Panel Discussion

Performance Tools and Their Interfaces

Moderator: Martin Schulz, LLNL

Panelists: Ron Brightwell (Sandia NLs), Romain Cledat (Intel),
Jeff Hollingsworth (UMD), John Mellor-Crummey (Rice),
Brian Van Straalen (LBL)

Performance Tools and Their Interfaces

- Role of performance tools in the exascale software stacks?
  - What do users expect from tools?
  - Which questions should they address?
  - What abstractions should they map to?
- What interfaces will be / can be available?
  - What can runtimes and low-level software layers expose?
  - How can we capture and map to high-level semantics?
- What is currently missing?

Panelists

- Ron Brightwell (Sandia NLs)
- Romain Cledat (Intel)
- Jeff Hollingsworth (UMD)
- John Mellor-Crummey (Rice)
- Brian Van Straalen (LBL)

Q1: What types of performance problems do you want tools to diagnose?

- Short answer: Any and All
- High-level views down to instruction scheduling
- Important target is data motion
  - Cache behavior, data transferred, prefetch efficiency, …
- Limited concurrency (e.g., in DAGs / critical paths)
- Understand impact of autotuners and policy decisions

Q2: What abstractions should a tool use to display its information in?

- Abstraction needs to match the programming model
- Need support for mapping abstractions
  - Who implements this, the developer of the code, tool or DSL?
  - Should be "standardized" – DWARF3?
- Needs static and dynamic support - No longer a question of "or"!
- Tension between abstractions and seeing raw measurements
  - Everything has its target audience and time
  - Must be configurable

Q3: What interfaces should we and can we expose in the software stack?

- Interfaces to all layers of the stack
  - Break open the black box
  - Observe any changes in underlying resources
- Interfaces for sensors and actors
  - Ability to measure, decide, and steer
  - Influence policy where necessary
  - Feedback from runtime on decision impact
- Feedback ability for the user (flag wrong behavior)

Q4: What one thing do you fear that we will not be able to know/measure that will limit applications at exascale?

- Interference and contention measurements
  - OS containers may help
- Hardware features for performance isolation
- Artificial slowing of machine to improve observations
- System will turn into a stochastic system
  - Hard to attribute performance data
  - Need new ways to distinguish sources
- Hard to get data out of lightweight cores/accelerators