Reproducible Experimentation Workflows Using Popper

Ivo Jimenez

Research Scientist and CROSS Incubator Fellow

UC Santa Cruz

UNIVERSITY OF CALIFORNIA SANTA CRUZ

SYSTEMS RESEARCH LAB

CROSS CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE

SC16 Explores Reproducibility for Advanced Computing Through Student Cluster Competition by Michela Taufer

March 16, 2016 — Leave a Comment

Data sets and software are important by-products products of research in fields that depend upon data-intensive and high performance computing. But these elements are typically absent when research results are recorded in a journal article or conference proceedings. There is a growing sense in the computational community that this gap needs to be filled if we are to create a stable base of research upon which reliable advances may be built. In short, we need to ensure that computational results are as reproducible as

SC16's SCC Reproducibility Committee

Blog Archive

November 2016 (37)
October 2016 (15)
September 2016 (17)
August 2016 (20)
July 2016 (10)
June 2016 (11)
May 2016 (9)
April 2016 (11)
March 2016 (8)
February 2016 (3)
December 2015 (2)
November 2015 (22)

Artifact Description (AD) and Evaluation (AE) Appendices[1]

A two-page narrative on how to replicate results, including detailed descriptions of:

1. The environment where the experiment originally ran.
2. How to compile and install software.
3. Validation methodology.

[1]: http://ctuning.org/ae/submission_extra.html

Extreme Scale Multi-Physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake

Carsten Uphoff
Sebastian Rettenberger
Michael Bader
uphoff@in.tum.de
rettenbs@in.tum.de
bader@in.tum.de
Technical University of Munich
Boltzmannstr. 3
85748 Garching, Germany

Elizabeth H. Madden
Thomas Ulrich
Stephanie Wollherr
Alice-Agnes Gabriel
madden@geophysik.uni-muenchen.de
ulrich@geophysik.uni-muenchen.de
wollherr@geophysik.uni-muenchen.de
gabriel@geophysik.uni-muenchen.de
Ludwig-Maximilians-Universität München
Theresienstr. 41
80333 Munich, Germany

ABSTRACT

We present a high-resolution simulation of the 2004 Sumatra-Andaman earthquake, including non-linear frictional failure on a megathrust-splay fault system. Our method exploits unstructured meshes capturing the complicated geometries in subduction zones that are crucial to understand large earthquakes and tsunami generation. These up-to-date largest and longest dynamic rupture simulations enable analysis of dynamic source effects on the seafloor displacements.

To tackle the extreme size of this scenario an end-to-end optimization of the simulation code SeisSol was necessary. We implemented a new cache-aware wave propagation scheme and optimized the dynamic rupture kernels using code generation. We established a novel clustered local-time-stepping scheme for dynamic rupture. In total, we achieved a speed-up of 13.6 compared to the previous implementation. For the Sumatra scenario with 221 million elements this reduced the time-to-solution to 13.9 hours on 86,016 Haswell cores. Furthermore, we used asynchronous output to overlap I/O and compute time.

CCS CONCEPTS

• **Applied computing → Earth and atmospheric sciences**;

KEYWORDS

ADER-DG; earthquake simulation; tsunami coupling; dynamic rupture; petascale performance; hybrid parallelization; local time stepping; asynchronous output

ACM Reference format:

Carsten Uphoff, Sebastian Rettenberger, Michael Bader, Elizabeth H. Madden, Thomas Ulrich, Stephanie Wollherr, and Alice-Agnes Gabriel. 2017. Extreme Scale Multi-Physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake. In *Proceedings of SC17, Denver, CO, USA, November 12–17, 2017*, 16 pages.
DOI: 10.1145/3126908.3126948

1 INTRODUCTION

"Megathrust" earthquakes in subduction zones, caused by oceanic lithosphere sliding into the mantle beneath an overriding tectonic plate, have released most of the seismic energy over the last century. No other type of tectonic activity is known to produce earthquakes that exceed moment magnitudes M_w of 9 and trigger teletsunamis traveling whole oceans.

A particularly devastating example is the M_w 9.1 2004 Sumatra-Andaman Earthquake and Indian Ocean Tsunami. The Sumatra earthquake ruptured the greatest fault length of any recorded earthquake and triggered a series of tsunamis, killing up to 280,000 people in 14 countries, and even displaced Earth's North Pole by 2.5 cm.

For the coming decades, there is no realistic hope to reliably predict earthquakes, such that the method of choice to mitigate earthquake- and tsunami-related damage to our societies is to forecast seafloor displacement and strong ground motions for likely earthquake scenarios. Dynamic rupture simulations combine non-linear frictional failure and seismic wave propagation in a multi-physics manner to produce physics-based forecasts [e.g. 5, 20, 30, 38, 65] and provide insight into the poorly understood fundamental processes of earthquake faulting [e.g. 2, 4, 29, 32, 33, 68].

A correct representation of subduction zone complexity is crucial, but, in concurrence with the giant spatial extent and temporal duration to be resolved, extremely challenging from a numerical point of view. Simulations must capture slip and seismic waves occurring for hundreds of seconds along hundreds of kilometers but at the same time resolve the meter-scale physical processes taking place at the rupture tip of the earthquake source.

A recent rise of 3D HPC earthquake simulation software [e.g. 18, 43, 48, 49, 72, 75, 76] allows for the simulation of various aspects of earthquake scenarios and enables researchers to answer geophysical questions complicated by the lack of sufficiently dense

A ARTIFACT DESCRIPTION: EXTREME SCALE MULTI-PHYSICS SIMULATIONS OF THE 2004 SUMATRA MEGATHRUST EARTHQUAKE

A.1 Abstract

This artifact description contains information about the complete workflow required to set up simulations with the Shaking Corals version of SeisSol. We describe how the software can be obtained and the build process as well as necessary preprocessing steps to generate the input dataset for the node level performance measurements. Input datasets for the scaling and production runs are not publicly available due to their size. In addition, the artifact description outlines the complete workflow from the raw input data to the final visualization of the output.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** Arbitrary high-order DERivative Discontinuous Galerkin (ADER-DG) with clustered local time stepping.
- **Program:** SeisSol (www.seissol.org); version: *Shaking Corals*.
- **Compilation:** Intel C/C++ and Fortran Compiler.
- **Binary:** -
- **Data set:** CAD model of Sumatra subduction zone and Bay of Bengal assembled with GoCAD; Mesh generated with the Simulation Modeling Suite from Simmetrix (http://simmetrix.com/); see Section 5.1 for input data sets concerning topography, etc.
- **Run-time environment:** Lenovo NeXtScale nx360M5 WCT with IBM MPI (SuperMUC), Cray XC40 with Cray MPI (Shaheen II and Cori)
- **Hardware:** Optimized code is available for Intel Haswell, Intel Knights Landing and other Intel architectures. An unoptimized fallback version is also available.
- **Output:** Timings from the log file; optional receiver (seismic stations) and visualization output.
- **Experiment workflow:** See below.
- **Publicly available?** Code and example datasets are publicly available. The original input datasets for this paper are available

A.2.5 Datasets. A setup including a mesh with over 3 million elements for the 2004 Sumatra-Andaman earthquake can be obtained from Zenodo https://dx.doi.org/10.5281/zenodo.439946. Due to the large size of the production-run meshes, these are only available upon request.

A.3 Installation

Shaking Corals uses SCons for compilation and supports various options to customize the binary. The following command compiles the release version using optimized Intel Haswell kernels for convergence order 6. The resulting binary will support a hybrid MPI+OpenMP parallelization and netCDF for mesh initialization. Note that netCDF is optional but recommended for large runs.

```
$ scons order=6 compileMode=release \
    generatedKernels=yes arch=dhsw \
    parallelization=hybrid commThread=yes \
    netcdf=yes
```

To get a full list of all available options, run:

```
$ scons --help
```

For a more detailed description, see https://github.com/SeisSol/SeisSol/wiki.

A.4 Experiment workflow

SeisSol requires at least three input files: The file DGPATH, a parameter file, and a mesh file. DGPATH should contain a single text line with the full path to the Maple folder inside the repository.

The parameter file is in Fortran namelist format and contains all parameters required to setup the simulation. Parameters can reference other files, e.g. to specify a list of receiver stations or more complex material parameters. The parameter file for the Sumatra earthquake can be used as a starting point for simulations.

Meshes are usually constructed from CAD models using either the Simulation Modeling Suite from Simmetrix or Gmsh. To convert meshes to the custom netCDF format, the preprocessing tool PUMGen is available on Github https://github.com/TUM-I5/PUML/

History of the SC Reproducibility Initiative

2015

The SC steering committee approved the reproducibility initiative. Authors of SC15 papers were invited to submit an AD Appendix after the conference: one paper did so, became the source for the SC16 Student Cluster Competition Reproducibility Challenge and the first SC paper to display an ACM badge.

2016

Authors submitting to the SC16 conference could optionally submit an AD Appendix: nine authors submitted one, three were finalists, and one was selected to become the source for the SC17 Student Cluster Competition Reproducibility Challenge.

2017

SC made the AD Appendix a requirement to be considered for the Best Paper or Best Student Paper awards. SC17 also introduced the Computational Results Analysis (CRA) Appendix. 40% of submitted and 50% of accepted papers included an AD appendix; nine submitted papers (six accepted) included a CRA Appendix.

2018

SC extended the option of submitting AD Appendices to Workshops and Posters. The CRA Appendix was renamed Artifact Evaluation (AE) Appendix, and limited to four pages. AD Appendices were limited to 2 pages and remained optional (but required for consideration as Best Paper/Best Student Paper, and also Best Poster/Best Student Poster).

2019

AD Appendices will be mandatory for all submissions. AE Appendices are still optional, and both will be submitted via a standard form in the conference submission system. Three new Technical Program tracks, with their respective committees and chairs, are introduced in support of the SC Reproducibility Initiative.

