# Challenges in Computing at Extreme Scale

Esmond G. Ng

Applied Mathematics Department

Lawrence Berkeley National Laboratory

# Outline

- Status of high-speed computing
- Technology trends and implications
- Challenges in the foreseeable future
- Opportunities and more challenges ☺

- Disclaimer: *The views expressed here are solely those of the speaker and do not in any way represent the views of the U.S. Department of Energy or the Lawrence Berkeley National Laboratory.*

# Take-home Message …

**Lots of computational challenges due to trends in extreme-scale computing, but these challenges will lead to opportunities.**

# Current State of Affairs in High-speed Computing …

- Moore's Law:  number of transistors in a dense integrated circuit doubles roughly every two years.

- General agreement that Moore's Law has ended.
  - Number of transistors that can put on a dense integrated circuit does not appear to follow Moore's Law anymore.
  - Clock speed is not increasing, or increasing very slowly.

- Power consumption has become a big concern.

# Future High-performance Computing

- Need for more computing power is increasing.
  - Alternative mode of computing is needed.

- Currently, lots of activities on exascale computing …
  - Clock rate: O(1GHz) → O($10^9$) ops/sec sequential.
  - Exascale: $10^{18}$ ops/sec → O($10^9$) simultaneous ops.
    - **So expect billion-way parallelism …**
  - Many different ways of putting an exascale computer together …

| # parallel intra-nodes | $10^1$ | $10^2$ | $10^3$ | $10^4$ | $10^5$ |
|---|---|---|---|---|---|
| # parallel inter-nodes | $10^8$ | $10^7$ | $10^6$ | $10^5$ | $10^4$ |

# Hardware Challenges in Current Path of Computing

- Increasing number of processing units …
  - Connecting a large number of processing units is complicated.
    - **Impact on communication cost?**
  - Power consumption.
    - **Electric bill is going to be large.**
      - **Some current machines use more than 15 MWatts.**
    - Use of low-energy devices on the rise.
      - Accelerators: GPU's, FPGA's
    - Manycore architectures don't seem to go anywhere
      - Intel just announced the end of Xeon Phi.
      - AMD may provide an alternative???
  - Billion-way parallelism?
    - **Programmability?**

# TOP500 List, June 2018

| Rank | System | Processor | Core Count | Sustainable Perf (TFlop/s) | Peak Perf (TFlop/s) | Power (kW) |
|---|---|---|---|---|---|---|
| #1 | Summit | IBM Power 9 + NVIDIA Volta | 2,282,544 | 122,300.0 | 187,659.3 | 8,806 |
| #2 | Sunway TaihuLight | Sunway SW26010 | 10,649,600 | 93,014.6 | 124,435.9 | 15,371 |
| #3 | Sierra | IBM Power 9 + NVIDIA Volta | 1,572,480 | 71,610.0 | 119,193.6 | |
| #4 | Tianhe-2A | Intel Xeon + Intel Xeon Phi | 4,981,760 | 61,444.5 | 100,678.7 | 18,482 |
| #5 | ABCI | Intel Xeon + NVIDIA Tesla | 391,680 | 19,880.0 | 32,576.6 | 1.649 |

- Three of top 5 machines in the list are GPU-based.
- Out of the top 500 machines, 110 are GPU-based.  The number is expected to grow.

