ASCR Programming Challenges Workshop

William Harrod
Sonia R. Sachs
July 27-29, 2011
Welcome and Goals

- Welcome
- Workshop Goals:
  - *Define objective criteria* for assessing programming models and language features that enable effective use of diverse Exascale architectures for important science applications.
  - *Prioritize challenges for programming models, languages, compilers and runtime systems for Exascale*
  - *Prioritize options* for
    - evolutionary path,
    - revolutionary path and
    - bridging the gap between evolutionary and revolutionary paths
  - *Create a roadmap*, with options, timeline, and rough cost estimates for programming Exascale systems that are responsive to the needs of applications and to future architectural constraints
• **Presentations** on compiler and runtime engines that support advanced programming models and languages.

• **Focused Parallel Panel Discussions**
  – **Language Constructs Panel:** Develop objective criteria to assess language constructs in the context of co-designed applications and hardware to execute them.
  – **Compiler and Runtime Engines Panel:** Develop objective criteria to assess compiler and runtime engines to support advanced programming models and language constructs.
  – **Migration Panel:** Develop options to automatically connect/transform/migrate to advanced programming environments.

• **General Panel Discussions**
Challenges to the HPC Community

• Do Embedded Systems Concurrent Languages have interesting/appealing features for HPC?
  – Data Flow languages:
    • Synchronous Data Flow
    • Dynamic Data Flow
    • LUSTRE\(^1\) (a synchronous data flow language in which equations uniquely define variables),
    • SIGNAL\(^2\) (another synchronous data flow language in which programs are equations and block diagrams),
    • Esterel\(^3\),
    • Actors\(^4\).

Challenges to the HPC Community

• Do Embedded Systems Concurrent Languages have interesting/appealing features for HPC?
  – Control-oriented languages:
    • Event-driven state machines,
    • Statecharts (event-driven model, equivalent to FSMs in expressiveness, but exponentially smaller).
  – Heterogeneous models/languages frameworks
    • Ptolemy,
    • Metropolis,
    • Simulink.
General Discussion Questions

• How can we enable scientists to focus on their science rather than the fine details of a complex Exascale system?
• Will Exascale applications become a collaborative effort, requiring a hierarchy of developers for the different levels of programming focus?
• Will programs at a given level of abstraction be optimized without knowledge of how lower layers are implemented?
General Discussion Questions

• Which novel work on programming languages and compilers is needed in order to enable the mapping of the high level constructs, devised by domain scientists, to the machine optimized constructs?

• What new programming models and language constructs can express fine-grained asynchronous parallelism to achieve performance, programmability, and efficiency in the face of disruptive technology changes while simultaneously meeting the needs of an evolving and increasingly unstructured application base?
• Programs for Exascale computing should be portable by design (“write-once-run-everywhere”) and written to scale across multiple hardware generations. What can be done to ensure that this is the case?

• Programs must be able to exploit massive parallelism and must do so in a way that avoids indeterminacy. Today, only modest parallelism can currently be extracted from programs. What can be done to make programs much more concurrent?