U.S. DEPARTMENT OF ENERGY

Briefing to the ASCAC on the Extreme-Scale Solvers Workshop

Karen Pao

Advanced Scientific Computing Research

Karen.Pao@science.doe.gov

27 March 2012

Extreme-Scale Solvers Workshop Summary

"*DOE Workshop on Extreme-Scale Solvers: Preparing for Future Architectures*" was held March 8-9, 2012 at the American Geophysical Union in Washington, DC

Objective: Determine research areas needed for extreme-scale algorithms and software to utilize effectively the 100PF systems and *prepare* for the exascale systems

About 40 researchers from DOE labs, universities, and industry attended the workshop, as well as ASCR and ASC HQ staff

Draft workshop report due 4/10/12 (tentative)

Workshop Website: http://www.orau.gov/extremesolvers2012/

U.S. DEPARTMENT OF ENERGY

Paradigm shift is coming

Systems	Now	Intermediate	Exascale
System peak	2 Peta	100 -- 300 Peta	**1 Exa**
Power	6 MW	~15 MW	**~20 MW**
System memory	0.3 PB	5 PB	**64 PB**
System size (processors)	18,700	50,000 -- 500,000	**O(100,000) -- O(1M)**
Total concurrency	225,000	10’s - 100’s of millions	**10’s - 100’s of billions**
Storage	15 PB	150 PB	**500-1000 PB**
I/O bandwidth	0.2 TB/s	10 TB/s	**60 TB/s**
MTTI	days	O(1day)	**O(1 day)**

Algorithmic challenges have been identified in previous workshops

Select findings from the 2010 Workshop "Scientific Grand Challenges: Crosscutting Technologies for Computing at the Exascale"

- Recast critical applied mathematics algorithms
  - New PDE discretizations to reflect shift from FLOP- to memory-constrained hardware
  - Take advantage of data-movement constraints to redesign UQ and data analysis algorithms and techniques
  - Need to reduced global communication in linear and nonlinear solvers
- Formulate new algorithms to take advantage of emerging architecture
  - New approach to solving conservation laws?
  - Stochastic solutions to counteract fault tolerance?
- Study numerical analysis issues associated with moving away from bulk-synchronous programming
  - Stability and accuracy of asynchronous multiphysics updates
- Need tools to help us understand effects of algorithms on performance
  - Data locality, data locality, data locality!
  - Joule per op?

Scope of workshop

Objective: Determine research areas needed for extreme-scale algorithms and software to utilize effectively the 100PF systems and *prepare* for the exascale systems

- "100 PF" because we have some idea what that machine looks like
  - It will likely be a variation or natural evolution of today's machines
  - In particular, it will likely be heterogeneous due to power constraints
- We are not seeking a Band Aid for the 100 PF machines – we are looking for solutions that have impact and will be relevant beyond the 100 PF machines!
- *Independent of the actual systems designs, s*erious rethinking of today's numerical algorithms for large-scale scientific simulations will be required to deal with common characteristics of these extreme-scale systems, such as
  - Extreme parallelism
  - Data movement
  - Resilience
  - Heterogeneity (maybe)
- Opportunities may exist for the solver community to influence the design of future extreme-scale computers.

U.S. DEPARTMENT OF ENERGY

Organizing Committee

James Ang	SNL	jaang@sandia.gov
Kate Evans	ORNL	evanskj@ornl.gov
Al Geist	ORNL	gst@ornl.gov
Michael Heroux	SNL	maherou@sandia.gov
Paul Hovland	ANL	hovland@mcs.anl.gov
Osni Marques	LBNL	OAMarques@lbl.gov
Lois Curfman McInnes	ANL	curfman@mcs.anl.gov
Esmond Ng	LBNL	EGNg@lbl.gov
Stefan Wild	ANL	wild@mcs.anl.gov

