SCEC Collaboratory for Interseismic Simulation and Modeling (CISM) Infrastructure

Philip J. Maechling (SCEC)

September 13, 2015

CISM Development Plan

- **System Requirements**
- System Architecture
- Computing Environment
- Essential Software Components
- Computational and Data Estimates
- Software Development Process
- IP Considerations

Use Cases in CISM Proposal

Year 1:

- Couple the empirical Uniform California Earthquake Rupture Forecast to the CyberShake ground-motion forecasting models of the Los Angeles region.
- Provide new computational tools to assist the development of rupture simulators such as RSQSim and ground-motion simulators such as CyberShake.

Year 2:

- Couple the RSQSim physics-based rupture simulator to the CyberShake ground-motion forecasting models
- Retrospectively calibrate and test the resulting comprehensive forecasting models.

Year 3:

- Construct a computational environment that can sustain the long-term development of comprehensive, physics-based earthquake forecasting models
- Submitted exemplars to CSEP for prospective testing against observed earthquake activity in California.

Additional CISM System Requirements

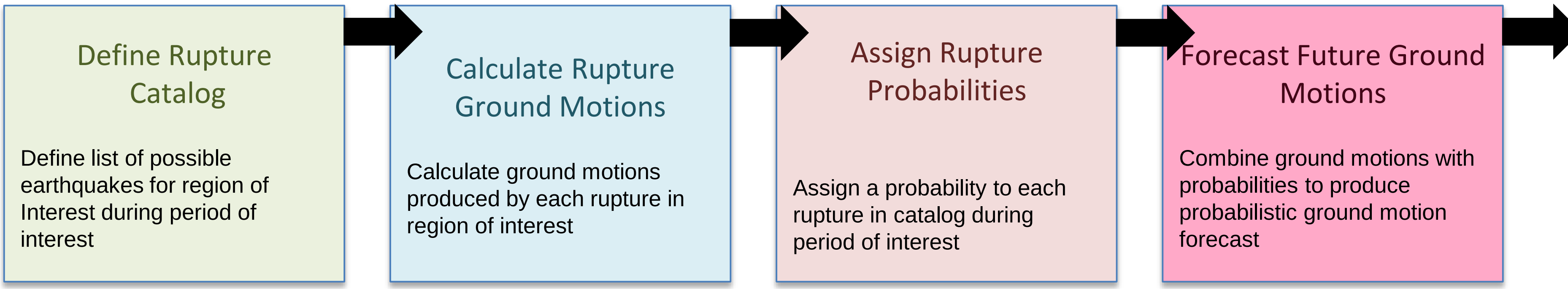
CISM designed to meet several non-functional requirements:

1. Use existing scientific software written in a variety of programming languages
2. Use local computing resources and high-performance parallel computing resources from external resource providers
3. Be able to “show our work” to support scientific review of results.
4. Be inexpensive to design, build, maintain, and operate
5. Be easy to modify without significant re-implementation or down time.
6. Support new development without impacting ongoing operations
7. Run for years to get statistically significant results

CISM Development Plan

- System Requirements
- **System Architecture**
- Computing Environment
- Essential Software Components
- Computational and Data Estimates
- Software Development Process
- IP Considerations