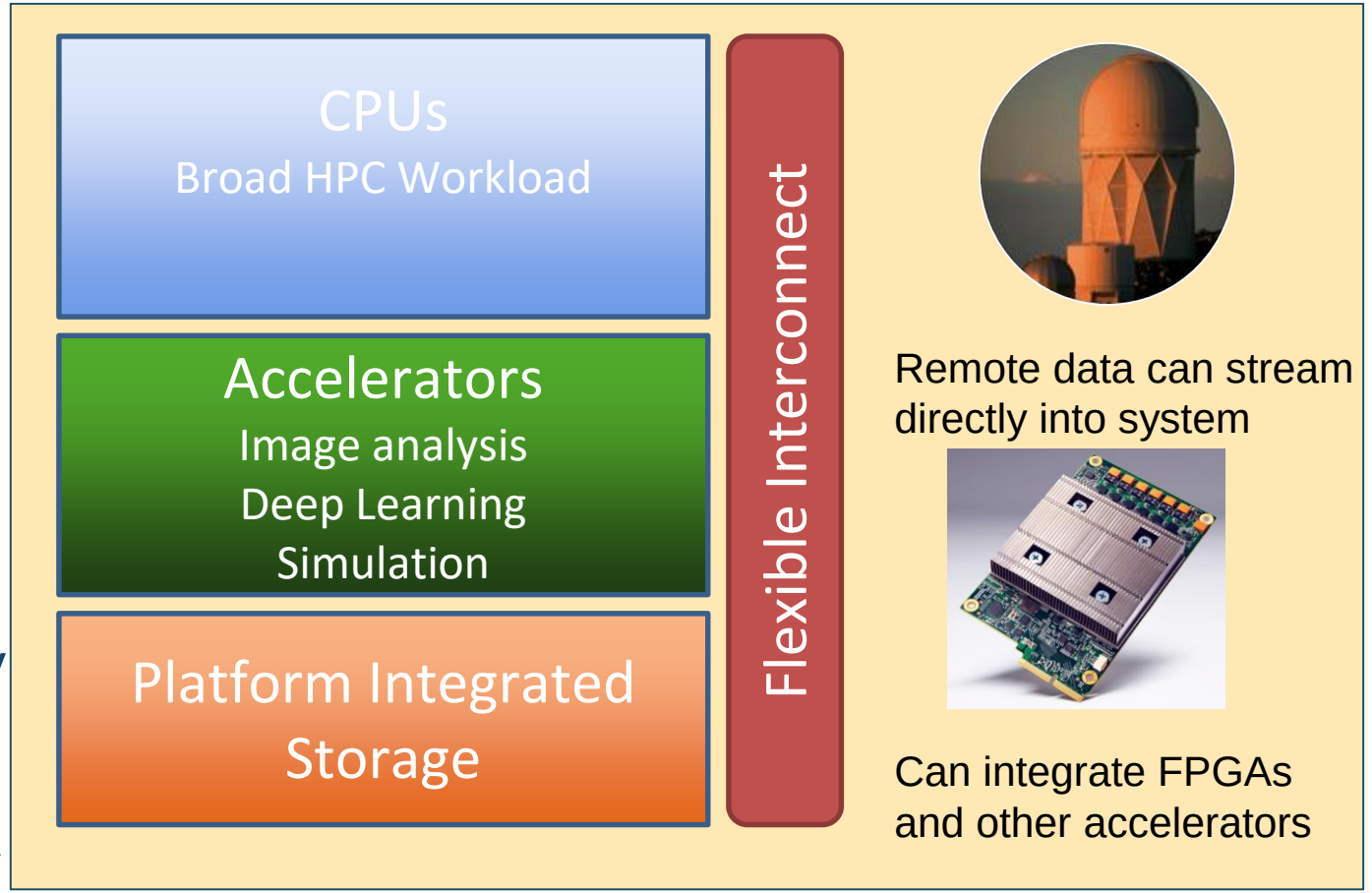
# NERSC-9: A 2020 Pre-Exascale Machine

## Capabilities

- **3-4x capability of Cori**
- **Optimized for both simulation and data analysis**
- **Looks ahead to exascale with specialization and heterogeneity**

## Features

- **Energy Efficient architecture**
  - Large amount of High-BW memory
  - High BW, low latency network
- **Production deployment of accelerators for the DOE community**
- **Single-tier All Flash HPC filesystem**




*System will be announced in 2018*

# Memory Trend

**NERSC System History**

**(NERSC = DOE National Energy Research Scientific Computing Center)**

| Installed | System Name | System Type | Speed | Nodes | SMP Size | Total Cores | Aggregated Memory (GB) | Average Memory/ CPU |
|---|---|---|---|---|---|---|---|---|
| 1997 | MCurie | Cray T3E | 450 MHz | 692 | 1 | 512 | 173 | 256 MB |
| 1999 | GSeaborg | IBM SP | 200 MHz | 256 | 2 | 512 | 256 | 0.5 GB |
| 2001 | Seaborg | IBM SP | 375 MHz | 208 | 16 | 3,328 | 3,328 | 1.0 GB |
| 2005 | Bassi | IBM p575 | 1.9 GHz | 111 | 8 | 888 | 3,552 | 4.0 GB |
| 2007 | Franklin | Cray XT4 | 2.6 GHz | 9,660 | 2 | 19,320 | 38,640 | 2.0 GB |
| 2010 | Hopper | Cray XE6 | 2.1 GHz | 6,384 | 24 | 153,216 | 216,832 | 1.4 GB |
| 2013 | Edison | Cray XC30 | 2.4 GHz | 5,586 | 24 | 134,064 | 365,568 | 2.7 GB |
| 2016 | Cori | Cray XC40 | 1.4 GHz | 9,688 | 68 | 658,784 | 1,048,576 | 1.4 GB |

# Memory Trend

- Memory per core … seems to be decreasing (or not increasing) …
- Current memory/core may be “healthy” for most scientific applications.
- Challenges:
  - Available memory/core unclear.
    - 512 MB/core may be a “luxury”.
  - May have a very deep memory hierarchy.
  - **May need higher core count to get the memory needed or may need to recompute some or all of the time.**
  - Data locality has always been important, but will worsen due to deeper memory hierarchy.
    - **Cost of data transfer?**
    - **Initial data placement in the memory hierarchy important** (unless FLOP rich).
  - **Accelerators – added complexity.**

# Resilience Issues

- If we assume the core/node counts on extreme-scale computers to increase, then MTBF is expected to decrease ☹

- What to do when faults occur?
  - Abort the run?
  - Try to recover from faults (assuming faults can be detected)?