Workshop Website

http://www.orau.gov/extremesolvers2012/

U.S. DEPARTMENT OF ENERGY

Agenda

Day 1			
8:00	Continental Breakfast		
8:20	ASCR Welcome and Introduction	ASCR	Room A
8:30	Charge to Workshop Participants	Organizing Committee	
8:45	Keynote Briefing I	Bob Lucas	
9:15	Keynote Briefing II	Bill Gropp	
9:45	Q&A		
10:00	Break		
10:30	Concurrent Breakout Sessions #1: Architectural Considerations (61 KB)		Rooms A, B, C
12:00	Lunch (on your own)		
13:30	Outbrief #1; Q&A		Room A
14:00	Concurrent Breakout Sessions #2: Algorithmic Research and Development Needs (62 KB)		Rooms A, B, C
15:30	Break		
16:00	Outbrief #2; Q&A		Room A
16:30	Adjourn for the day (dinner on your own)		
Day 2			
8:00	Continental Breakfast		
8:30	Panel Discussion & Q&A	Moderator: Mike Heroux Panelists: TBA	Room A
9:45	Break		
10:00	Concurrent Breakout Sessions #3: Transition Strategy: Gaps (71 KB)		Room A, B, C
11:30	Break		
12:00	Outbrief #3; Q&A		Room A
13:00	Workshop Adjourns		

Attendance

- About 40 attendees from labs, universities, and industry, plus 7 organizers and 7 lab POCs
- SC/ASCR and NNSA/ASC HQ personnel in attendance
- Typical survey from one of the breakouts (three breakouts in each session):

Self-Identification	Count
Numerical algorithms design	11
Numerical methods deployment to HPC	14
HPC numerical library developer	12
User/science application developer	6
Computer scientist	10
Hardware architect	3
Others	

USC Viterbi
School of Engineering

The Sky is Falling

Extreme-scale Challenges for Solvers

Information Sciences Institute

Information Sciences Institute
USC

8 March 2012
Bob Lucas
rflucas@isi.edu

Outline

Today's Execution Model

- What we're doing today

Extreme-scale expectations

- What you need to be worrying about

Expectations for the Future

Seminal DARPA study

- Peter M. Kogge (editor), "Exascale Computing Study: Technology Challenegs in Achieving Exascale Systems", Univ. of Notre Dame, CSE Detp. Tech. Report, TR-2008-13, Sept. 28, 2008

Principle challenges

Concurrency	O(1B ALUs)
Energy	Hundreds of MWs
Memory	Falling off Moore's Law
Resilience	Soft error rate skyrockets

Summary

- Some people will make it to extreme-scale
  - I’m literally betting on Malcolm Stocks
  - I worry about mine
- Solvers will have to change
  - Scalable, perhaps stochastic algorithms
  - Resilient S/W (self checking/correcting)
  - Reason about energy and memory hierarchy
- Solvers need to evolve
  - Don’t unnecessarily throw away working code

Rethinking Solvers for Extreme Scale Architectures

William Gropp
www.cs.illinois.edu/~wgropp
parallel.illinois.edu

1867

Exascale Directions

- Exascale systems are likely to have
  - Extreme power constraints, leading to
    - Clock Rates similar to today's systems
    - A wide-diversity of simple computing elements (simple for hardware but complex for software)
    - Memory per core and per FLOP will be much smaller
    - Moving data anywhere will be expensive (time and power)
  - Faults that will need to be detected and managed
    - Some detection may be the job of the programmer, as hardware detection takes power
  - Extreme scalability and performance irregularity
    - Performance will require enormous concurrency
    - Performance is likely to be variable
      - Simple, static decompositions will not scale
  - A need for latency tolerant algorithms and programming
    - Memory, processors will be 100s to 10000s of cycles away. Waiting for operations to complete will cripple performance

Summary

- Extreme scale architecture *forces* us to confront architectural realities
- Even approximate (yet realistic) performance models can guide development
  - “All models are wrong; some are useful”
- Opportunities abound

Workshop Discussions

For the “solvers” that you are familiar with, what are the research needs from three different angles?

- Architectural considerations
  - What are the most important architectural features for efficient, robust, scalable, and portable solvers for HPC science applications?
  - When should new architectural features be communicated?
- Algorithmic R&D needs
  - What are the most important algorithmic developments needed for efficient, robust, scalable, and portable solvers for HPC science applications?
  - What are some unavoidable requirements and new opportunities for solvers under new architectures?
  - What tools do you need to help you design and implement solvers?
- Transition strategy
  - What are some architecture-specific vs. architecture-agnostic solutions for libraries?
  - What are the evolutions vs revolutions in algorithmic development?