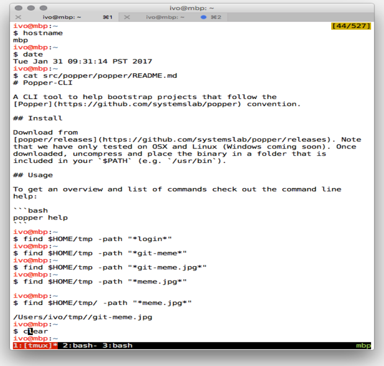
AD/AE Appendix Review Process and Criteria

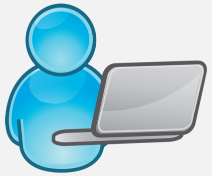
- Each appendix gets reviewed (manually inspected).
- For SC19[1]
  - "Author created artifacts are the hardware, software, or data created by the paper's authors. Only these artifacts need be made available to facilitate reproducibility. Proprietary, closed source artifacts (e.g. commercial software and CPUs) will necessarily be part of many research studies. These proprietary artifacts should be described to the best of the author's ability but do not need to be provided.".

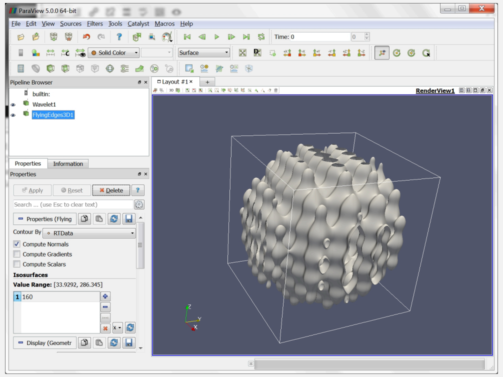
[1]: https://sc19.supercomputing.org/submit/reproducibility-initiative/

How is ***usually*** an AD Appendix Prepared?

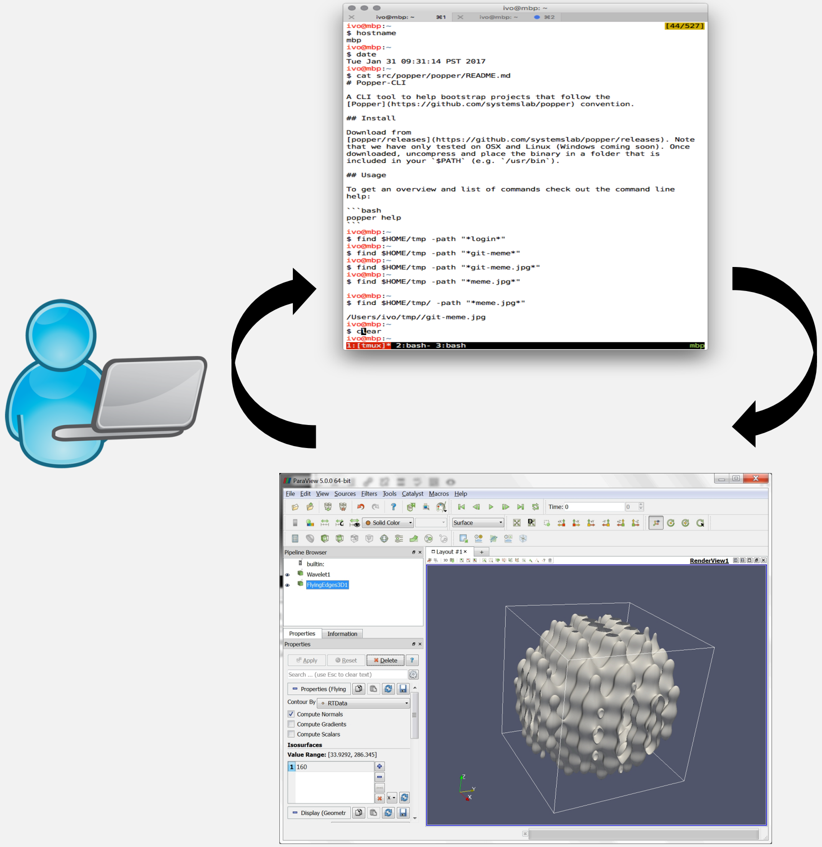
How is *usually* an AD Appendix Prepared?



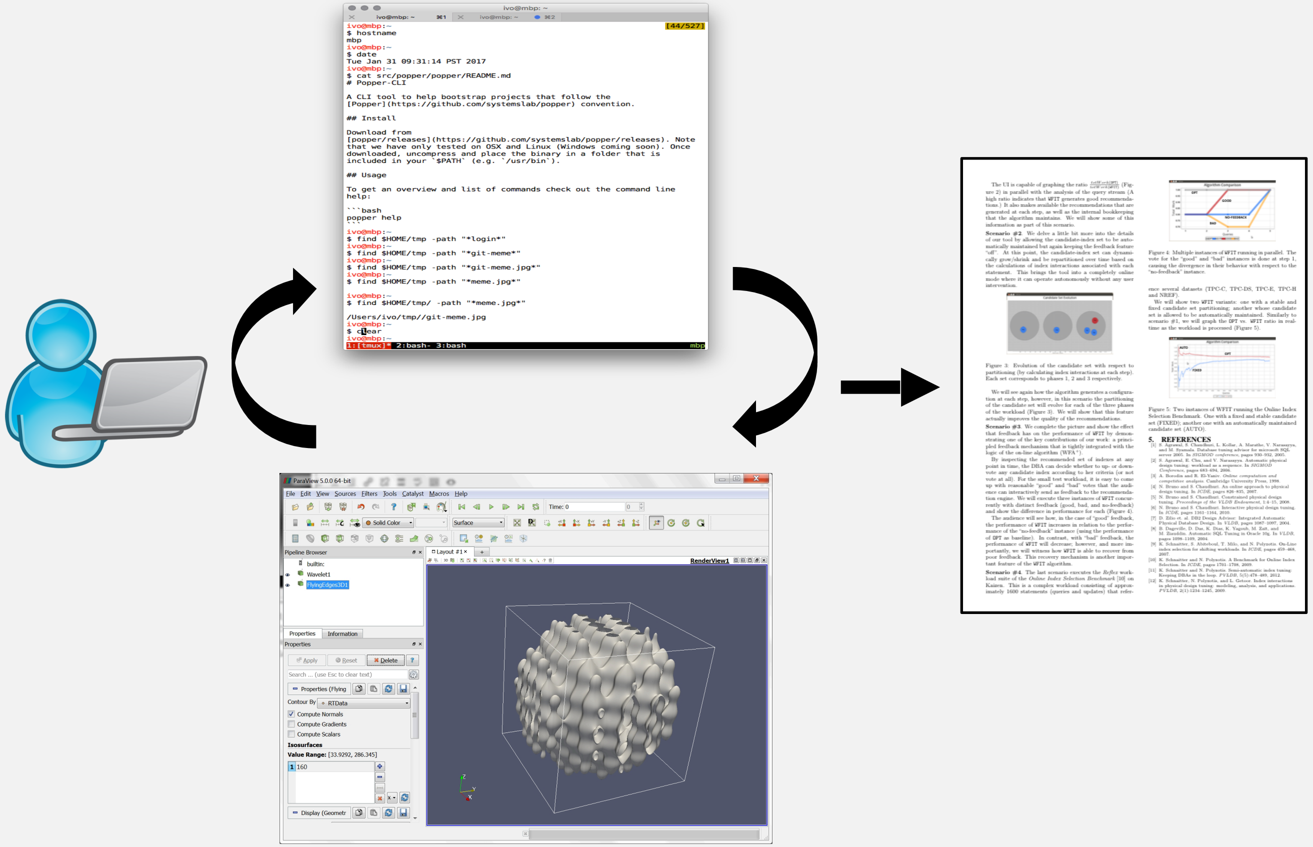




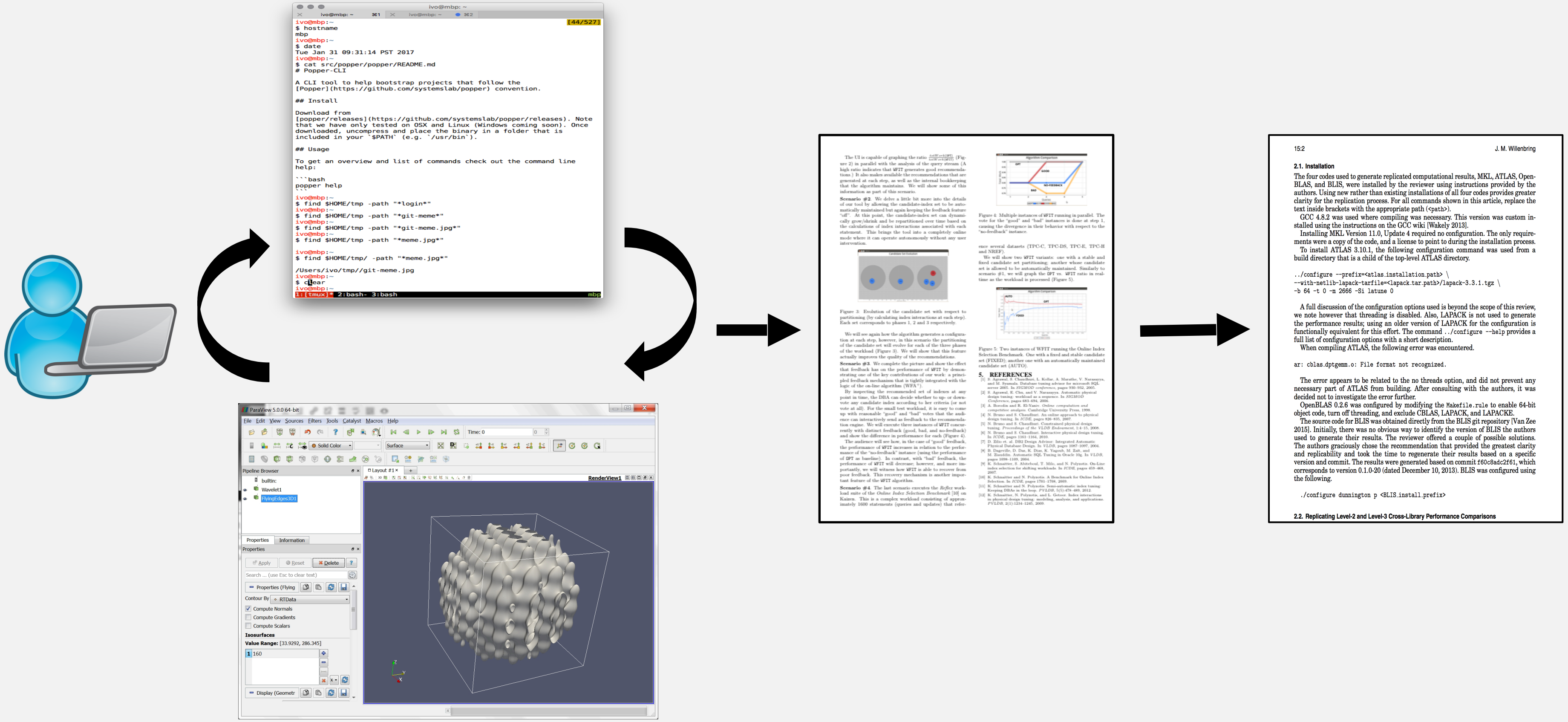
How is *usually* an AD Appendix Prepared?