CISM Modular Processing Architecture

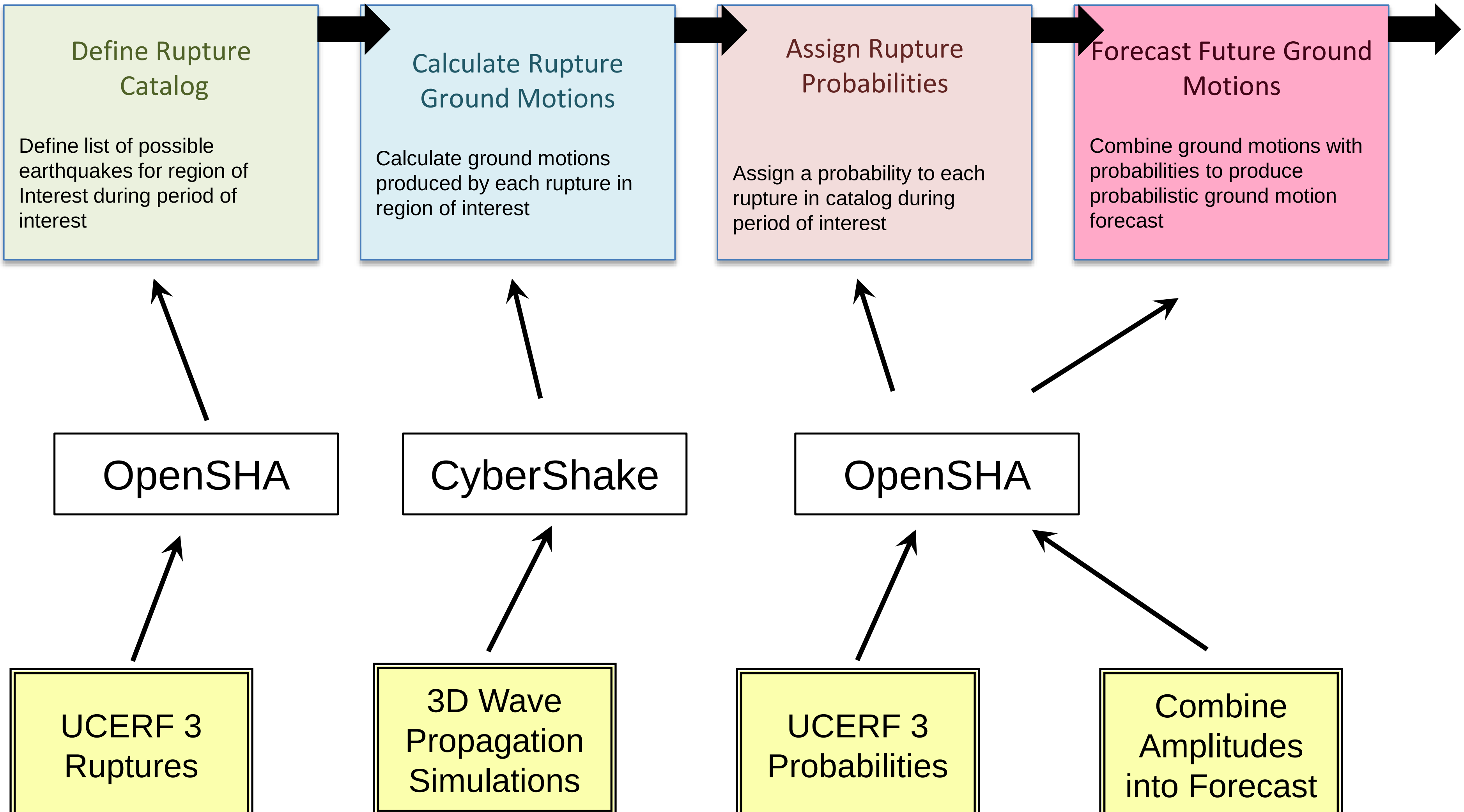


Build a modular, extensible, distributed, high performance computing framework:

