Library of Mini-Applications for Exascale Component-Based Performance Modelling

Bof124s1

SC13, Denver

Our Speakers

- Dr. Naoya Maruyama for Riken
- Dr. Jackie Chen for Coe Exact
- Dr. Richard Barrett for Mantevo
- Dr. Marie-Christine Sawley for the EU project Exa2CT and EESI2

Overview: MiniApps BoF

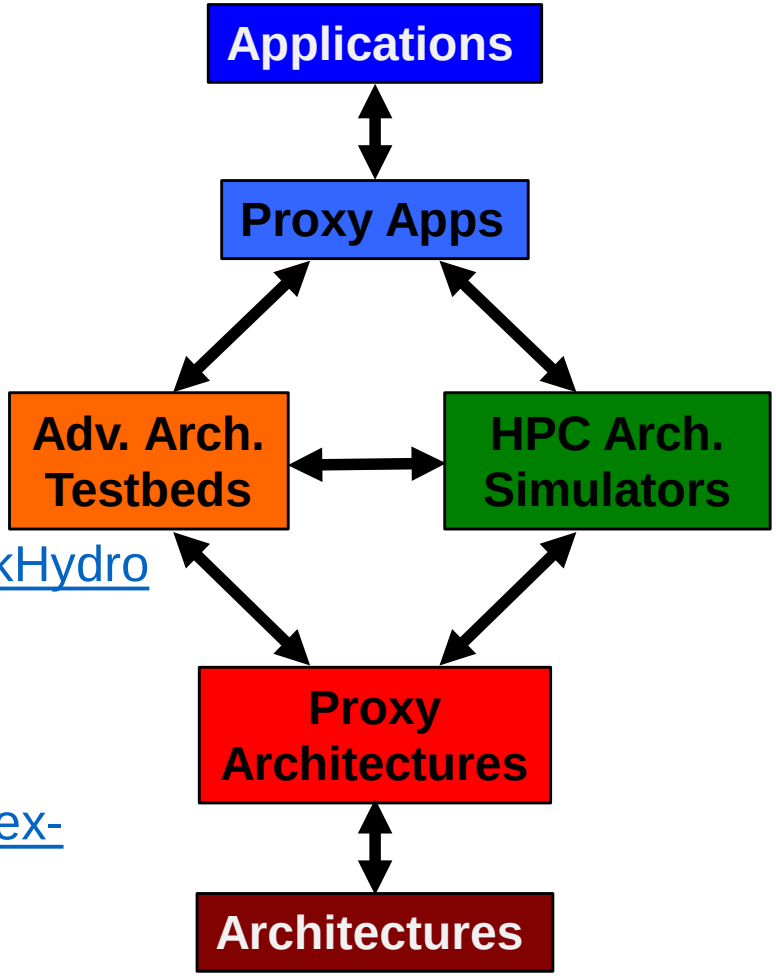
- Mini-Applications and Proxy Apps
  - Express **essential** parts of application that are essential to science
  - Express **pain points** for performance
  - Hide complexities of **REAL** apps that are **inconsequential** to science

- Proxy Hardware Models are a Reflection of the Underlying Machine Architecture
  - *Express what is **important** for performance*
  - *Hide complexity that is **not** consequential to performance*

- Use Proxy Applications on Proxy Hardware Architectures to understand what programming model choices offer the highest value *(using modeling and simulation)*
  - Not absolute performance predictions
  - Scientific experiments that gauge relative improvements over ***control***

Proxy Architectures are a a key Capability for HPC Co-design

- Key Co-design Elements
  - HPC Architectural Simulators
  - Proxy Applications
  - Advanced Architecture Testbeds
  - Proxy Architectures
- Proxy Applications
  - http://www.mantevo.org
  - http://www.github.com/losalamos
  - https://computation.llnl.gov/casc/ShockHydro
- HPC Simulation
  - http://code.google.com/p/sst-simulator
  - http://www.nersc.gov/research-and-development/exascale-computing/codex-project/



Discussions

- Characteristics you expect from a mini application
  - Reflects full-scale application performance characteristics
  - Captures end-to-end application behavior with manageable codebase
    - Unlikely to be a single-loop kernel
  - Downloadable source code
  - Performance models for explorative performance studies
- Features to have the impact in the community
  - Allows for application-level performance studies
    - Used to explore architecture parameters in the Exascale Feasibility Study projects
- Gaps that you can identify
  - Mechanism to check whether mini-apps accurately reflect original full-scale apps
  - No comprehensive, permanent performance database

