How the CCA

Common Component Architecture

Advances Computational Science

Gary Kumfert

with David E Bernholdt, Thomas Epperly, James Kohl, Lois Curfman McInnes, Steven Parker, and Jaideep Ray

UCRL-PRES-222508

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

SciDAC
Scientific Discovery through Advanced Computing

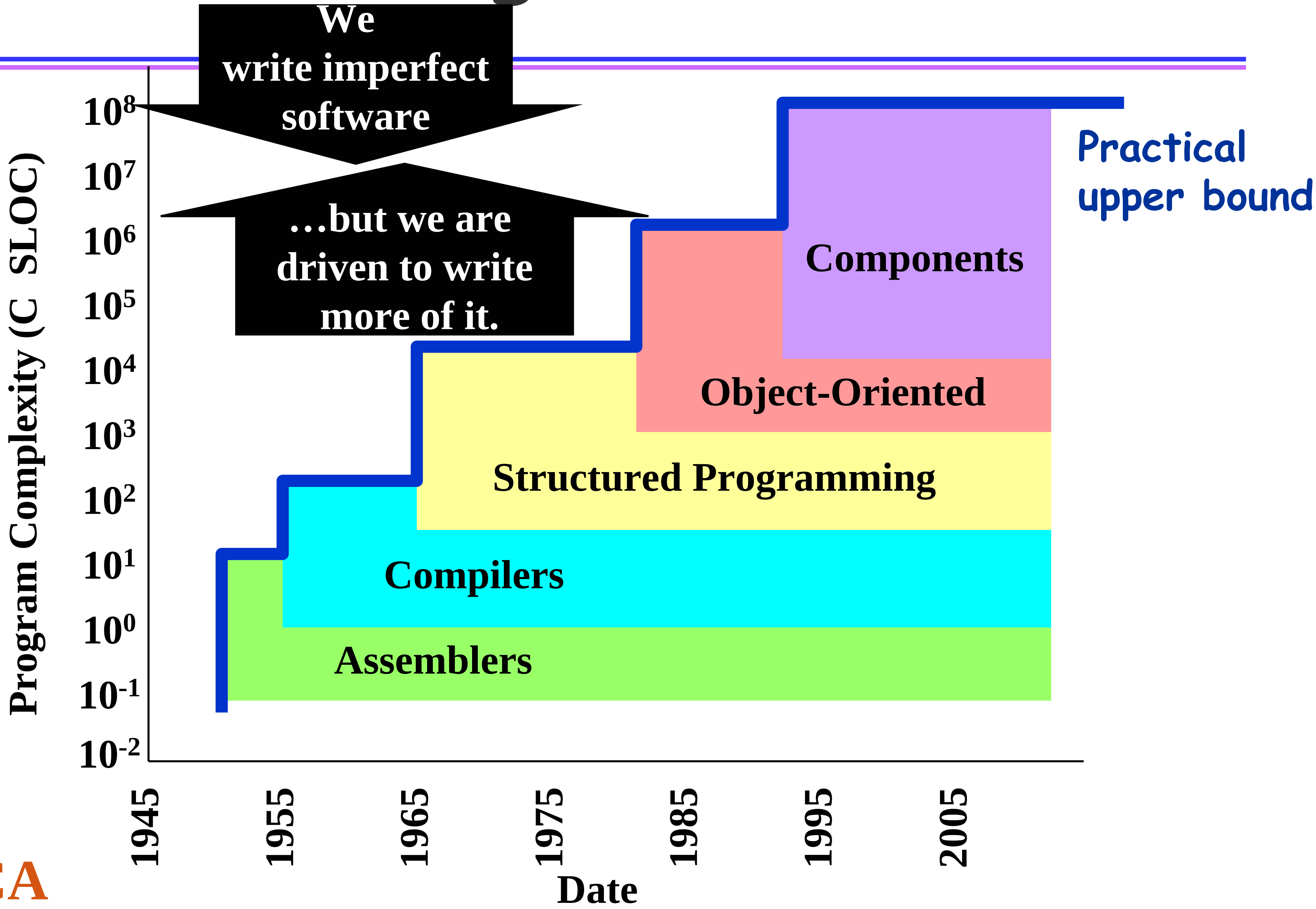
University of California
Lawrence Livermore National Laboratory