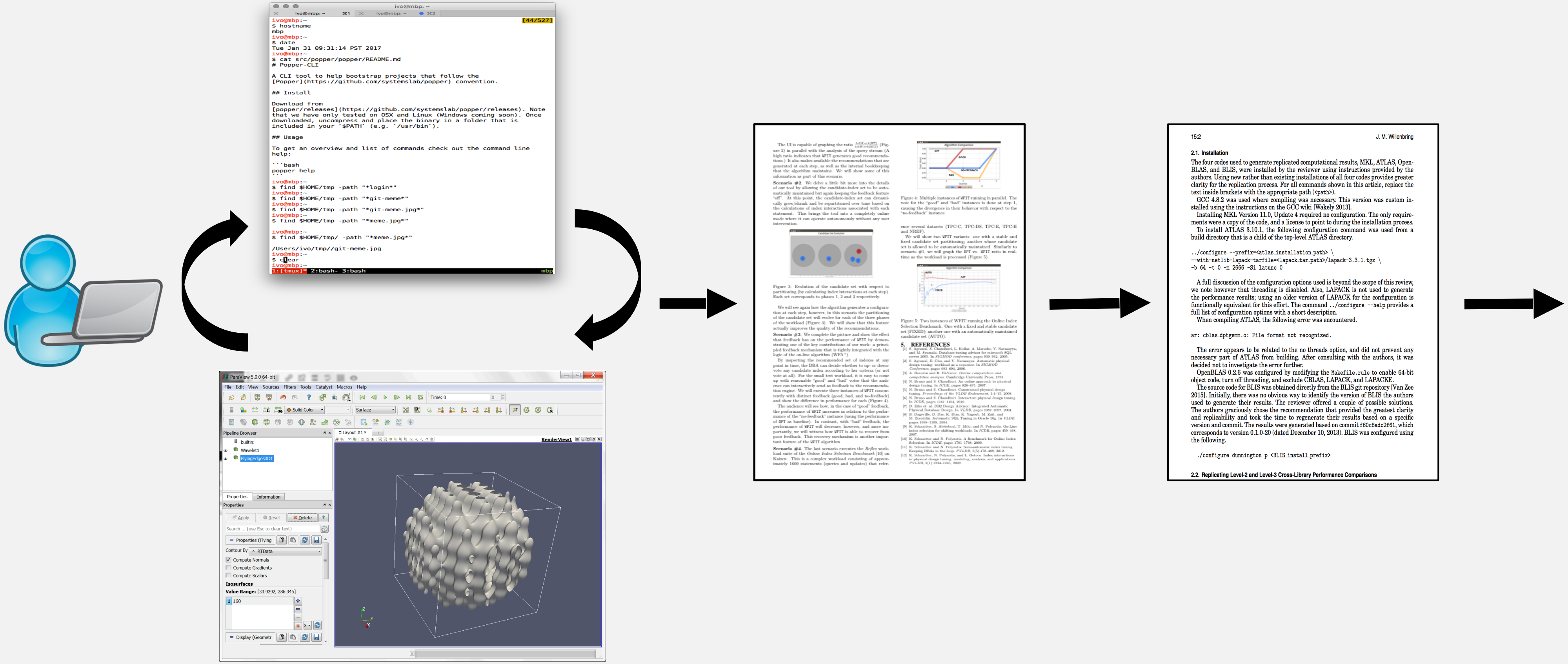
How is *usually* an AD Appendix Prepared?



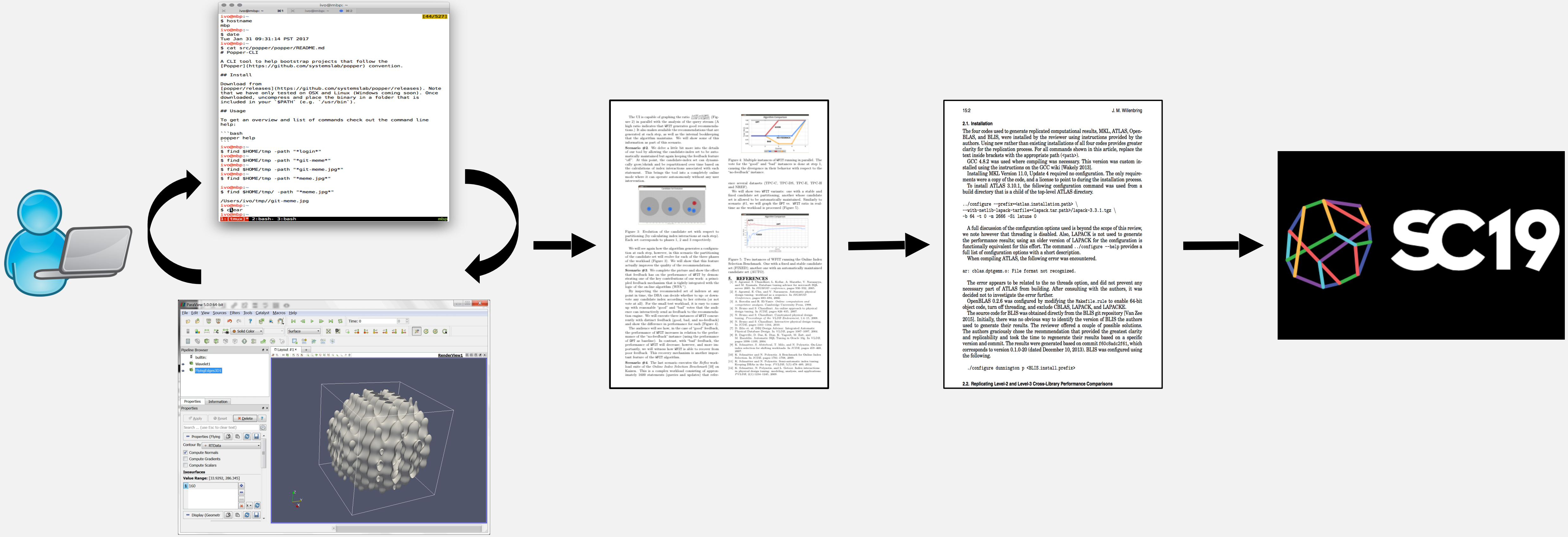
How is *usually* an AD Appendix Prepared?



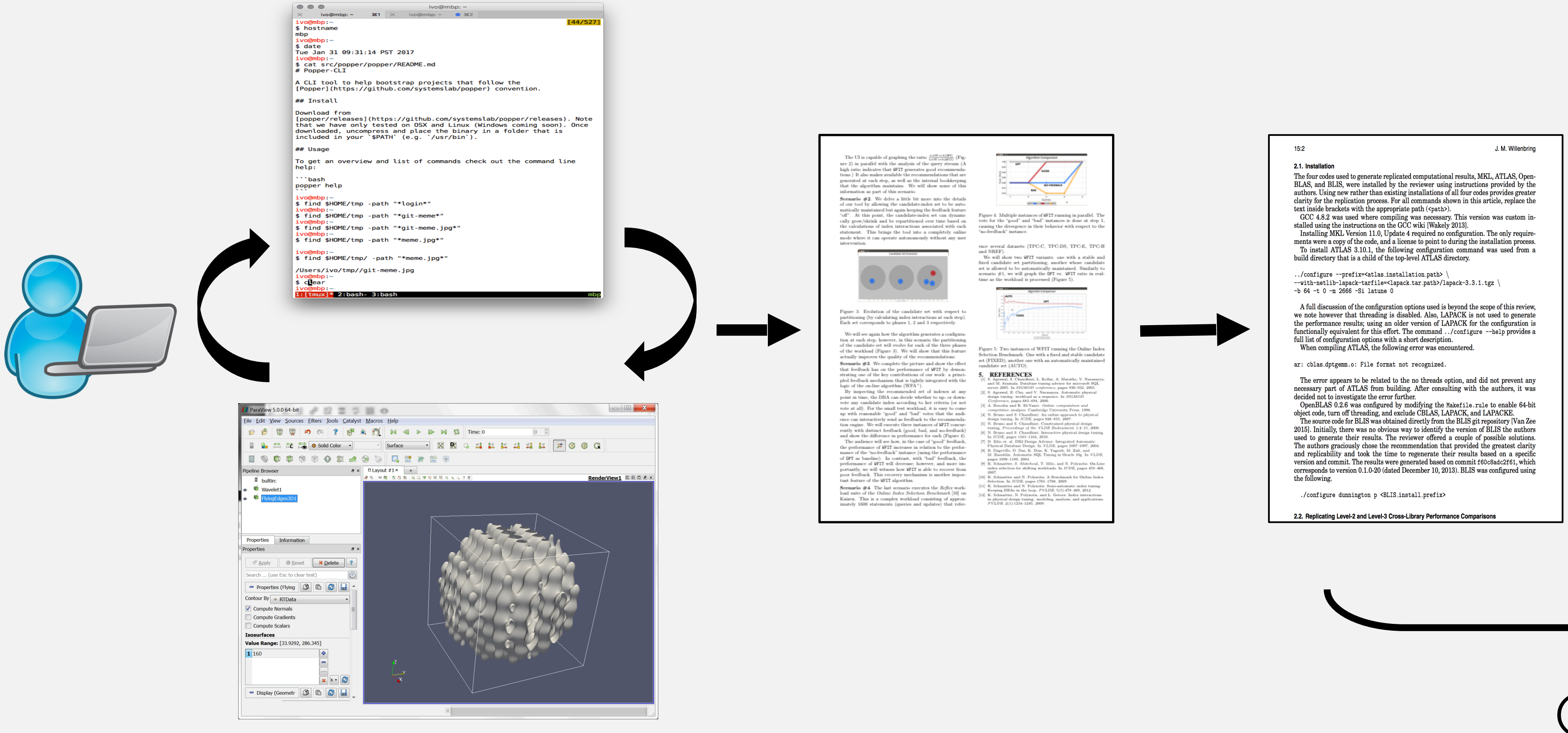
How is *usually* an AD Appendix Prepared?