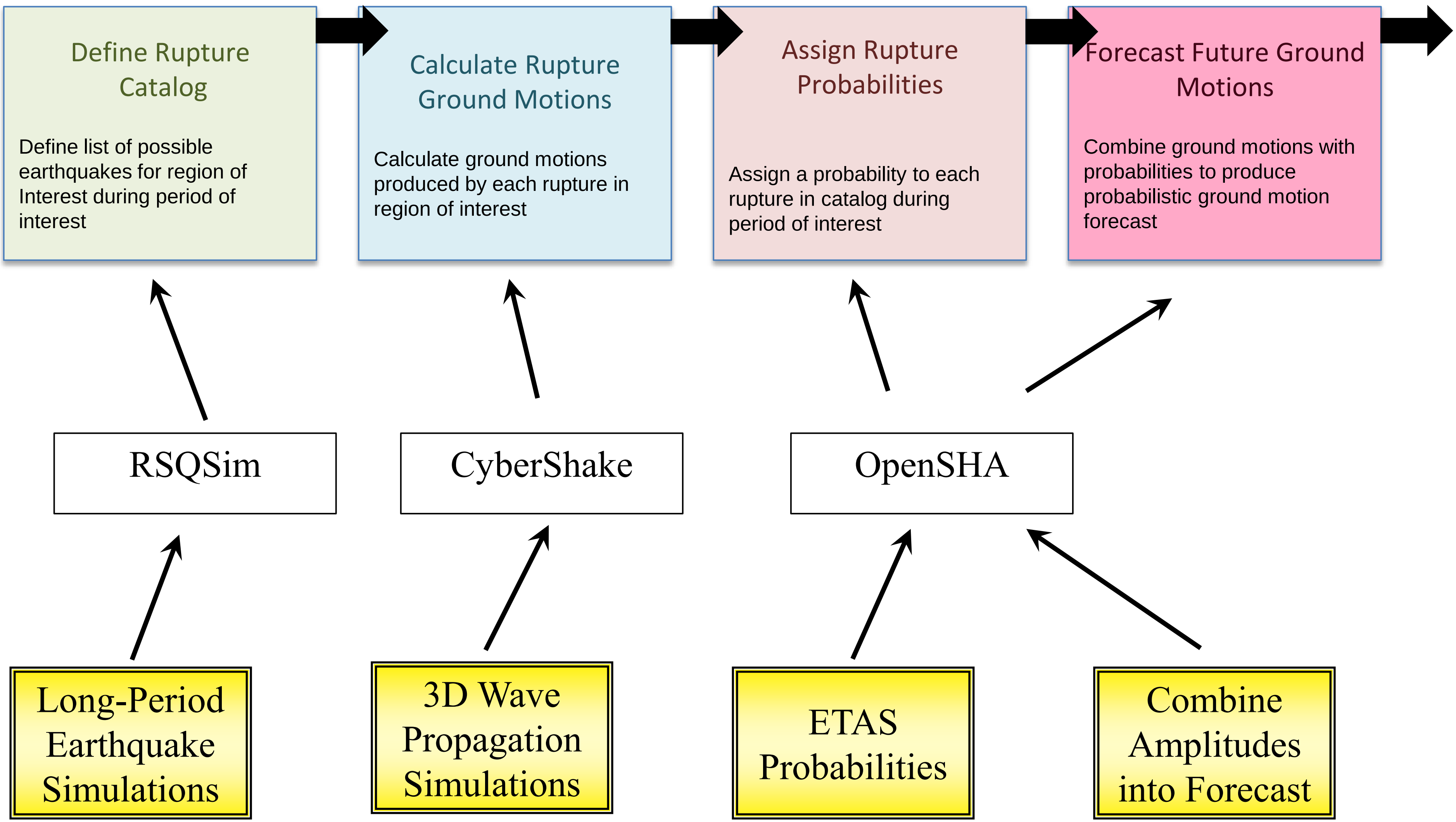
1. Define and execute a multi-stage series of scientific calculations
2. Execute calculations on external resources and return results to SCEC
3. Modularize construction to enable evaluation of multiple alternative methods
4. Ensure repeatable and reviewable results

** We will use a workflow-based distributed computing framework developed on SCEC HPC Projects*

CISM Modular Processing Architecture



CISM Modular Processing Architecture



Focus CISM Software Development on Defining Workflows to Minimize Software Development

Workflow Configuration Environment

(CISM Software Development)

Workflow Execution Environment

(Existing Open-Source Software)

Workflow-Oriented System Implementation

1. CISM forecasts are implemented by running a series of scientific programs
2. CISM Workflows define the programs used, the input and output files, and order they must be run
3. Workflows are defined without machine, or computing environment, specific details (called abstract workflows)
4. After target run site is selected, abstract is “planned” and specific executables, and physical file names are inserted (called concrete workflows)
5. This technique is well suited for computing environments that move computing from one system to another