This talk is a survey of how CCA is used in science

Outline:

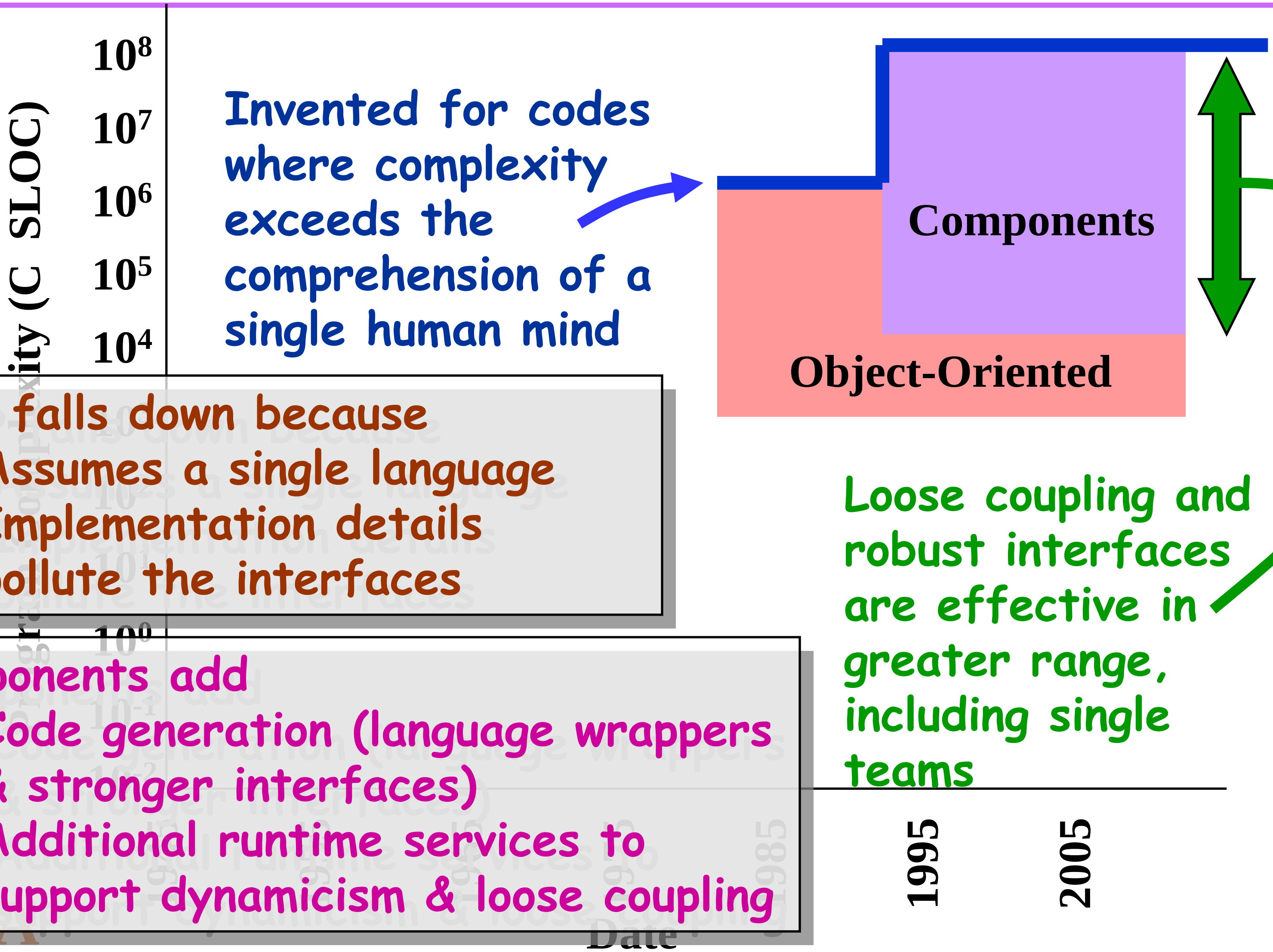
- **Components are important because…**
- **25 examples of CCA impact on science**
- **How CCA will lead future of scientific software technology (next 5 years)**