How is *usually* an AD Appendix Prepared?



How is *usually* an AD Appendix Prepared?

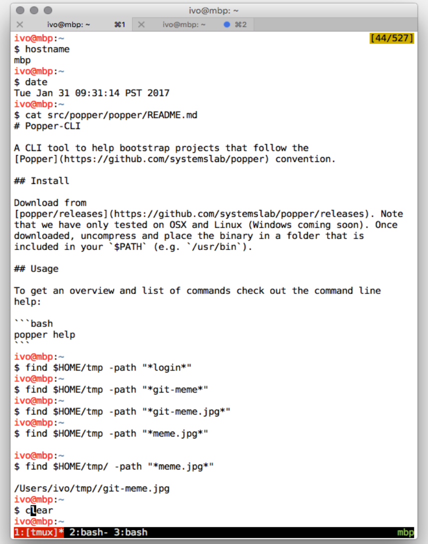


On a rush!

A Deadline-friendly Approach

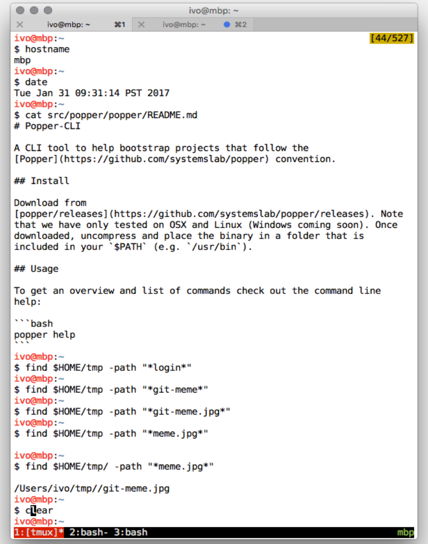
A Deadline-friendly Approach

Typical



A Deadline-friendly Approach

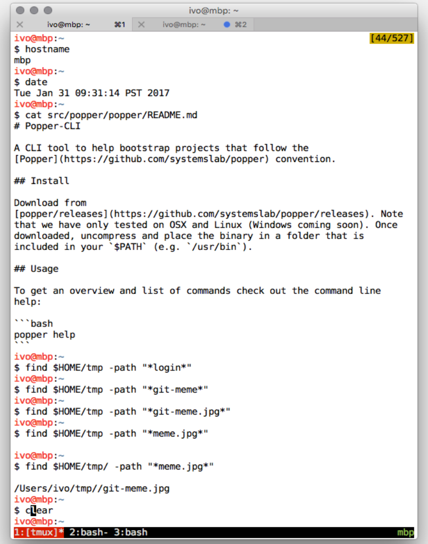
Typical



DevOps

A Deadline-friendly Approach

Typical



```
ivo@mbp: ~
ivo@mbp: ~    ⌘1    ivo@mbp: ~    ⌘2
ivo@mbp:~                                              [44/527]
$ hostname
mbp
ivo@mbp:~
$ date
Tue Jan 31 09:31:14 PST 2017
ivo@mbp:~
$ cat src/popper/popper/README.md
# Popper-CLI

A CLI tool to help bootstrap projects that follow the
[Popper](https://github.com/systemslab/popper) convention.

## Install

Download from
[popper/releases](https://github.com/systemslab/popper/releases). Note
that we have only tested on OSX and Linux (Windows coming soon). Once
downloaded, uncompress and place the binary in a folder that is
included in your `$PATH` (e.g. `/usr/bin`).

## Usage

To get an overview and list of commands check out the command line
help:

```bash
popper help
```
ivo@mbp:~
$ find $HOME/tmp -path "*login*"
ivo@mbp:~
$ find $HOME/tmp -path "*git-meme*"
ivo@mbp:~
$ find $HOME/tmp -path "*git-meme.jpg*"
ivo@mbp:~
$ find $HOME/tmp -path "*meme.jpg*"

ivo@mbp:~
$ find $HOME/tmp/ -path "*meme.jpg*"

/Users/ivo/tmp//git-meme.jpg
ivo@mbp:~
$ clear
ivo@mbp:~
1:[tmux]* 2:bash- 3:bash   mbp
```

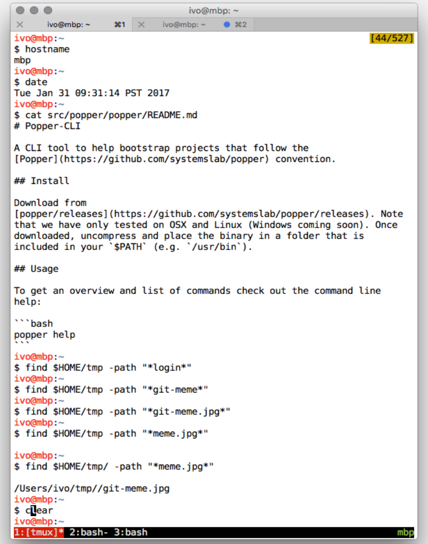
DevOps

myscript.sh

`$ bash myscript.sh`

A Deadline-friendly Approach

Typical



DevOps

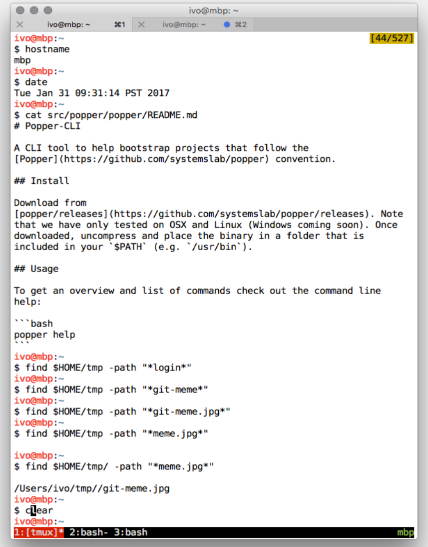
myscript.sh

```
$ bash myscript.sh
```

- Automation
  - Bash, Python, etc.

A Deadline-friendly Approach

Typical



DevOps

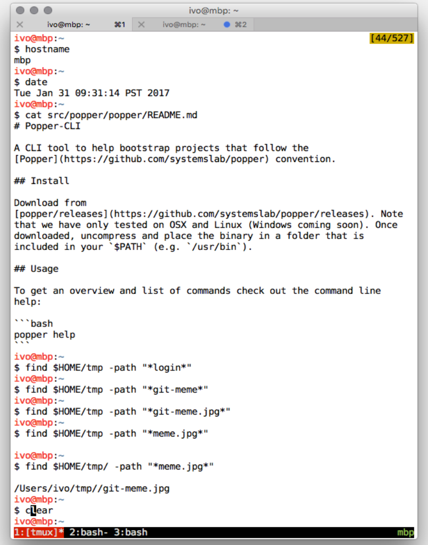
myscript.sh

```
$ bash myscript.sh
```

- Automation
  - Bash, Python, etc.
- Version-control
  - Git, Mercurial

A Deadline-friendly Approach

Typical



DevOps

myscript.sh

`$ bash myscript.sh`

- Automation
  - Bash, Python, etc.
- Version-control
  - Git, Mercurial
- Portability
  - Containers, Spack

Github Actions Workflows

Github Actions Workflows

```
workflow "test-and-deploy" {
  on = "push"
  resolves = "deploy"
}

action "install" {
  uses = "docker://node:11.6.0"
  args = "npm install"
}

action "test" {
  uses = "docker://node:11.6.0"
  args = "npm test"
  needs = ["install"]
}

action "lint" {
  uses = "./.github/actions/jshint"
  needs = ["install"]
}

action "branch-filter" {
  needs = ["test","lint"]
  uses = "actions/bin/filter@master"
  args = "branch master"
}

action "deploy" {
  needs = ["branch-filter"]
  uses = "actions/bin/sh@master"
  args = ["env"]
  secrets = ["PHONY_SECRET"]
  env = {
    PHONY_ENV = "foo"
  }
}
```

Github Actions Workflows

```
workflow "test-and-deploy" {
  on = "push"
  resolves = "deploy"
}

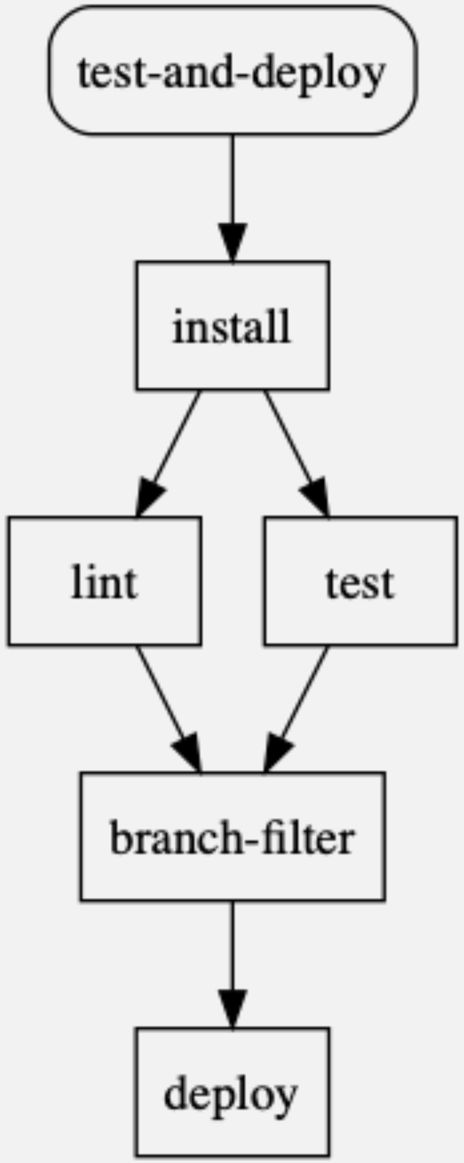
action "install" {
  uses = "docker://node:11.6.0"
  args = "npm install"
}

action "test" {
  uses = "docker://node:11.6.0"
  args = "npm test"
  needs = ["install"]
}

action "lint" {
  uses = "./.github/actions/jshint"
  needs = ["install"]
}

action "branch-filter" {
  needs = ["test","lint"]
  uses = "actions/bin/filter@master"
  args = "branch master"
}

action "deploy" {
  needs = ["branch-filter"]
  uses = "actions/bin/sh@master"
  args = ["env"]
  secrets = ["PHONY_SECRET"]
  env = {
    PHONY_ENV = "foo"
  }
}
```



