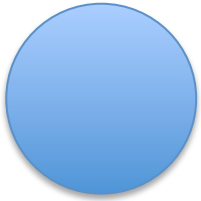

Blue Dot Discussion

Computational Challenges and Needs for Academic and Industrial Application Communities

Jean-Claude Andre, Jean-Yves Berthou,

Richard Kenway, David Keyes

Technology Charge

- Assessing the short-term, medium-term, and long term software and algorithm needs of applications for peta/exascale systems
- Refining the roadmap for software and algorithms on extreme-scale systems
- Setting a prioritized list of software components of exascale computing as outlined in the roadmap

Organization Charge

- Developing a governance, management, and organizational structure for IESP
- Exploring ways for funding agencies to coordinate their support of IESP-related R&D so that they complement each other
- Exploring how laboratories, universities, and vendors can work together on coordinated HPC software
- Creating a plan for working closely with hardware vendors and application teams to co-design future architectures

Execution Charge

- Developing a strategic plan for moving forward with the roadmap
- Creating a realistic timeline for constructing key organizational structures and achieving initial goals
- Exploring community development techniques and risk plans to ensure key components are delivered on time
- Exploring key components of any needed intellectual property agreements

How to pursue the exascale?

- Applications as co-design vehicles

or

- Libraries, kernels, factored components, etc.

Criteria for Co-Design Vehicles (CDVs) [Tsukuba + 2]

- Application running at at least the terascale today, with obvious need for more
- In progressing to the exascale, should be able to achieve scientific goals in its own domain
  - amenability to experimental validation
- Should possess a well-defined set of steps to progress to exascale
- Should have a community supporting the application that has the skills and experience to engage with the co-design process
- Has to be open and should ideally spawn modules for common use
- Overall portfolio should attempt span application space

Different categories of CDVs

- Societally relevant simulations (e.g., climate, patient-specific medicine)
- More likely readily scaled simulations (e.g., QCD, cosmology)
- Data processing problems (e.g., Square Kilometer Array in Australia, which generates 1 EB/s of data and needs FFTs per image while data is streaming)
- Perhaps a “surprise outsider”, not currently practical at the tera-/petascale
- Our breakout does not designate the CDVs, but seeks to identify the features by which the CDV can itself be identified

Issues for scaling up CDVs

- Weak scale application up to distributed memory limits
  - Proportional to number of nodes
- Strong scale application beyond this
  - Proportional to cores per node/memory unit
- Scale the workflow, itself
  - Proportional to the number of instances (ensembles)
  - Integrated end-to-end simulation
- Co-design process is staged, with any of these types of scaling valuable by themselves
- Question: does the software for co-design factor? Or is all the inefficiency at the data copies at interfaces between the components after a while?

Exascale candidates

- Multiscale, multiphysics problems
- Problems that have a superlinear scaling of work with memory capacity
  - for cost- and power-feasible exascale hardware design, which is memory- and communication bandwidth-limited

Building up CDVs

- Capability (scaling up individual codes)
- Complexity (combining multiple codes)
- Scientific or engineering understanding (exploring parameters space for sensitivity, stability, optimization, inversion, etc.)

Software enabling technologies (Tsukuba applications)

Model-related

- Geometric modelers
- Meshers
- Discretizers
- Partitioners
- Solvers / integrators
- Adaptivity systems
- Random number generators
- **Uncertainty quantification**
- **Dynamic load balancing**
- Graphs and combinatorial algorithms
- Compression

Development-related

- Configuration systems
- Source-to-source transformations
- Compilers
- **Programming models**
- Debuggers
- Profilers
- Codesign technologies

Production-related

- Dynamic resource management
- Data management
- Dynamic performance optimization
- **I/O systems**
- Visualization systems
- Workflow controllers
- Frameworks
- Data miners
- **Fault monitoring, application reporting, and recovery**

EOF