CISM Development Plan

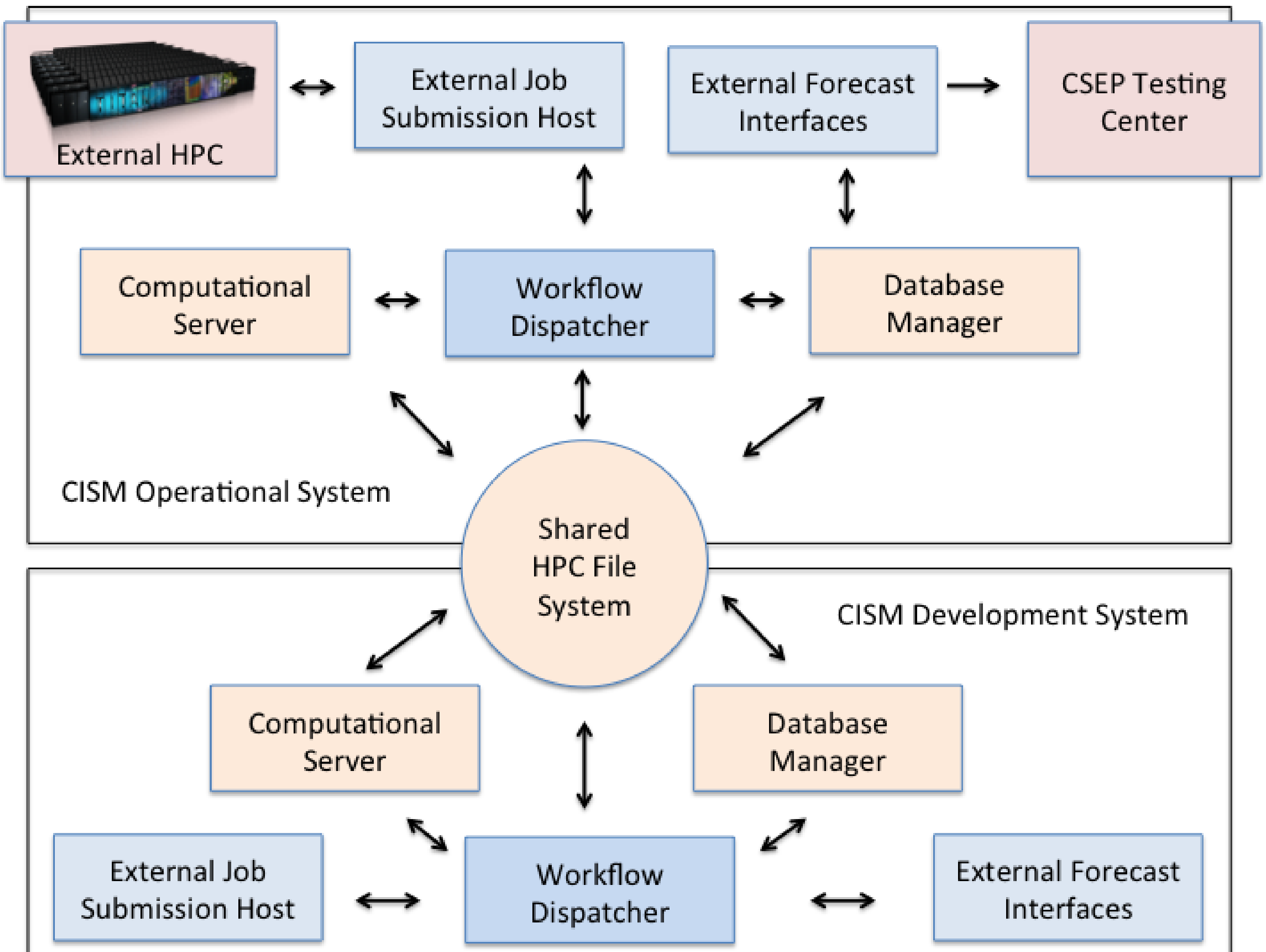
- System Requirements
- System Architecture
- **Computing Environment**
- Essential Software Components
- Computational and Data Estimates
- Software Development Process
- IP Considerations

Computing Environment

Develop a distributed computing environment, based at USC HPCC, that utilizes NSF and DOE HPC systems

- Establish both an operational and development computing environment
- Maintain cumulative data results locally
- Provide external interfaces to forecasts and forecast results

CISM Computational System



CISM Development Plan

- System Requirements
- System Architecture
- Computing Environment
- **Essential Software Components**
- Computational and Data Estimates
- Software Development Process
- IP Considerations