Human Beings Do Not Scale



CCA

In Industry, All Enterprise Software is Component Software



Invented for codes where complexity exceeds the comprehension of a single human mind

Loose coupling and robust interfaces are effective in greater range, including single teams

OOP falls down because

1. Assumes a single language
2. Implementation details pollute the interfaces

Components add

1. Code generation (language wrappers & stronger interfaces)
2. Additional runtime services to support dynamicism & loose coupling

Code Reuse is NOT the Reason for Components

Real reasons are robust interfaces & loose coupling, which can be used to great effect…

- **Corporate/For Profit**
  - ▶ **Time to Market**
- **Science/Research**
  - ▶ **Maintaining Correctness in the Face of Change**

CCA Delivers Component Technology to Scientific Computing

- **SIDL – Interface Language**
- **Babel – Reads SIDL, Generates wrapper code in C, C++, Fortran, Java, & Python**
- **CCA Specification – Defines a component, an interface (“port”) and how they interact with frameworks (written in SIDL)**
- **CCA Frameworks**
  - **Implement the CCA specification & provide services to components**
  - **Examples: CCaffiene, Uintah, XCAT**

CCA's Impact is as Diverse as the Applications in HPC

25 examples grouped roughly into six categories of impact/use

1. **CCA in single codes for extra flexibility**
2. **CCA to combine incompatible codes**
3. **CCA to develop community standards (& deliver interchangeable codes)**
4. **CCA a la carte: Using parts of CCA tech.**
5. **CCA to bridge frameworks**
6. **CCA's impact on competing technologies**

1. CCA in single codes for increased flexibility

Application	Project	POC
Combustion	CFRFS	Jaideep Ray, Sandia
Chemistry	NWChem & Global Arrays	Theresa Windus, PNNL
Subsurface Transport	PSE Compiler	Jans Prins, UNC Chapel Hill
Geomagnetics	–	Shujia Zhou, NASA Goddard
Performance Monitoring	TAU	Sameer Shende, U Oregon
Sparse Linear Algebra	Sparsekit-CCA	Masha Sosonkina, Ames Lab

Example: CCA in Combustion

- Novel high order (4th & 6th) discretization for SAMR
- Developed an extended stability R-K-C integrator for ADR on SAMR
- 5 refereed science papers
- 8 refereed software papers
- Quantitative study on how components affected their code

OH concentration in advective-diffusive-reactive simulation using 4th order Runge-Kutta-Chebyshev integrator on 4 levels of AMR