Github Actions Workflows

Popper
falsifiable.us

```
workflow "test-and-deploy" {
  on = "push"
  resolves = "deploy"
}

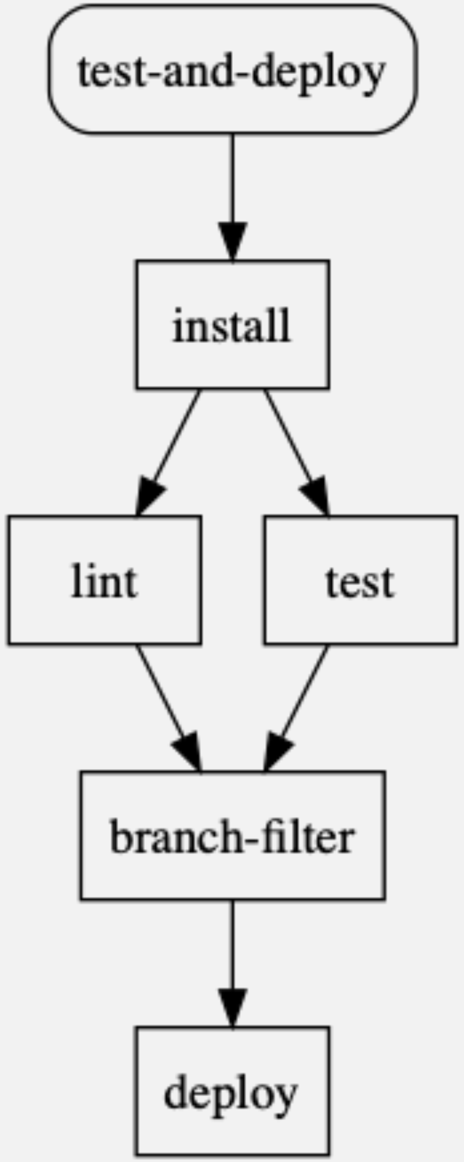
action "install" {
  uses = "docker://node:11.6.0"
  args = "npm install"
}

action "test" {
  uses = "docker://node:11.6.0"
  args = "npm test"
  needs = ["install"]
}

action "lint" {
  uses = "./.github/actions/jshint"
  needs = ["install"]
}

action "branch-filter" {
  needs = ["test","lint"]
  uses = "actions/bin/filter@master"
  args = "branch master"
}

action "deploy" {
  needs = ["branch-filter"]
  uses = "actions/bin/sh@master"
  args = ["env"]
  secrets = ["PHONY_SECRET"]
  env = {
    PHONY_ENV = "foo"
  }
}
```



https://github.com/systemslab/popper

Extreme Scale Multi-Physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake

Carsten Uphoff
Sebastian Rettenberger
Michael Bader
uphoff@in.tum.de
rettenbs@in.tum.de
bader@in.tum.de
Technical University of Munich
Boltzmannstr. 3
85748 Garching, Germany

Elizabeth H. Madden
Thomas Ulrich
Stephanie Wollherr
Alice-Agnes Gabriel
madden@geophysik.uni-muenchen.de
ulrich@geophysik.uni-muenchen.de
wollherr@geophysik.uni-muenchen.de
gabriel@geophysik.uni-muenchen.de
Ludwig-Maximilians-Universität München
Theresienstr. 41
80333 Munich, Germany

ABSTRACT

We present a high-resolution simulation of the 2004 Sumatra-Andaman earthquake, including non-linear frictional failure on a megathrust-splay fault system. Our method exploits unstructured meshes capturing the complicated geometries in subduction zones that are crucial to understand large earthquakes and tsunami generation. These up-to-date largest and longest dynamic rupture simulations enable analysis of dynamic source effects on the seafloor displacements.

To tackle the extreme size of this scenario an end-to-end optimization of the simulation code SeisSol was necessary. We implemented a new cache-aware wave propagation scheme and optimized the dynamic rupture kernels using code generation. We established a novel clustered local-time-stepping scheme for dynamic rupture. In total, we achieved a speed-up of 13.6 compared to the previous implementation. For the Sumatra scenario with 221 million elements this reduced the time-to-solution to 13.9 hours on 86,016 Haswell cores. Furthermore, we used asynchronous output to overlap I/O and compute time.

CCS CONCEPTS

• **Applied computing → Earth and atmospheric sciences**;

KEYWORDS

ADER-DG; earthquake simulation; tsunami coupling; dynamic rupture; petascale performance; hybrid parallelization; local time stepping; asynchronous output

ACM Reference format:
Carsten Uphoff, Sebastian Rettenberger, Michael Bader, Elizabeth H. Madden, Thomas Ulrich, Stephanie Wollherr, and Alice-Agnes Gabriel. 2017. Extreme Scale Multi-Physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake. In *Proceedings of SC17, Denver, CO, USA, November 12–17, 2017,* 16 pages.
DOI: 10.1145/3126908.3126948

DOI: 10.1145/3126908.3126948

1 INTRODUCTION

“Megathrust” earthquakes in subduction zones, caused by oceanic lithosphere sliding into the mantle beneath an overriding tectonic plate, have released most of the seismic energy over the last century. No other type of tectonic activity is known to produce earthquakes that exceed moment magnitudes M_w of 9 and trigger teletsunamis traveling whole oceans.

A particularly devastating example is the M_w 9.1 2004 Sumatra-Andaman Earthquake and Indian Ocean Tsunami. The Sumatra earthquake ruptured the greatest fault length of any recorded earthquake and triggered a series of tsunamis, killing up to 280,000 people in 14 countries, and even displaced Earth's North Pole by 2.5 cm.

For the coming decades, there is no realistic hope to reliably predict earthquakes, such that the method of choice to mitigate earthquake- and tsunami-related damage to our societies is to forecast seafloor displacement and strong ground motions for likely earthquake scenarios. Dynamic rupture simulations combine non-linear frictional failure and seismic wave propagation in a multi-physics manner to produce physics-based forecasts [e.g. 5, 20, 30, 38, 65] and provide insight into the poorly understood fundamental processes of earthquake faulting [e.g. 2, 4, 29, 32, 33, 68].

A correct representation of subduction zone complexity is crucial, but, in concurrence with the giant spatial extent and temporal duration to be resolved, extremely challenging from a numerical point of view. Simulations must capture slip and seismic waves occurring for hundreds of seconds along hundreds of kilometers but at the same time resolve the meter-scale physical processes taking place at the rupture tip of the earthquake source.

A recent rise of 3D HPC earthquake simulation software [e.g. 18, 43, 48, 49, 72, 75, 76] allows for the simulation of various aspects of earthquake scenarios and enables researchers to answer geophysical questions complicated by the lack of sufficiently dense

A ARTIFACT DESCRIPTION: EXTREME SCALE MULTI-PHYSICS SIMULATIONS OF THE 2004 SUMATRA MEGATHRUST EARTHQUAKE

A.1 Abstract

This artifact description contains information about the complete workflow required to set up simulations with the Shaking Corals version of SeisSol. We describe how the software can be obtained and the build process as well as necessary preprocessing steps to generate the input dataset for the node level performance measurements. Input datasets for the scaling and production runs are not publicly available due to their size. In addition, the artifact description outlines the complete workflow from the raw input data to the final visualization of the output.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** Arbitrary high-order DERivative Discontinuous Galerkin (ADER-DG) with clustered local time stepping.
- **Program:** SeisSol (www.seissol.org); version: *Shaking Corals*.
- **Compilation:** Intel C/C++ and Fortran Compiler.
- **Binary:** -
- **Data set:** CAD model of Sumatra subduction zone and Bay of Bengal assembled with GoCAD; Mesh generated with the Simulation Modeling Suite from Simmetrix (http://simmetrix.com/); see Section 5.1 for input data sets concerning topography, etc.
- **Run-time environment:** Lenovo NeXtScale nx360M5 WCT with IBM MPI (SuperMUC), Cray XC40 with Cray MPI (Shaheen II and Cori)
- **Hardware:** Optimized code is available for Intel Haswell, Intel Knights Landing and other Intel architectures. An unoptimized fallback version is also available.
- **Output:** Timings from the log file; optional receiver (seismic stations) and visualization output.
- **Experiment workflow:** See below.
- **Publicly available?** Code and example datasets are publicly available. The original input datasets for this paper are available

A.2.5 Datasets. A setup including a mesh with over 3 million elements for the 2004 Sumatra-Andaman earthquake can be obtained from Zenodo https://dx.doi.org/10.5281/zenodo.439946. Due to the large size of the production-run meshes, these are only available upon request.

A.3 Installation

Shaking Corals uses SCons for compilation and supports various options to customize the binary. The following command compiles the release version using optimized Intel Haswell kernels for convergence order 6. The resulting binary will support a hybrid MPI+OpenMP parallelization and netCDF for mesh initialization. Note that netCDF is optional but recommended for large runs.

```
$ scons order=6 compileMode=release \
      generatedKernels=yes arch=dhsw \
      parallelization=hybrid commThread=yes \
      netcdf=yes
```

To get a full list of all available options, run:

```
$ scons --help
```

For a more detailed description, see https://github.com/SeisSol/SeisSol/wiki.

A.4 Experiment workflow

SeisSol requires at least three input files: The file DGPATH, a parameter file, and a mesh file. DGPATH should contain a single text line with the full path to the Maple folder inside the repository.

The parameter file is in Fortran namelist format and contains all parameters required to setup the simulation. Parameters can reference other files, e.g. to specify a list of receiver stations or more complex material parameters. The parameter file for the Sumatra earthquake can be used as a starting point for simulations.

