Report on Exascale Software Roadmap

Naoya Maruyama (RIKEN AICS)
(presented on behalf of the SDHPC software groups)

Working Groups

- Systems Software
  - Akihiro Nomura, Masaaki Shimizu, Ryosei Takano, Shunji Uno, Hiroya Matsuba, Toshihiro Konda, Takeshi Nanri, Osamu Tatebe, Hitoshi Sato, Takashi Yasui, Yoshiyuki Ohno, Hidemoto Nakada, Atsuko Takefusa, Toshio Endo, Yoshikazu Kamoshida, Shinichiro Takizawa, Hideyuki Jitsumoto
  - Advisors: Taisuke Boku, Yutaka Ishikawa, Atsushi Hori, Mitaro Namiki, Shinji Sumimoto
- Programming
  - Naoya Maruyama, Hiroyuki Takizawa, Kenjiro Taura, Tasuku Hiraishi, Atsushi Kubota, Masahiro Yasugi, Masahiro Nakao
  - Advisors: Hiroshi Nakashima, Mitsuhisa Sato, Akinori Yonezawa
- Numerical Libraries
  - Takahiro Katagiri, Reiji Suda, Daisuke Takahashi, Takeshi Iwashita, Kenji Ono, Satoshi Ito
  - Advisors: Kengo Nakajima, Ryutaro Himeno, Satoshi Sekiguchi, Kimihiko Hirao, Akira Ukawa

Software R&D Roadmap

- Objective: Supporting scalable, low-power, and fault-tolerant application executions

- Focused research themes
  - *Heterogeneity*
  - *Scalability*
  - *Memory wall*
  - *Power*
  - *Fault tolerance*
  - *Productivity*

Systems Software (1)

- OS and runtime re-designed for heterogeneous architecture
  - Light-weight OS for throughput-oriented cores
  - Support of the POSIX interface?
- Scalability for O(100K)--O(1M)-node systems
  - Removing OS noise
  - Scalable thread management
  - Adaptive communication algorithms
- Fine-grained power control for power-provisioned systems
  - Needs standardized APIs for accessing critical data for controlling system power consumption
  - Provides interface for higher-level software layers for controlling power allocations

Systems Software (2)

- Communications
  - Low-latency, asynchronous communication libraries
  - Dedicates a few cores for communications?
  - Direct communications between accelerators
- Scalable I/O
  - High-level I/O libraries (e.g., PnetCDF, netCDF4, HDF5)
  - Scalable fault-tolerant parallel file systems
  - Node-local NVRAM
- Fault resilience framework
  - API for fault detection and notification
  - Scalable checkpointing using local NVRAM

Needs to work with international communities for standardized common APIs

Programming

Future-proof programming models supporting:

- Heterogeneity
  - Portable, productive programming models
- Scalability
  - Programming models and interfaces for hierarchical algorithms
- Memory wall
  - PGAS-like programming models for deep memory hierarchy
- FT
  - Allows programmers to express critical and non-critical data → Reduced checkpoint image size by selectively checkpointing critical data
- Power
  - Programming interface/model for fine-grained power control

Programming R&D Approaches

- Directive-based programming
  - Directives for PGAS (e.g., XMP)
  - Directives for accelerators (e.g., OpenACC)
  - Allows for incremental development
- Domain-specific frameworks
  - Programming interface with domain vocabularies
    - E.g., MapReduce, PDF frameworks, etc.
  - Transparent domain-specific optimizations
    - E.g., Computation and communication overlapping in stencils
  - Allows for highly productive programming

Needs to establish long-term development and support community

Numerical Libraries

- Heterogeneity
  - High performance libraries for heterogeneous machines
- Scalability
  - Communication-avoiding algorithms
- Memory wall
  - Locality optimizations
- Fault tolerance
  - Algorithm-based fault tolerance
  - Fault-aware high-performance libraries
- Power
  - Power-efficient algorithms

Needs to extract common computation patterns from the target applications

Overall Roadmap