Essential CISM Scientific Codes

1. **OpenSHA.** Implement Uniform California Earthquake Rupture Forecast2 and 3, GMPEs, and probabilistic seismic hazard processing / Language: Java / Mult-threaded / Primary Developers: Ned Field, Kevin Milner
2. **RSQSim.** Large-scale simulations of earthquake occurrence to characterize system-level response of fault systems including processes that control time, place, and extent of earthquake slip/ Language: C / MPI-based / Primary Developers: James Dieterich, Keith Richards-Dinger
3. **CyberShake.** 3D wave propagation simulations for large set of ruptures, and seismogram processing resulting in peak ground motions and other parameters / Language: C / MPI-based / Primary Developers: Robert Graves, Scott Callaghan, Philip Maechling, Thomas Jordan
4. **CSEP.** Automated execution and evaluation of short term earthquake forecast models / Language: Python / Multi-threaded / Primary Developers: D. Schorlemmer, T. Jordan, M. Liukis

---

1. Field, E.H., T.H. Jordan, and C.A. Cornell (2003), OpenSHA: A Developing Community-Modeling Environment for Seismic Hazard Analysis, Seismological Research Letters, 74, no. 4, p. 406-419.
2. Richards-Dinger, K., and James H. Dieterich (2012) RSQSim Earthquake Simulator Seismological Research Letters, 2012 v. 83 no. 6 p. 983-990 doi: 10.1785/0220120105
3. Graves, R., T. Jordan; S. Callaghan; E. Deelman; E. Field; G. Juve; C. Kesselman; P. Maechling; G. Mehta; K. Milner; D. Okaya; P. Small; and K. Vahi (2010). CyberShake: A Physics-Based Seismic Hazard Model for Southern California, Pure Applied Geophys.,v.169,i.3-4 DOI: 10.1007/s00024-010-0161-6
4. Zechar,J. D., D. Schorlemmer, M. Liukis, J. Yu, F. Euchner, P. J. Maechling and T. H. Jordan (2010) The Collaboratory for the Study of Earthquake Predictability Perspective on Computational Earthquake Science Concurrency and Computation: Practice and Experience, Vol. 22, 1836-1847, 2010.

CISM Development Plan

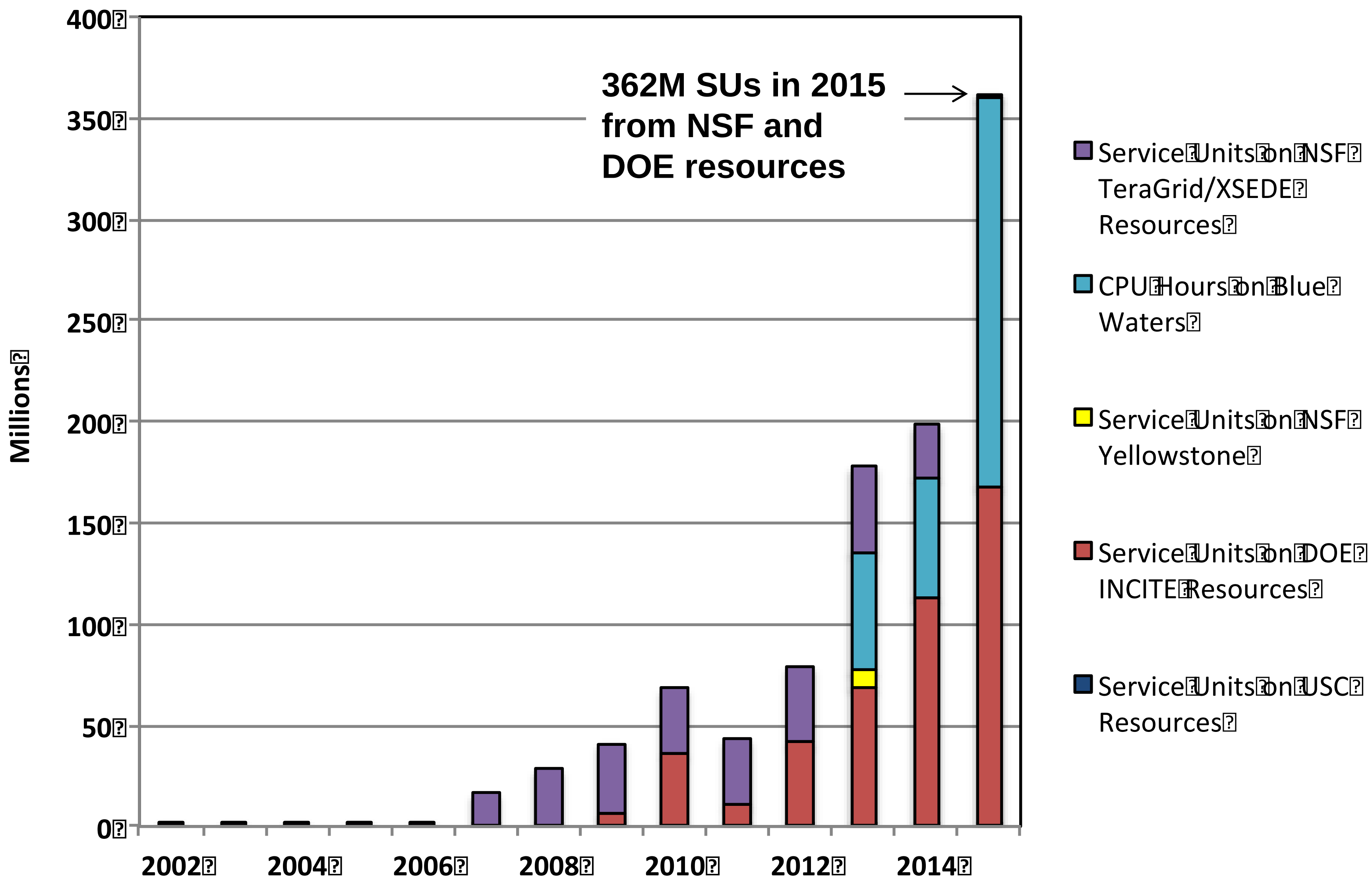
- System Requirements
- System Architecture
- Computing Environment
- Essential Software Components
- **Computational and Data Estimates**
- Software Development Process
- IP Considerations