Meshes are usually constructed from CAD models using either the Simulation Modeling Suite from Simmetrix or Gmsh. To convert meshes to the custom netCDF format, the preprocessing tool PUMGen is available on Github https://github.com/TUM-I5/PUML/

A ARTIFACT DESCRIPTION: EXTREME SCALE MULTI-PHYSICS SIMULATIONS OF THE 2004 SUMATRA MEGATHRUST EARTHQUAKE

A.1 Abstract

This artifact description contains information about the complete workflow required to set up simulations with the Shaking Corals version of SeisSol. We describe how the software can be obtained and the build process as well as necessary preprocessing steps to generate the input dataset for the node level performance measurements. Input datasets for the scaling and production runs are not publicly available due to their size. In addition, the artifact description outlines the complete workflow from the raw input data to the final visualization of the output.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** Arbitrary high-order DERivative Discontinuous Galerkin (ADER-DG) with clustered local time stepping.
- **Program:** SeisSol (www.seissol.org); version: *Shaking Corals*.
- **Compilation:** Intel C/C++ and Fortran Compiler.
- **Binary:** -
- **Data set:** CAD model of Sumatra subduction zone and Bay of Bengal assembled with GoCAD; Mesh generated with the Simulation Modeling Suite from Simmetrix (http://simmetrix.com/); see Section 5.1 for input data sets concerning topography, etc.
- **Run-time environment:** Lenovo NeXtScale nx360M5 WCT with IBM MPI (SuperMUC), Cray XC40 with Cray MPI (Shaheen II and Cori)
- **Hardware:** Optimized code is available for Intel Haswell, Intel Knights Landing and other Intel architectures. An unoptimized fallback version is also available.
- **Output:** Timings from the log file; optional receiver (seismic stations) and visualization output.
- **Experiment workflow:** See below.
- **Publicly available?** Code and example datasets are publicly available. The original input datasets for this paper are available

A.2.5 Datasets. A setup including a mesh with over 3 million elements for the 2004 Sumatra-Andaman earthquake can be obtained from Zenodo https://dx.doi.org/10.5281/zenodo.439946. Due to the large size of the production-run meshes, these are only available upon request.

A.3 Installation

Shaking Corals uses SCons for compilation and supports various options to customize the binary. The following command compiles the release version using optimized Intel Haswell kernels for convergence order 6. The resulting binary will support a hybrid MPI+OpenMP parallelization and netCDF for mesh initialization. Note that netCDF is optional but recommended for large runs.

```
$ scons order=6 compileMode=release \
      generatedKernels=yes arch=dhsw \
      parallelization=hybrid commThread=yes \
      netcdf=yes
```

To get a full list of all available options, run:

```
$ scons --help
```

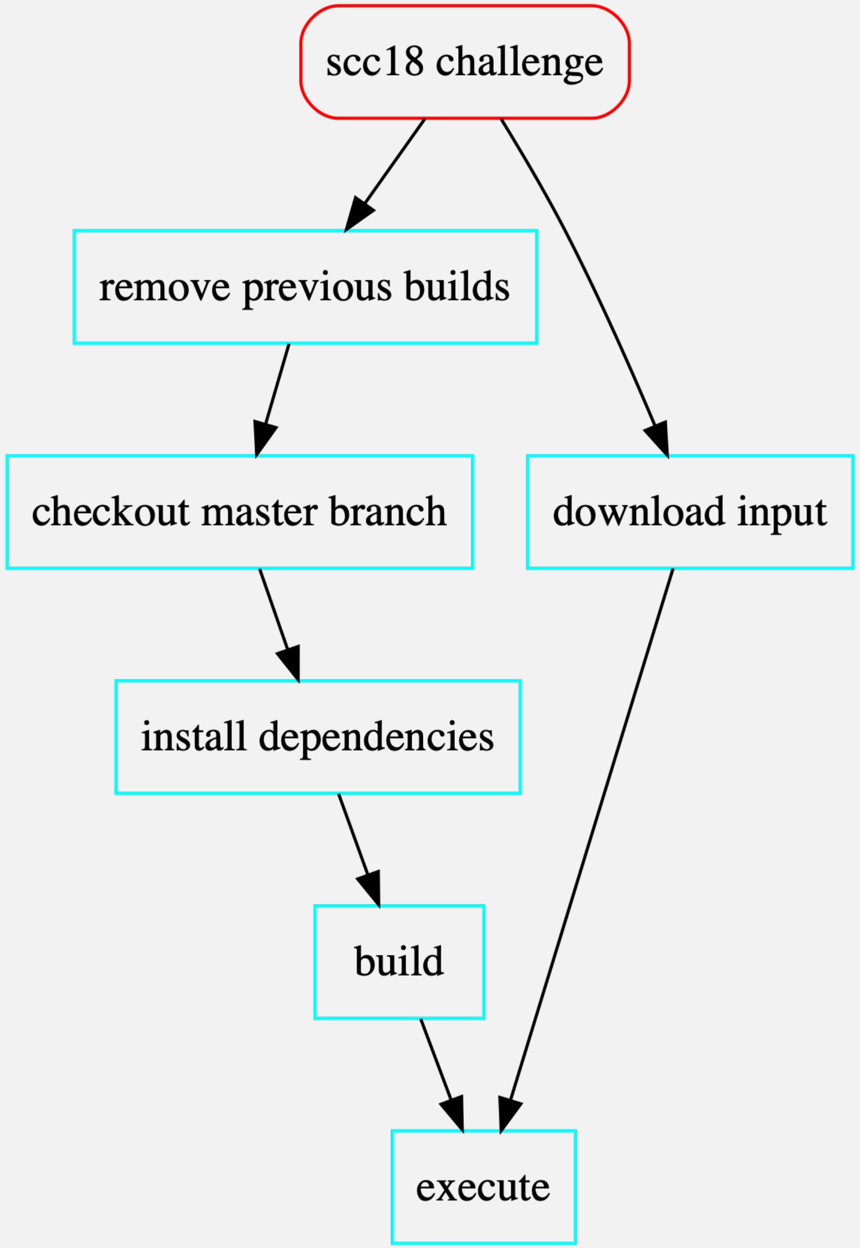
For a more detailed description, see https://github.com/SeisSol/SeisSol/wiki.

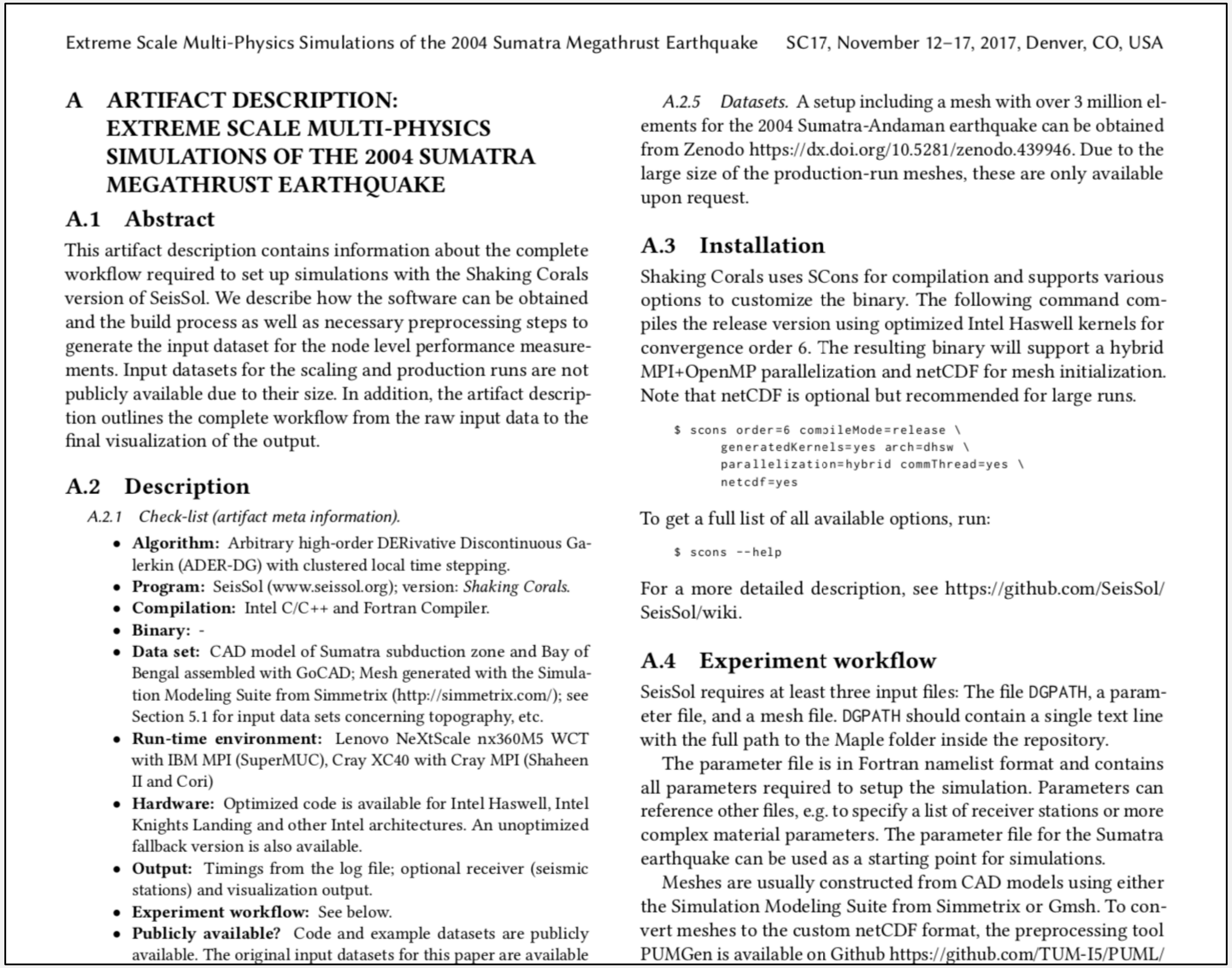
A.4 Experiment workflow

SeisSol requires at least three input files: The file `DGPATH`, a parameter file, and a mesh file. `DGPATH` should contain a single text line with the full path to the Maple folder inside the repository.

The parameter file is in Fortran namelist format and contains all parameters required to setup the simulation. Parameters can reference other files, e.g. to specify a list of receiver stations or more complex material parameters. The parameter file for the Sumatra earthquake can be used as a starting point for simulations.