Panel Discussions

- Four workshop participants were volunteered for a panel discussion
  - The panelists were Victor Eijkhout (TAMU/TACC), Robert Falgout (LLNL), Tim Mattson (Intel), and Barry Smith (ANL)
- Panelists answered questions on “game changers” for solvers and “external capabilities” (those that the panelists do not control) necessary for extreme-scale solvers
- Panelists were also asked about what mathematicians and computer scientists should know, beyond their usual knowledge bases, for the era of extreme computing

Summary & Path Forward

Draft workshop report due: 4/10/2012

ASCR POC: Karen Pao

ASC POC: Paul Adamson

Event Support: ORISE (Deneise Terry & Jeannie Robinson)

Workshop Website (include talks by Lucas and Gropp, reference material, and full discussion questions):

http://www.orau.gov/extremesolvers2012/

Discussion Questions (1) Architectural Considerations

- Consider system features needed for efficient, robust, scalable, and portable, high-performance solvers for science applications.
  - What is the most important architectural feature for
    - Scalable solvers?
    - Portable solvers?
    - Fault tolerant, resilient solvers?
    - Robust and efficient solvers?
  - What additional architectural features are desired, and what are absolutely required?
- With respect to high-performance solvers,
  - What are the most important architectural features that need to be settled on first, i.e. code developers need to know as early as possible?
  - What architectural features salient to solvers design, development, and deployment can be decided later by architecture designers? If not “co-designed”, when would code developers need to know about these features (0-3yr, 3-5yr, 5-8yr)?

U.S. DEPARTMENT OF ENERGY

Discussion Questions (2) Algorithmic R & D Needs

- Consider algorithmic features needed to exploit extreme systems.
  - What is the most important algorithmic development for
    - Scalable solvers?
    - Portable solvers?
    - Fault tolerant, resilient solvers?
    - Robust and efficient solvers?
  - What additional algorithmic developments are desired, and what are absolutely required? In particular, the near-term future systems are likely to be heterogeneous. What are the challenges for writing solvers for heterogeneous systems?
- The new architectures will likely enable new and different kinds of scientific simulations.
  - What are some unavoidable requirements for solvers for scientific applications, and what are the necessary capabilities we need to develop?
  - What are the opportunities brought on by the new architectures? What features in solvers will be needed to fully support these new opportunities?
- Code developers will need some help to deal with extreme-scale architectures.
  - Describe needed tools, programming environment, etc., and their uses (no need to “name” them). What are their drivers (e.g., extreme parallelism, data movement minimization, fault tolerance, etc.)?
  - Does the tool exist? How urgently is it needed (0-3 yrs, 3-5 yrs, 5-10 yrs)?

Panel Questions

1. Are there algorithms and problem formulations, presently not widely used, that could be "game changers" and impact solver development plans?
2. What external capabilities (anything you don't control) are absolutely essential (not just nice to have!) for solvers on extreme-scale systems?
3. For the extreme scale computers, everyone in the code chain (mathematicians, computer scientists, code developers) will need to know more about one another's fields and about the machine to be able to write high-performance solvers for science applications
   a) What (minimal set of) architectural features should mathematicians understand?
   b) What areas of knowledge traditionally not considered by mathematicians will become important for the design of numerical algorithms?
   c) How much mathematics should one expect the code developers and the computer scientists to know?
4. When you get back to work on Monday, what are you going to do that will bring us closer to success on future extreme-scale systems?

Discussion Questions (3) Transition Strategy; Gaps

- In the history of HPC, solutions have often been machine-dependent, if not machine specific (e.g., languages, directives, and other quirks).
  - What are some architecture-specific solutions for transitioning current solvers to the extreme scale? What are their respective strengths and weaknesses?
  - What are some possible architecture-independent solutions for transitioning current solvers to the extreme scale? What are their respective strengths and weaknesses?
- We will transition from today's solvers to the extreme-scale algorithms and software needed for these 100 PF systems and architectures of the future.
  - What classes of solvers can be evolved to million-way parallelism? What are the major challenges to this evolution? What is the urgency for this evolution (0-3 yrs, 3-5 yrs, 5-10 yrs)?
  - What classes of solvers are likely to need a revolutionary change? How do we prepare the science community for these revolutionary changes? What is the urgency for this evolution (0-3 yrs, 3-5 yrs, 5-10 yrs)?
- Are there any additional issues that you believe are important and relevant to extreme-scale solvers but haven't been addressed thus far at the workshop?