

Runtime Systems Panel

- Panelists: David Grove, Sanjay Kale, Kevin Pedretti, Wilf Pinfold, Kathy Yelick
- Moderator: Vivek Sarkar
- Format:
 - Short position statements by panelists (at most 3 minutes, 1 slide)
 - Questions from audience
 - See https://xstackwiki.modelado.org/Runtime_Research_Questions for possible topics



Summary of Panelist Statements

- Grove: X10/APGAS Runtime includes X10RT, Native Runtime, XRX --- supports Resilient X10, Elastic X10
- Kale: need communication between whole machine runtime, single-job runtime, language runtime; inter-node runtime is more important than intra-node runtime!
- Pedretti: dynamic adaptation will add complexity to OS-RT interfaces, resource management and sharing
- Pinfold: need hints to balance scheduling responsibility between app developer and adaptive runtime
- Yelick: two possible approaches with same end goal --- static mapping as default with dynamic extensions, dynamic mapping as default with “localization” extensions



Key points in discussion (1/2)

- Debugging bulk-synchronous programs is easier than debugging dynamic parallelism
 - Counterpoint 1: we have no choice --- debugging vector programs was easier than debugging bulk-synchronous programs!
 - Counterpoint 2: event-driven runtime model are already known to be better suited than bulk-synchronous model for some applications
- Runtime viewpoints appear to be computation-centric with locality goals; can we instead consider a data-centric view with computation goals?
 - Focus on memory/network bandwidth, rather than computation, as main optimization goal
 - How should we measure impact of runtime on power and data movement?



Key points in discussion (2/2)

- If successful, dynamic runtimes can potentially make future hardware more affordable due to reduced guard bands
- How to create testbeds with increased performance variability to better represent future hardware?
 - Turn on Turbo boost
 - Run synthetic workloads (e.g., STREAM) on selected cores to increase perturbation
- How can a runtime support both user directives and automated adaptations in an integrated manner?
 - Analogous to integrating user optimizations and auto-tuning
- What runtime approaches should be pursued to support heterogeneous memories?