Meshes are usually constructed from CAD models using either the Simulation Modeling Suite from Simmetrix or Gmsh. To convert meshes to the custom netCDF format, the preprocessing tool PUMGen is available on Github https://github.com/TUM-I5/PUML/





A ARTIFACT DESCRIPTION: EXTREME SCALE MULTI-PHYSICS SIMULATIONS OF THE 2004 SUMATRA MEGATHRUST EARTHQUAKE

A.1 Abstract

This artifact description contains information about the complete workflow required to set up simulations with the Shaking Corals version of SeisSol. We describe how the software can be obtained and the build process as well as necessary preprocessing steps to generate the input dataset for the node level performance measurements. Input datasets for the scaling and production runs are not publicly available due to their size. In addition, the artifact description outlines the complete workflow from the raw input data to the final visualization of the output.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** Arbitrary high-order DERivative Discontinuous Galerkin (ADER-DG) with clustered local time stepping.
- **Program:** SeisSol (www.seissol.org); version: *Shaking Corals.*
- **Compilation:** Intel C/C++ and Fortran Compiler.
- **Binary:** -
- **Data set:** CAD model of Sumatra subduction zone and Bay of Bengal assembled with GoCAD; Mesh generated with the Simulation Modeling Suite from Simmetrix (http://simmetrix.com/); see Section 5.1 for input data sets concerning topography, etc.
- **Run-time environment:** Lenovo NeXtScale nx360M5 WCT with IBM MPI (SuperMUC), Cray XC40 with Cray MPI (Shaheen II and Cori)
- **Hardware:** Optimized code is available for Intel Haswell, Intel Knights Landing and other Intel architectures. An unoptimized fallback version is also available.
- **Output:** Timings from the log file; optional receiver (seismic stations) and visualization output.
- **Experiment workflow:** See below.
- **Publicly available?** Code and example datasets are publicly available. The original input datasets for this paper are available

A.2.5 Datasets. A setup including a mesh with over 3 million elements for the 2004 Sumatra-Andaman earthquake can be obtained from Zenodo https://dx.doi.org/10.5281/zenodo.439946. Due to the large size of the production-run meshes, these are only available upon request.

A.3 Installation

Shaking Corals uses SCons for compilation and supports various options to customize the binary. The following command compiles the release version using optimized Intel Haswell kernels for convergence order 6. The resulting binary will support a hybrid MPI+OpenMP parallelization and netCDF for mesh initialization. Note that netCDF is optional but recommended for large runs.

```
$ scons order=6 compileMode=release \
      generatedKernels=yes arch=dhsw \
      parallelization=hybrid commThread=yes \
      netcdf=yes
```

To get a full list of all available options, run:

```
$ scons --help
```

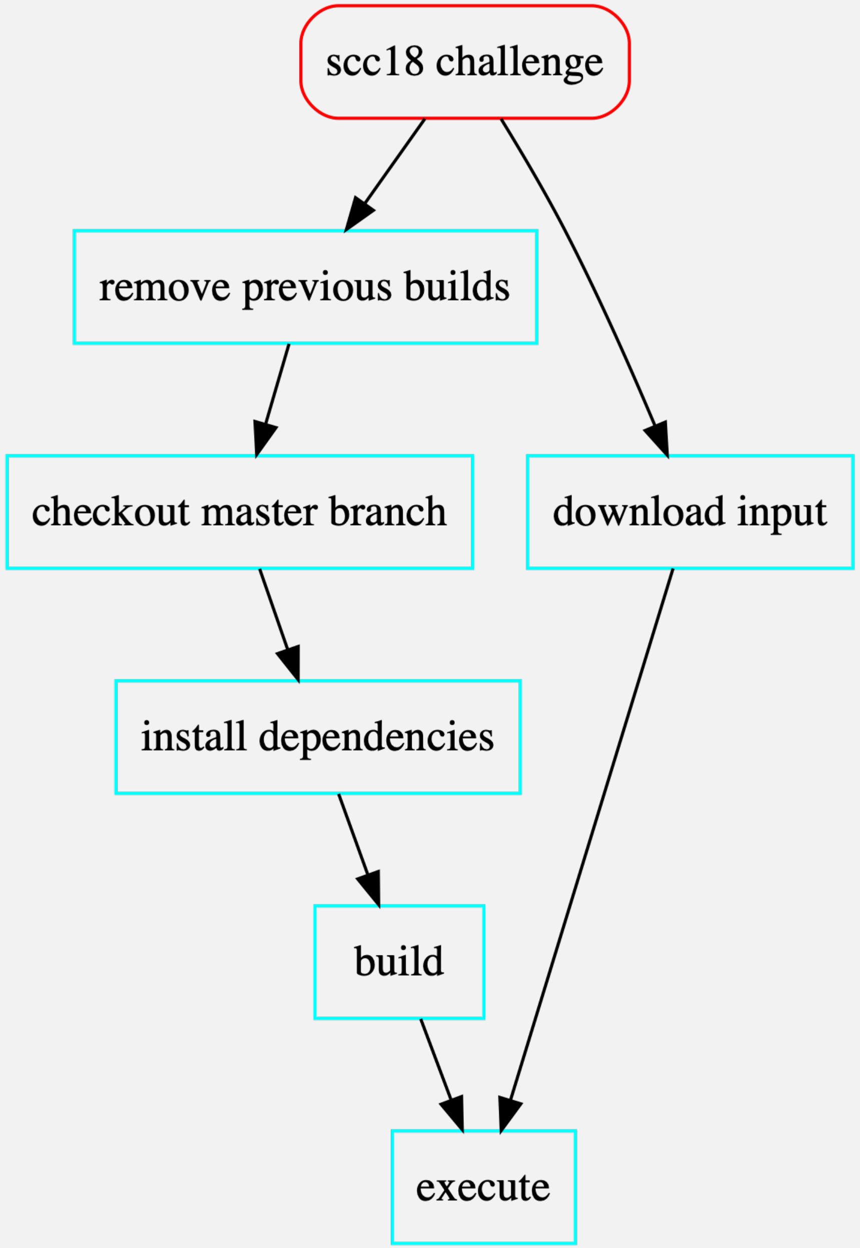
For a more detailed description, see https://github.com/SeisSol/SeisSol/wiki.

A.4 Experiment workflow

SeisSol requires at least three input files: The file `DGPATH`, a parameter file, and a mesh file. `DGPATH` should contain a single text line with the full path to the Maple folder inside the repository.

The parameter file is in Fortran namelist format and contains all parameters required to setup the simulation. Parameters can reference other files, e.g. to specify a list of receiver stations or more complex material parameters. The parameter file for the Sumatra earthquake can be used as a starting point for simulations.

Meshes are usually constructed from CAD models using either the Simulation Modeling Suite from Simmetrix or Gmsh. To convert meshes to the custom netCDF format, the preprocessing tool PUMGen is available on Github https://github.com/TUM-I5/PUML/



A ARTIFACT DESCRIPTION: EXTREME SCALE MULTI-PHYSICS SIMULATIONS OF THE 2004 SUMATRA MEGATHRUST EARTHQUAKE

A.1 Abstract

This artifact description contains information about the complete workflow required to set up simulations with the Shaking Corals version of SeisSol. We describe how the software can be obtained and the build process as well as necessary preprocessing steps to generate the input dataset for the node level performance measurements. Input datasets for the scaling and production runs are not publicly available due to their size. In addition, the artifact description outlines the complete workflow from the raw input data to the final visualization of the output.

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm:** Arbitrary high-order DERivative Discontinuous Galerkin (ADER-DG) with clustered local time stepping.
- **Program:** SeisSol (www.seissol.org); version: *Shaking Corals.*
- **Compilation:** Intel C/C++ and Fortran Compiler.
- **Binary:** -
- **Data set:** CAD model of Sumatra subduction zone and Bay of Bengal assembled with GoCAD; Mesh generated with the Simulation Modeling Suite from Simmetrix (http://simmetrix.com/); see Section 5.1 for input data sets concerning topography, etc.
- **Run-time environment:** Lenovo NeXtScale nx360M5 WCT with IBM MPI (SuperMUC), Cray XC40 with Cray MPI (Shaheen II and Cori)
- **Hardware:** Optimized code is available for Intel Haswell, Intel Knights Landing and other Intel architectures. An unoptimized fallback version is also available.
- **Output:** Timings from the log file; optional receiver (seismic stations) and visualization output.
- **Experiment workflow:** See below.
- **Publicly available?** Code and example datasets are publicly available. The original input datasets for this paper are available

A.2.5 Datasets. A setup including a mesh with over 3 million elements for the 2004 Sumatra-Andaman earthquake can be obtained from Zenodo https://dx.doi.org/10.5281/zenodo.439946. Due to the large size of the production-run meshes, these are only available upon request.

A.3 Installation

Shaking Corals uses SCons for compilation and supports various options to customize the binary. The following command compiles the release version using optimized Intel Haswell kernels for convergence order 6. The resulting binary will support a hybrid MPI+OpenMP parallelization and netCDF for mesh initialization. Note that netCDF is optional but recommended for large runs.

```
$ scons order=6 compileMode=release \
      generatedKernels=yes arch=dhsw \
      parallelization=hybrid commThread=yes \
      netcdf=yes
```

To get a full list of all available options, run:

```
$ scons --help
```

For a more detailed description, see https://github.com/SeisSol/SeisSol/wiki.

A.4 Experiment workflow

SeisSol requires at least three input files: The file DGPATH, a parameter file, and a mesh file. DGPATH should contain a single text line with the full path to the Maple folder inside the repository.