- Size of data sets may limit usage of standard checkpoint/restart.
  - 1M nodes with 64GB per node … How much need to be checkpointed???

# Algorithmic Issues

- **How to realize potentially billion-way parallelism?**
  - intra-node parallelism of O(1K-10K).
  - Inter-node parallelism of O(1,000K-100K).
- Constraints …
  - Within a more complicated memory hierarchy.
  - With reduced relative memory capacity.
  - With potentially complicated communication network (for connecting the processing units).
  - With potentially heterogeneous cores.
  - With decreasing reliability
- **Need to rethink/reformulate algorithms or even the problems to be solved.**

**Computing on future high-performance computers poses new challenges that must be overcome.**

**New algorithms will be needed.**
**Some may be evolutionary, and others may be revolutionary.**

# What are the Computational Challenges?

- High degree of parallelism
- Algorithmic scalability on heterogeneous systems
- Deep memory hierarchy – data movement or communication
- Limited memory size per code
- Resilience

- We will look at some ideas that emerged from recent discussions.
  - Will use mostly linear algebra language …

# Multiple-precision Algorithms

- Facts …
  - Lower precision ops are often faster than higher precision ops.
  - Lower precisions require less memory ⇒ require less data movement.
- Use of multiple precisions is not new …
  - e.g., in the solution of linear systems using iterative refinements (as early as 1970's) for accuracy reasons.
  - But may become more important in extreme scale for data movement and limited memory reasons.
- Open questions …
  - Determining when lower/higher precisions should be used in different parts of a calculation.
  - Reliability, robustness, accuracy of multi-precision algorithms?

# Data Compression

- For data intensive calculations, transfer of data within a deep memory hierarchy may be expensive.
- Would data compression be a reasonable approach to manage the data volume issue?
- For linear algebra …
  - Data compression (including lossy compression) has been employed to reduce storage, operations, and communication.
  - In some cases, compressed data (as opposed to uncompressed data) are used in calculations.
  - Resulted in significant improvement in performance in some cases.
- Open questions …
  - Complexity analysis - Trade off between compression cost and possible reduction in memory?
  - Robustness, reliability, accuracy?

# Randomization and Sampling

- Randomized algorithms have gained quite a bit of popularity in recent years.
  - Not entirely because of extreme-scale computing.
  - But some interesting ideas here.
- Randomization has helped the solution of ill-conditioned problems.
- Sampling has allowed much larger problems to solved.
- Advantages …
  - Do not need entire all the data; just need to be able to generate portion of the data.
  - Completely parallel.
    - Can start different samplings in parallel.
- Open questions …
  - To what extent are randomization and sampling useful?
  - Robustness, reliability, accuracy? What if the approach fails?

# Communication/Synchronization Reduction

- Communication and synchronization are becoming more and more expensive relative to computation, creating bottlenecks in computation.
  - Communication: moving data within the local memory system or across the network in a distributed memory setting.
  - Synchronization:  coordinating computation.
- Known for a long time.
  - Generally agreed that it is important to design algorithms to reduce the amount of communication/synchronization as much as possible.
  - But may become worse at extreme scale.

- A very active area of research in computational mathematics.
  - Many ”new” algorithms have been proposed recently, particularly in linear algebra.

# Comm/Sync Avoiding/Reducing Algorithms

- Notes …
  - Some of the ideas in some of these algorithms are not entirely new, but being re-discovered.
  - It's often the case that such algorithms may require more memory and/or more computation.
  - Some algorithms have communication/synchronization complexities that match theoretical lower bounds.
  - Some algorithms may not be as stable as conventional algorithms.
- Open questions …
  - New algorithms for other computation that require less communication/synchronization?
  - Can existing algorithms be reformulated to reduce communication/synchronization?
  - Numerical behavior of such algorithms?

# Fine-grained Parallel Algorithms

- Extreme-scale computing promises high degree of parallelism.
- Fine-grained parallel algorithms for computation?
  - Not entirely new.
  - Theoretical computer scientists talked about this mode of computation in the 1970's and 1980's.
- Probably need to come up with out-of-the-box ideas.