Performance Data Repository

- Database to store:
  - Performance results
  - Performance models
  - Application and system information (meta data)
- Allows for comparisons across machines and applications
- Allows for “what-if” studies with performance models
- Ongoing collaborative projects between Tokyo Tech and ORNL

Miniapps :
Tools enabling exploration

Focus	**Proxy for a key app performance issue**
Developer/owner	Application team
Intent	Output is information
Scope of change	Any and all
Size	A few thousand lines of code
Availability	Open source (LGPL)
Life span	Until its no longer useful

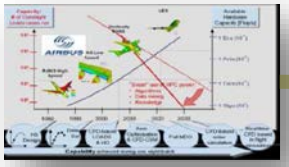
Related:

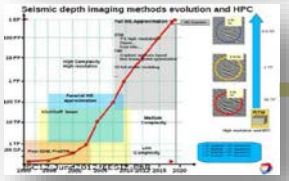
Benchmark	**Output: metric to be ranked.**
Compact app	Application relevant answer.
Skeleton app	Inter-process comm, application “fake”
Proxy app	Über notion

Gaps…coming soon:

- Adaptive mesh refinement
  - miniGhost, Cloverleaf; miniAMR, CleverLeaf
- miniContact : solid mechanics
- Data analytics:
  - PathFinder
  - GraphApp
  - SAT(isfiability)
- Sorting algorithm targeting multi-level memory
  - Sparse matrix-matrix multiplication
    - K means clustering
    - Betweenness Centrality
    - Algebraic multi-grid (AMG)
- Your problem?

Proto-Apps

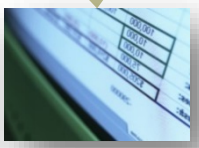




Sometimes considered identical to
Mini-apps, *Proxy-Apps*, *Skeleton Apps*

However proto-apps are not reduced version of real-world applications, they are specifically designed to be :

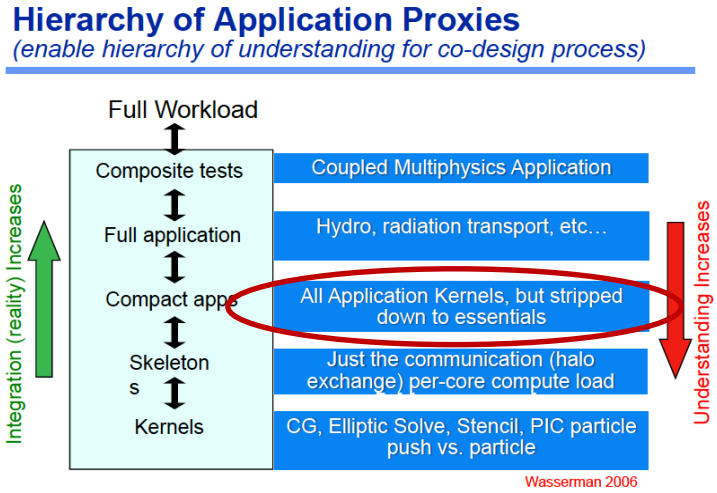
- demonstrating high scalability for re-factored applications
- encapsulating essential characteristics:
  - New exascale algorithms
  - Advanced parallel implementations



R6. Mini apps: an holistic approach

Motivations: contribute to the design of the Exascale hardware, software and underlying

- project how future applications will interact with future computing hardware
- represent the behaviour of the complete workload with sufficient fidelity



Proposal :

- flexible test bed: architectural simulation framework, pre-production HW prototype, scalable Exascale software, component base systems, HPC providers
- capture a code's essential features - core data structures, algorithms, parallel constructs- while demonstrating state of-the-art implementations relying on modern programming paradigms
- an holistic approach : applications, sofware&hardware
- An international effort: US/DOE co-design initiatives, Japanese HPCI Exascale Feasibility Study-Mini Applications project, FP7 projects (CRESTA, Mont Blanc, DEEP), Intel European Exascale Labs

Questions

- Do we want to network around a common platform of mini apps or vision of what they bring?
- How to bridge between the different relevant efforts?
  - Building a repository, library, join efforts
- How do we push our findings into the community?
  - Webinars
  - Conferences: EESI2, BDEC February 2014, etc…