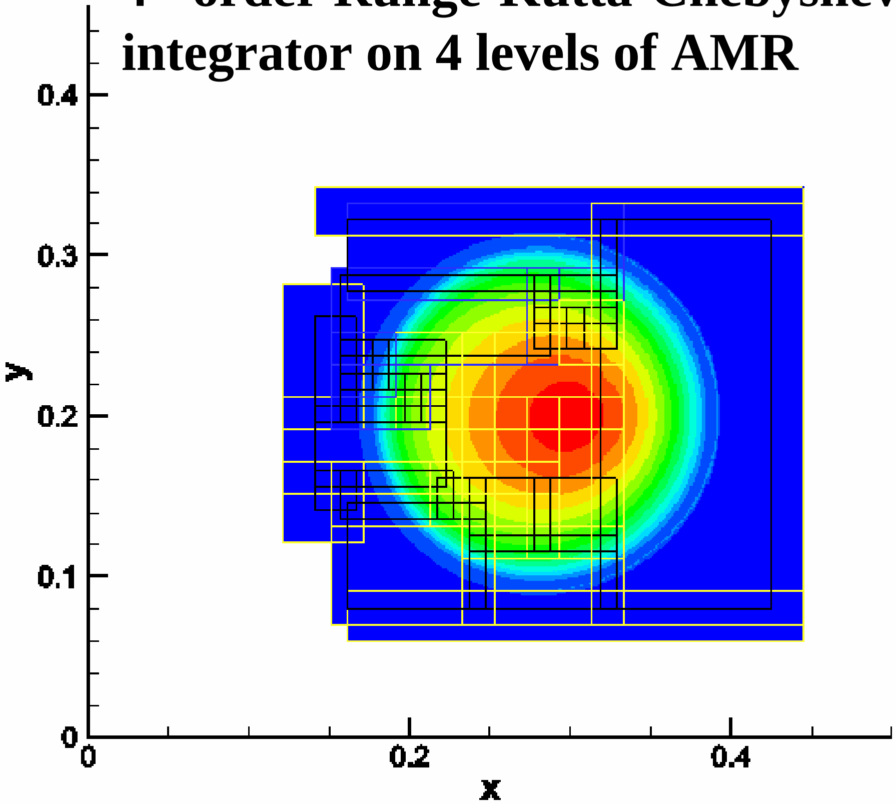


Figure courtesy of Jaideep Ray, SNL

CCA

2. CCA to combine previously incompatible codes

Application	Project	POC
Quantum Chemistry	MPQC & NWChem	Curtis Janssen, Sandia Theresa Windus, PNNL
Nuclear Power Plant Training Sim		M. Diaz, U. Malaga, Spain
Fusion	DFC	Nanbor Wang, Tech-X Corp.
Radio Astronomy	eMiriad	Athol Kemball, UIUC

Example: Quantum Chemistry

Better instruments for scientific inquiry by integrating best-in-class software packages