Computing and Data Estimates

Estimated Annual Large-scale HPC Runs:

- **RSQSim: California fault system; simulated time: 100K yrs; number of ruptures: 100M; repetitions: 50**
  - **Core Hours Required: 40M**
  - **Local Results Data: 60TB**
- **CyberShake: Urban regions; spacing: 10 km (300 sites); max freq: 1 Hz; min Vs: 500 m/s**
  - **Core Hours Required: 70M**
  - **Local Results Data: 10TB**

HPC Resources for SCEC Research



CISM Development Plan

- System Requirements
- System Architecture
- Computing Environment
- Essential Software Components
- Computational and Data Estimates
- **Software Development Process**
- IP Considerations

SCEC Software Engineering Practices

Iterative Development Process (quarterly cycle)

- Develop end-to-end processing that provides scientific value
- Deploy operational system and operate during next iteration
- Extended system, preserving existing and add new capabilities

Software Engineering Practices

- Software version control
- Automated testing frameworks
- Standards based data formats and management
- Metadata collection
- Process logging
- Error detection and monitoring

CISM Development Plan

- System Requirements
- System Architecture
- Computing Environment
- Essential Software Components
- Computational and Data Estimates
- Software Development Process
- **IP Considerations**

CISM IP Principles

1. Integrate best-available academic codes developed and contributed by research community
2. Accept NSF-support and private company gifts to support software development
3. Release as free and open-source software to support scientific transparency and build confidence in results
4. License software in way it can be used by academic and US agencies including USGS.

Software Licensing

Apache License v2.0: Key Rights and Issues

1. Software distribution must include license
2. Software distribution must include source code
3. No warranty offered
4. User agrees to no liability
5. User are granted copyright to software and source code
6. Users granted patent license to use software
7. Users are not permitted to use any trademarks in distribution without permission
8. Private use is allowed
9. Commercial use is allowed
10. Redistribution is allowed with licenses intact
11. Users is allowed to make modifications
12. User must state what changes they made
13. User can distribute modifications under different, including proprietary, licenses
14. Users are permitted to link to any other software that uses different, including proprietary, licenses

Thank you!