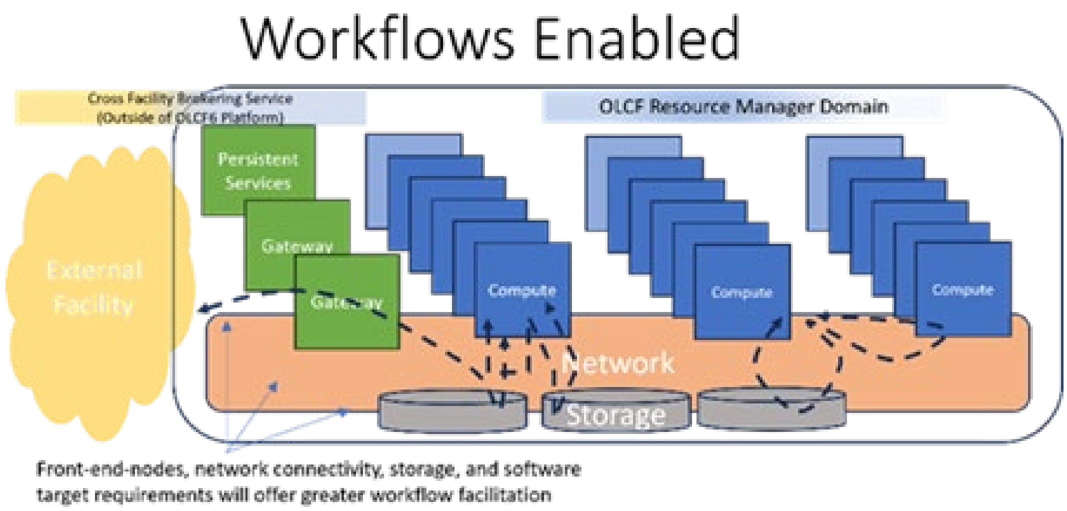

Figure 2: Integrated workflow enablers

References:

1. Miller, William L., Bard, Deborah, Boehnlein, Amber, Fagnan, Kjiersten, Guok, Chin, Lancon, Eric, Ramprakash, Sreeranjani, Shankar, Mallikarjun, Schwarz, Nicholas, and Brown, Benjamin L.. Integrated Research Infrastructure Architecture Blueprint Activity Final Report 2023. United States: N. p., 2023. Web. doi:10.2172/1984466

2. M. Shankar, S. Somnath, S. Alam, D. Feichtinger, L. Sala, J. Wells, Policy Considerations when Federating Facilities for Experimental and Observational Data Analysis, Handbook on Big Data and Machine Learning in the Physical Sciences, May 2020, https://doi.org/10.1142/11389-vol2

3. Ferreira da Silva, R., Filgueira, R., Pietri, I., Jiang, M., Sakellariou, R., & Deelman, E. (2017). A characterization of workflow management systems for extreme-scale applications. Future Generation Computer Systems, 75, 228-238.

4. Suter, F., Ferreira da Silva, R., Gainaru, A., Klasky, S. (2023), Driving Next-Generation Workflows, 19th IEEE Conference on eScience (eScience23).

5. Vescovi, R., Chard, R., Saint, N. D., Blaiszik, B., Pruyne, J., Bicer, T., ... & Foster, I. T. (2022). Linking scientific instruments and computation: Patterns, technologies, and experiences. Patterns, 3(10).

6. Workflows Archetypes White Paper, v1.0, https://www.nersc.gov/assets/NERSC-10/Workflows-Archetypes-White-Paper-v1.0.pdf, Accessed 10th September, 2023

16 APPENDIX B: I/O USE CASES

16.1 I/O

16.1.1 Hero Sequential, Hero Random

System performance verification via Hero Sequential and Hero Random tests.

16.1.2 Single-client

The "single-client" use case is to measure the best performance achievable for an I/O workload running on a single CN. The workload may utilize multiple processes on the CN as necessary to maximize the performance.

16.1.3 Application Checkpoint

The "application checkpoint" use case is to persist 15% of JACNM using up to 90% of the CNs in the system.

16.1.4 Application Restart

The "application restart" use case is to read the data from a recent "application checkpoint" using up to 90% of the CNs in the system.

16.1.5 Application Cold Restart

The "application cold restart" use case is to read the data from an "application checkpoint" that occurred in the past using up to 90% of the CNs in the system.

16.1.6 Application Cold Reboot Restart

The "application cold reboot restart" use case is to read the data from an "application checkpoint" that occurred before a recent complete storage system reboot using up to 90% of the CNs in the system.

16.1.7 Small Random Repeated Read

The "small random repeated read" use case is to use Job-specific Storage to accelerate read-intensive I/O workloads that use repeated random reads from varying subsets of CNs using up to 90% of the CNs in the system.

16.1.8 Object Storage Interface

The "object storage interface" use case is to access large datasets using object-based put/get interfaces similar to cloud storage systems such as Amazon S3.