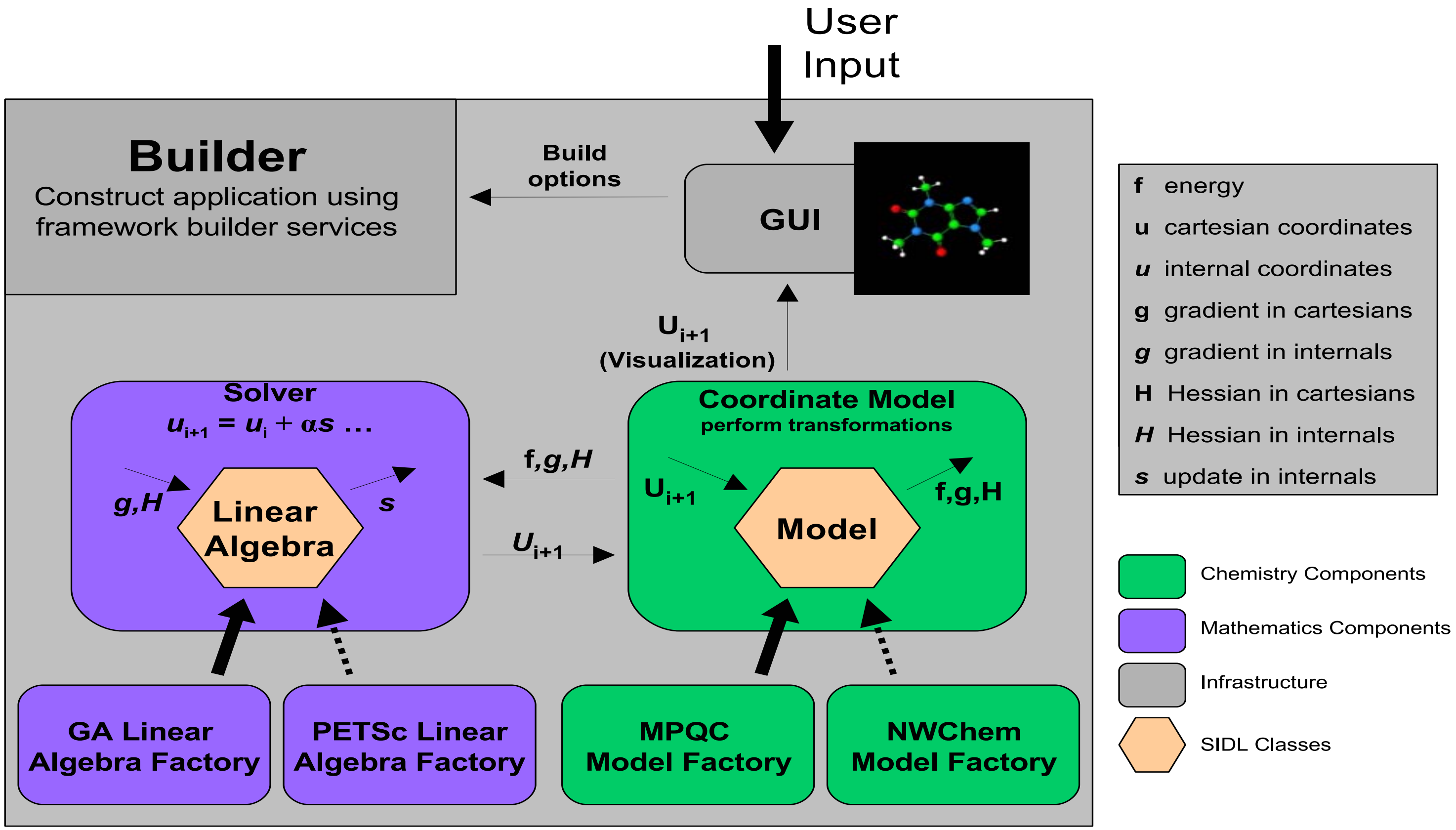


Figure courtesy of Curtis Janssen and Joe Kenny, SNL

CCA

3. CCA to Develop Community Standards

Application	Project	POC
Meshing	TSTT	Lori Diachin, LLNL
Solvers	TOPS	Barry Smith, Argonne
… and Applications using these interfaces		
Cell Biology	VMCS (using TSTT)	Harold Trease, PNNL
Accelerator Beam Dynamics	Beam-SBIR (will use TOPS)	Douglas Dechow, Tech-X Corp.
Chemistry	GAMESS-CCA (NWChem&MPQC)	Masha Sosonkina, Ames Lab

