H-New Approaches for Faster, More Affordable CSE Software

Lead: Sandy Landsberg
Presenter: Ira Baxter, Jim Kirby
Writer: Sandy Landsberg

H - Members

- Sandy Landsberg, sandy.landsberg@hpc.mil
- Devin Matthews (dmatthews@utexas.edu)
- Tom Clune Thomas.L.Clune@nasa.gov
- Walid Keyrouz (walid.keyrouz@nist.gov)
- Tien N. Nguyen (tien@iastate.edu)
- Mike Glass (mwglass@sandia.gov)
- Christos Kartsaklis (kartsaklisc@ornl.gov)
- Douglass Post (post@ieee.org)
- Elaine Kant (kant@scicomp.com)
- David Tarboton (dtarb@usu.edu)
- James Kirby (James.kirby@nrl.navy.mil)
- Ira Baxter (idbaxter@semanticdesigns.com)
- Hridesh Rajan (hridesh@iastate.edu)
- Albert Reuther (reuther@LL.mit.edu)

H-Charge

Explore flexible software engineering techniques (product line engineering, model driven architecture etc.); software architectures (cloud platforms, learning systems); system architectures (software defined networking); business/sustainability models.

A. What are some of the “next best thing” approaches that could revolutionize CSE Software?
B. What are the potential paths and payoffs of these approaches?
C. What role can funding agencies play in fostering efforts?

Long term vision: **Abstractions and Automation**

A. Developing shared, extensible abstractions is the future of CSE
   a. Community development facilitates expert contributions and increases buy-in
   b. Shared abstractions at multiple levels support mixing packages and automated tools
      - Science, math, algorithms, data, computer science, hardware
   c. Augment abstractions with flexible Interfaces

B. Capture design decisions behind models, software, hardware
   a. Useful for human communication
   b. Enable automated tools to offer choices and make default decisions

C. Understand and modernize legacy code
   a. Reverse engineer legacy code using abstraction and design choices
   b. Automate refinement back to efficient code

D. Software Synthesis / Correctness by construction
   a. Correct abstractions
   b. Validation and verification - at and between levels
   c. Automated generation between levels of abstraction
   d. Benefits: fast science experimentation *and* efficient implementations, portable, shorter turnaround time

Near term approaches

A. Start on long term goals

B. Software product line engineering
   a. Adaptable components
   b. Flexible interfaces

C. New techniques
   a. Exploiting asynchronous, heterogeneous execution and communication
   b. Hierarchical parallelism
   c. Visual modeling tools

D. Build a community to build, consume, contribute, and sustain software
   a. Consistency of interface, abstractions, community buy-in
   b. Incentivize conforming to interfaces and abstractions
   c. Multi-disciplinary teams that trust each other and have incentives to work together

Potential paths and payoffs

A. **End-to-end productivity improvement, faster time to solution**
B. **Critical part of our industrial and national infrastructure**
C. **Relieve burden on experts and improve productivity**
D. **Reduce coupling between experts in different disciplines**
E. Collect knowledge in re-usable forms
F. Pathway to rapid use of changing hardware
G. Integrity/Trustworthy computing - correctness by construction
H. Adaptability to inevitable change
I. Accessibility
J. Power efficiencies and environment

What role can funding agencies play in fostering efforts?

A. Fill gap between research and production usable software
B. Coordinate research programs in productivity
C. Incentivize appropriate reuse of knowledge as well as code
D. Software management plan / Software use plan
E. Enhance buy-in by funding demonstrations of application of automated tools
F. Foster communities
G. Business model for sustainability
H. Providing funding to develop and disseminate knowledge to scientists and engineers about abstractions, scientific programming, and software
    a. Current workforce
    b. Students (next-generation workforce)
I. Long-term funding of repositories & shared knowledge & abstractions
    a. NSF SSE repository

Questions?

What are some of the “next best things” approaches that could revolutionize CSE software?

A. Asynchronous communications
B. Visual modeling tools
C. Software product line engineering (near-term)
D. Flexible Interfaces
E. Multi-disciplinary teams that trust each other and have incentives to work together
F. Build a community to build, consume, contribute, and sustain software
    a. Consistency of interface, abstractions, community buy-in
    b. Incentivize

What are some of the “next best things” approaches that could revolutionize CSE software?

A. **Abstractions and Automation**
B. Finding the abstractions, that are extensible, is the future of CSE
C. Capture design behind the software to help the user use and sustain software with automated support that can interpret the design
    a. Capture design decision
D. Embedded DSLs - allow you to use abstraction where it best serves
E. Automated Code Generation
F. Correct abstractions: Hierarchical parallelism
G.

What are some of the “next best things” approaches that could revolutionize CSE software?

A. Asynchronous communications
B. Visual modeling tools
C. Software product line engineering (near-term)
D. Flexible Interfaces
E. Multi-disciplinary teams that trust each other and have incentives to work together
F. Build a community to build, consume, contribute, and sustain software
    a. Consistency of interface, abstractions, community buy-in
    b. Incentivize

What are the potential paths and payoff of these approaches?

A. **End-to-end productivity improvement. Faster time to solution**
B. **Critical part of our industrial and national infrastructure**
C. We have to do this to be able to use future machines
D. Being able to keep up with rapidly changing hardware
E. Trustworthy computing...correctness by construction
F. Integrity
G. Adaptability to inevitable change
H. Accessibility.
I. Power efficiencies and environment
J. ...

What role can funding agencies play in fostering efforts?

A. Fill gap between research and production usable software
B. Coordinate research programs in productivity..
C. Mandate reuse (needs further definition of what)
D. Software management plan / Software use plan
E. Foster communities
F. Business model for sustainability
G. Providing funding to develop and disseminate knowledge to scientists and engineers about abstractions, scientific programming, and software programming
    a. Current workforce
    b. Students (next-generation workforce)
H. Long-term funding of Repositories & Shared abstractions
    a. NSF SSE repository