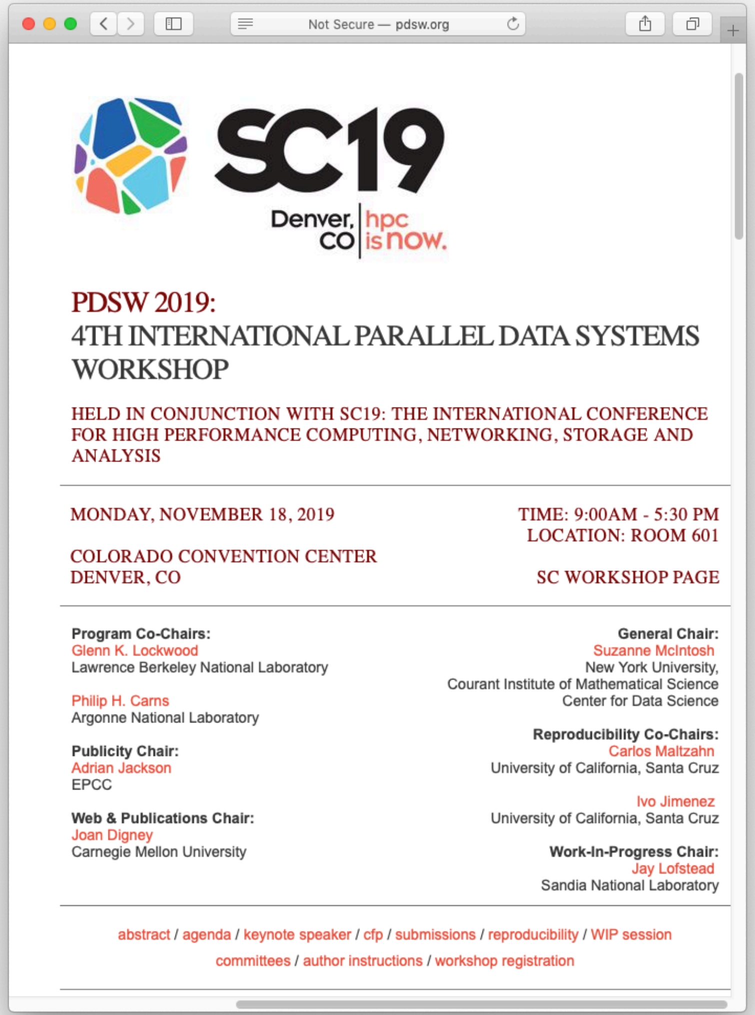
The parameter file is in Fortran namelist format and contains all parameters required to setup the simulation. Parameters can reference other files, e.g. to specify a list of receiver stations or more complex material parameters. The parameter file for the Sumatra earthquake can be used as a starting point for simulations.

Meshes are usually constructed from CAD models using either the Simulation Modeling Suite from Simmetrix or Gmsh. To convert meshes to the custom netCDF format, the preprocessing tool PUMGen is available on Github https://github.com/TUM-I5/PUML/

SCC18: https://github.com/popperized/seissol-workflows
SCC19: https://github.com/popperized/normalmodes-workflows

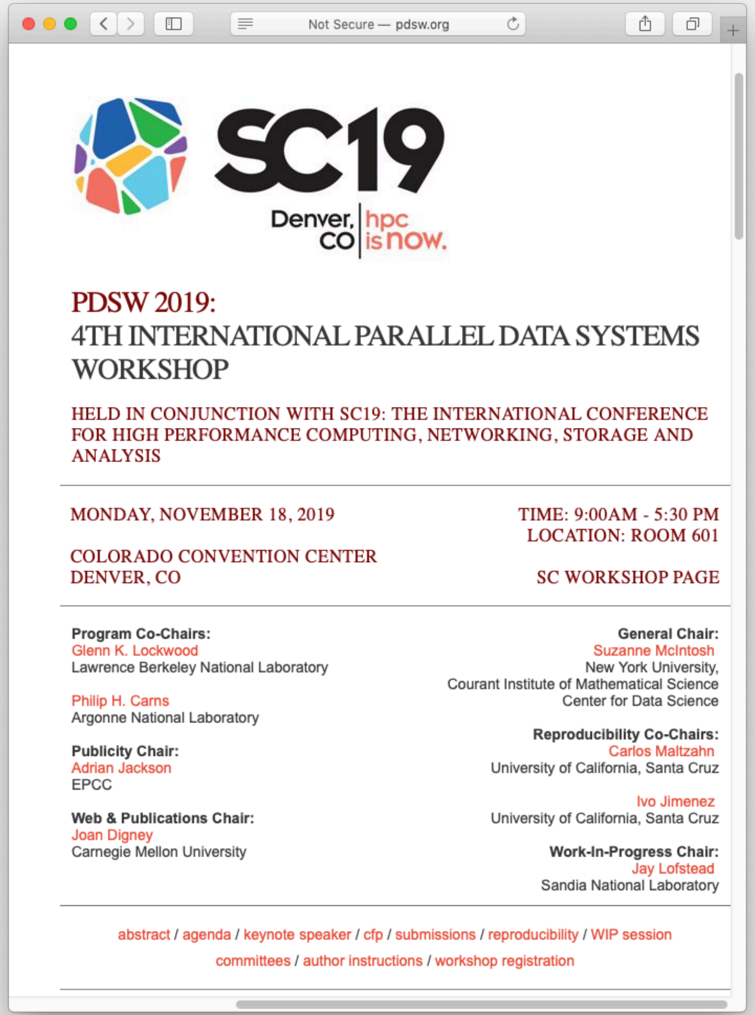
Automated Artifact Evaluation

https://bit.ly/pdsw-automatic



Automated Artifact Evaluation

https://bit.ly/pdsw-automatic


Not Secure — pdsw.org

SC19
Denver, CO
hpc is now.

PDSW 2019:
4TH INTERNATIONAL PARALLEL DATA SYSTEMS WORKSHOP

HELD IN CONJUNCTION WITH SC19: THE INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS

MONDAY, NOVEMBER 18, 2019
COLORADO CONVENTION CENTER
DENVER, CO

TIME: 9:00AM - 5:30 PM
LOCATION: ROOM 601
SC WORKSHOP PAGE

Program Co-Chairs:
Glenn K. Lockwood
Lawrence Berkeley National Laboratory

Philip H. Carns
Argonne National Laboratory

Publicity Chair:
Adrian Jackson
EPCC

Web & Publications Chair:
Joan Digney
Carnegie Mellon University

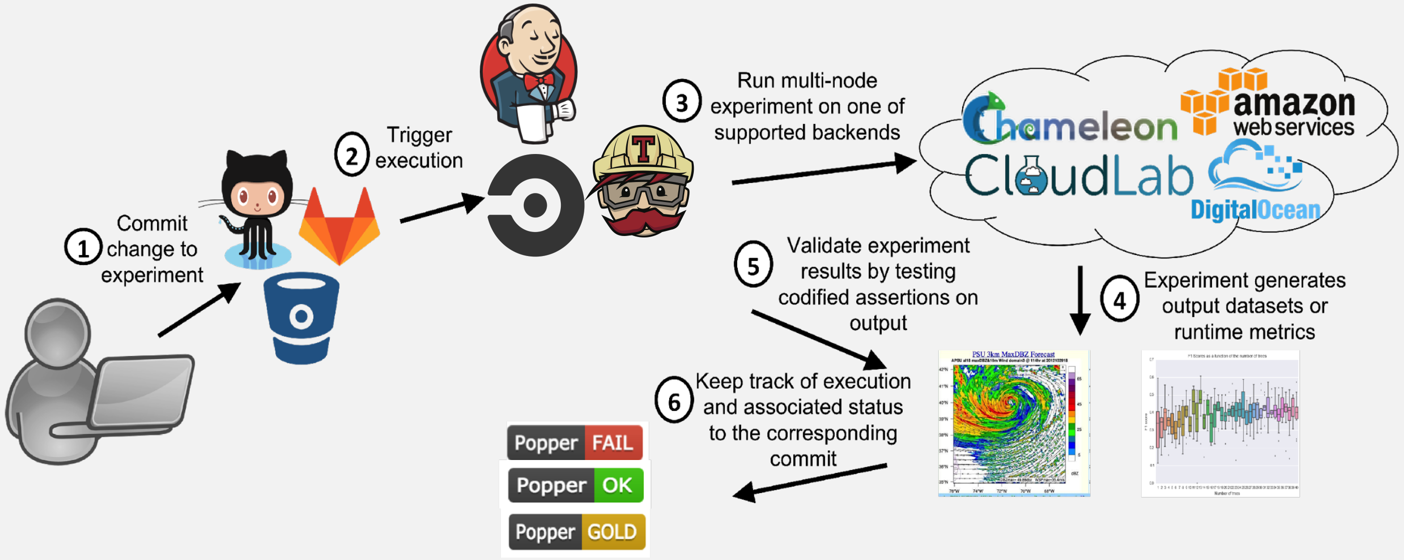
General Chair:
Suzanne McIntosh
New York University,
Courant Institute of Mathematical Science
Center for Data Science

Reproducibility Co-Chairs:
Carlos Maltzahn
University of California, Santa Cruz

Ivo Jimenez
University of California, Santa Cruz

Work-In-Progress Chair:
Jay Lofstead
Sandia National Laboratory

abstract / agenda / keynote speaker / cfp / submissions / reproducibility / WIP session
committees / author instructions / workshop registration



Better Scientific Software (BSSw)

better scientific software

https://bssw.io

Better Scientific Software (BSSw)

better scientific software

https://bssw.io

- BSSw has supported my work via a fellowship (2018).

Popper
falsifiable.us

Better Scientific Software (BSSw)

better scientific software

https://bssw.io

- BSSw has supported my work via a fellowship (2018).
- Create online training materials for teaching Popper
  - https://popperized.github.io/swc-lesson

Popper
falsifiable.us

Better Scientific Software (BSSw)

better scientific software

https://bssw.io

- BSSw has supported my work via a fellowship (2018).
- Create online training materials for teaching Popper
  - https://popperized.github.io/swc-lesson
- Organize one-day hands-on tutorials.

Popper
falsifiable.us

Better Scientific Software (BSSw)

https://bssw.io

- BSSw has supported my work via a fellowship (2018).
- Create online training materials for teaching Popper
  - https://popperized.github.io/swc-lesson
- Organize one-day hands-on tutorials.
- More information at http://falsifiable.us

Challenges

- Information Management
  - Code/Data Release Processes Will Be Stressed.
- Resources for Testing
  - Can we justify the allocation of resources for testing the reproducibility of an experiment? Even if it takes weeks?
- Cultural Issues
  - New methodology. New paradigm for writing papers.
  - New tools. Spack, Containers, GitLab-CI, etc.

Current Status

- CROSS CENTER FOR RESEARCH IN OPEN SOURCE SOFTWARE Incubator Fellowship.
  - Building a self-sustainable model for the project.
- Projects:
  - Ceph Storage Platform @ RedHat
  - Genomics Institute @ UCSC
  - ECP Workflows @ DOE
  - SCC19 @ SC19

Thanks!

Popper

falsifiable.us