4. CCA a la carte: using parts of CCA technology

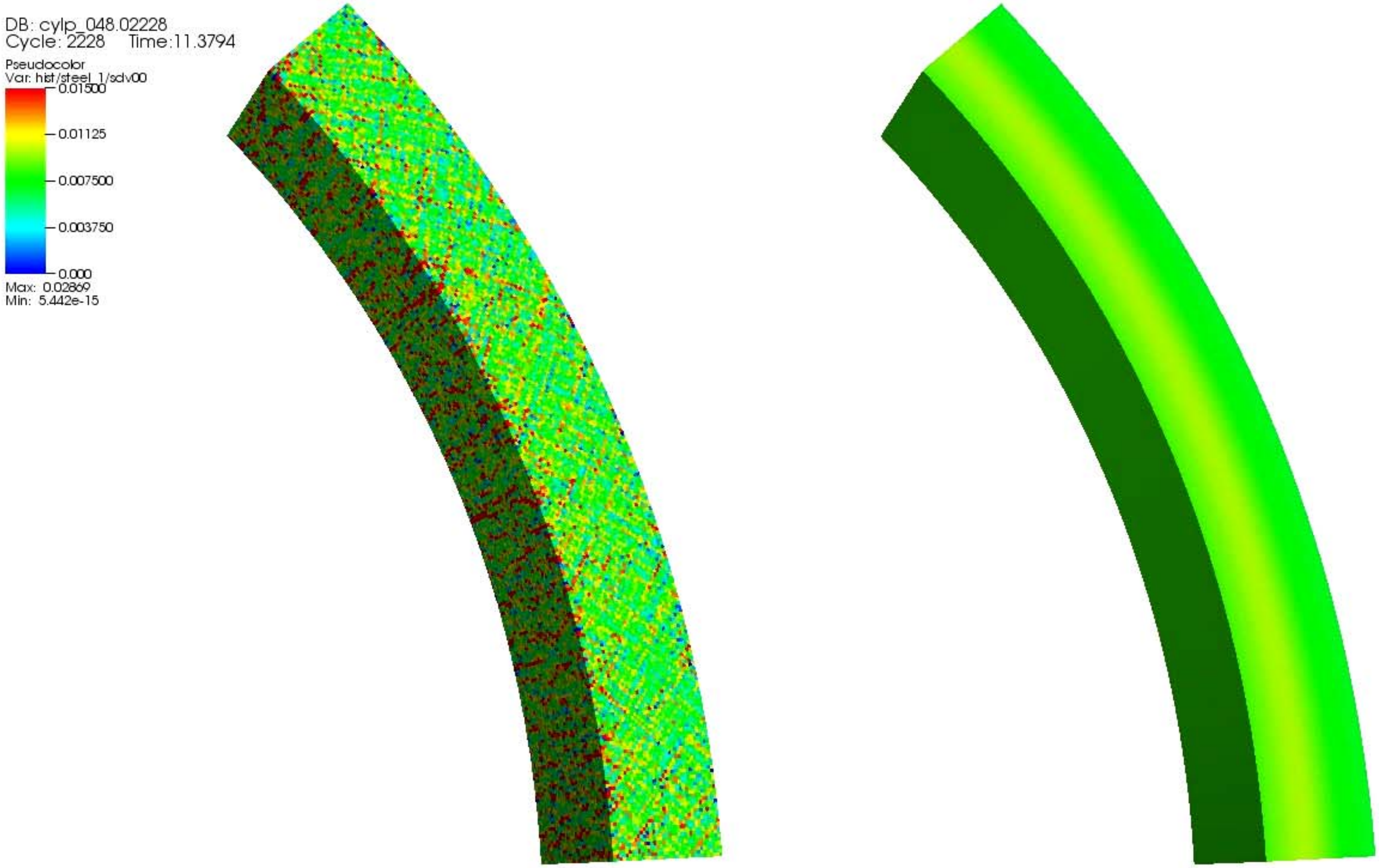
Application	Project	POC
Combustion	CFRFS	Jaideep Ray, Sandia
Electron Effects	CMEE	Peter Stoltz, Tech-X Corp.
Material Science	PSI	David Jefferson, LLNL
Computer-Assisted Source Refactoring	CASC	Dan Quinlan, LLNL
Fusion	FMCFM	Johann Carlsson, Tech-X Corp.
Solvers	Hypre	Jeff Painter, LLNL

Ccaffeine Classic
(C++ only)