- Recent example …
  - In a sparse matrix computation, each nonzero matrix element was assigned to one processing unit, and new parallel algorithms were created.
  - Applicable to other computation?
    + Also need to worry about amount of communication and sychronization.

# Resilience

- Resilience is concerned with dealing with and recovering from faults.
- Assume a fault has occurred and assume that it can be detected.
  - Then some unknowns that were being computed would be missing.
  - Attempt to reconstruct or approximate those missing values using known relationships and based on values that have most recently computed successfully.
- Open problems …
  - Resilient algorithms for what problems?
    - Possible for linear algebra problems.
  - Numerical behavior of such algorithms?
  - What to do if recovery fails?

# Summary

- Challenges along the path to extreme scale …
  - High degree of parallelism
  - High communication & synchronization overhead
  - Deep memory hierarchy
  - Limited memory
  - Resilience

- What we need to overcome these challenges … more research
  - Some existing approaches may evolve
  - Re-visit old ideas
  - Need new and out-of-the-box ideas

# Summary

- Some possible research opportunities …
  - Multiple-precision algorithms.
  - Use of data compression.
  - Algorithms based on randomization and sampling.
  - Communication and synchronization avoiding/reduction algorithms.
  - Fine-grained parallel algorithms.
  - Resilient algorithms.
- One of the common themes …
  - Robustness, reliability, accuracy.
    + Numerical algorithms have been based on well-understood error analysis and stability analysis.
    + We should not expect anything less as we move towards extreme-scale computing.

# Summary

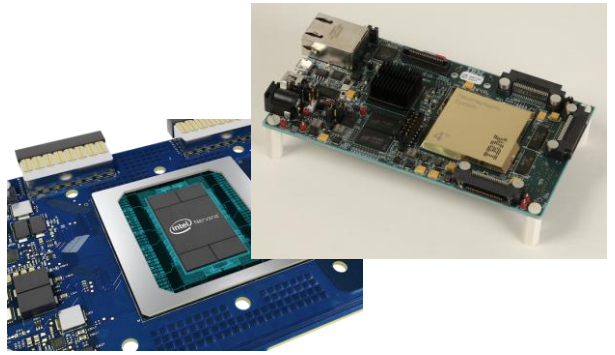
- This talk only scratches the surface of extreme-scale computing.
- Many things were not discussed …

- Other post-Moore architectures …
  - Quantum computing
  - Neuromorphic computing?

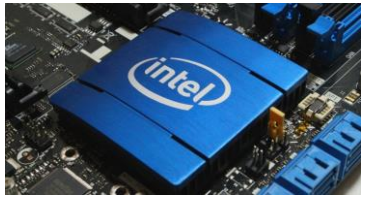
# Exploring Extreme Heterogeneity - Deploying Workflow Accelerators for SC Applications with NERSC-9

- *What accelerators map to existing SC workloads? And what SC challenges could be solved with emerging accelerators?*
- Key areas of investigation
  - Identify common algorithms, kernels, motifs that run well on emerging accelerators
  - Determine feasibility of configurable processing technologies, e.g. FPGAs
  - Analyze changing workload requirements, e.g. ML, EOD/Superfacility
- Will NERSC-10 be the last exascale machine or the first Beyond-Moores ?
- How many heterogeneous elements should NERSC-10 contain deploy and why ? Where will the specialized hardware that meets the needs of NERSC/DOE come from?

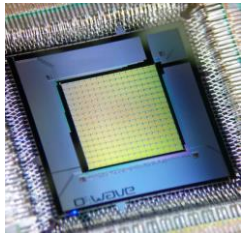
Neural Network Processors



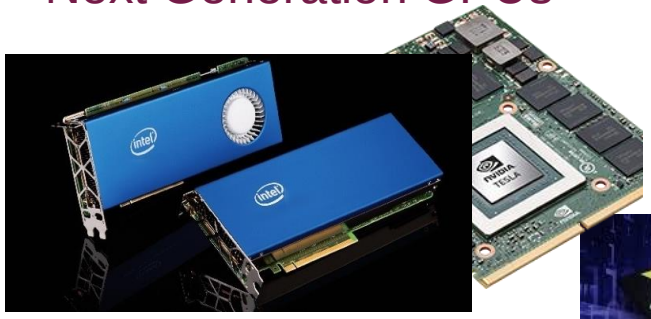
Emerging Technologies


Quantum Computing



Next Generation GPUs



Programmable Arrays

*Thank You!*

Contact: Esmond G. Ng (EGNg@lbl.gov)