BABEL

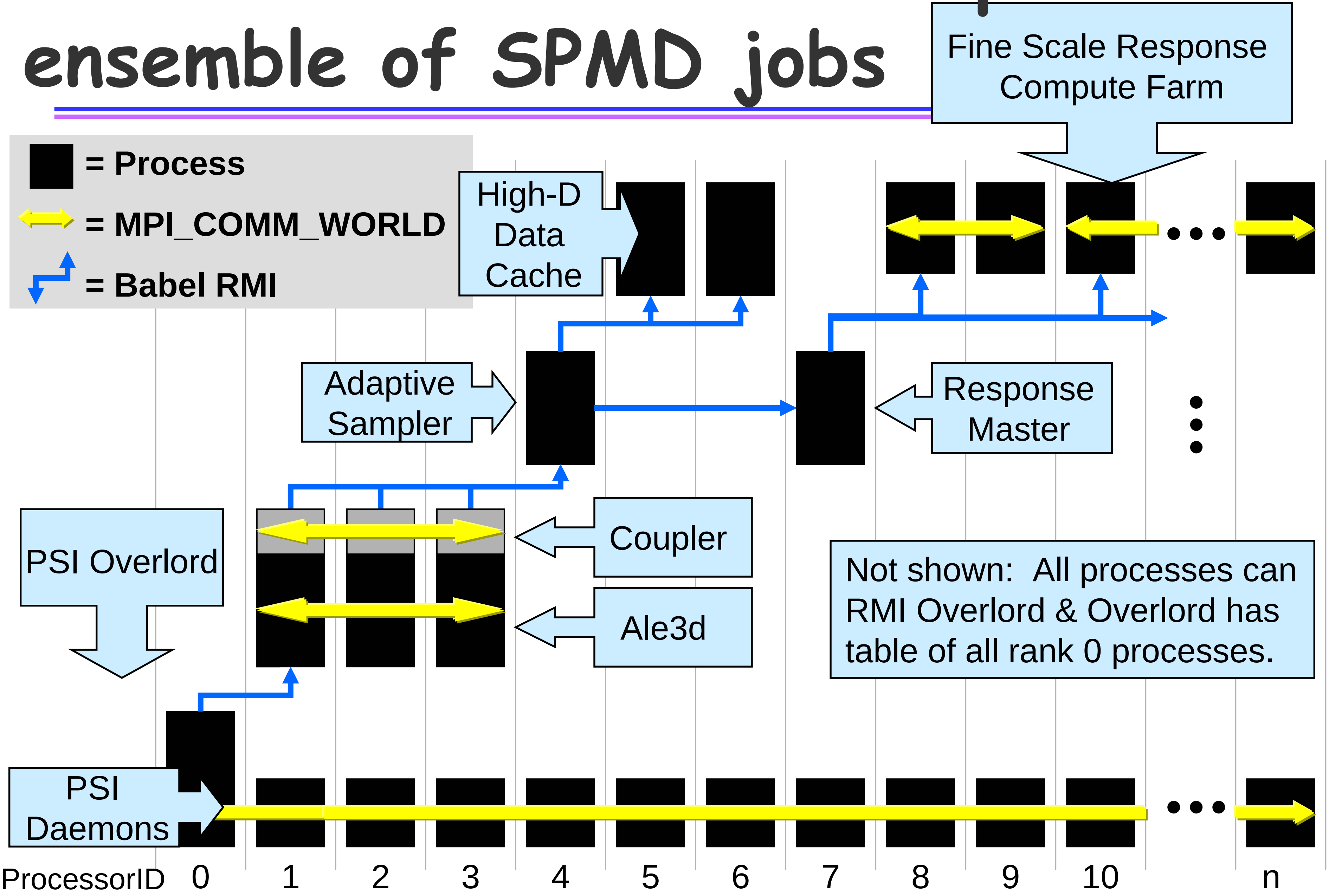
Strain on Shock-Driven Metal Cylinder

Courtesy, Nathan Barton, LLNL.



Multiscale **Continuum Only**

Vision: The Petascale computer as ensemble of SPMD jobs



Not shown: All processes can RMI Overlord & Overlord has table of all rank 0 processes.

5. CCA to connect frameworks

Framework	Comment	POC
SCIRun2	Meta-Component Bridging	Steve Parker, Utah
Legion-CCA	Extended Babel to Generate Legion	Michael J. Lewis, Binghamton University
MOCCA	Personal Grid Environments (Part of Harness)	Vaiday Sunderam, Georgia Tech

6. CCA's impact on competing technologies

Application	Project	POC
Climate	ESMF	Nancy Collins, NCAR
Astrophysics	TSI	Doug Swesty, SUNY Stony Brook

"I have become a complete convert to the idea of component-oriented design and it is now foremost in my mind when it comes to software architecture planning."

-- Doug Swesty, SUNY Stony Brook

" Gary, there are a b'jillion references to CCA at this HPDC/Compframe workshop… These are all Europeans we haven't met before. "

-- Rob Armstrong, Paris, last week

CCA

Future Directions of CCA: “Adaptivity”

- Computational Quality of Service (CQoS)
  - Tradeoffs: performance, accuracy, robustness
  - Motivated by: Accelerators, Combustion, Quantum Chemistry, Fusion,…
  - In collaboration with: PERC, TSTT, TOPS
- Hybrid Computing
  - Driven by: multi-core/hybrid-core arch.
- Interface Semantics
  - Dynamic enforcement of semantic errors

Conclusion

- Components are serious technology for building large scale codes
- CCA accomplishments include:
  - delivered technology uniquely applicable for HPC
  - Demonstrated broad impact across multiple application domains
  - Demonstrated technical leadership within our own CS discipline
- Vision: build a component ecosystem DoE
  - Researchers spend more time in the 10% of their code that is of scientific interest
  - Share the other 90% necessary for completeness

Thank You

CCA

Common Component Architecture

www.cca-forum.org