Exa-Scale FSIO
Can we get there?
Can we afford to?

07/2010
Gary Grider, LANL

LA-UR 10-04611
Mission Drivers

- **Climate Change:** Understanding, mitigating and adapting to the effects of global warming
  - Sea level rise
  - Severe weather
  - Regional climate change
  - Geologic carbon sequestration

- **Energy:** Reducing U.S. reliance on foreign energy sources and reducing the carbon footprint of energy production
  - Reducing time and cost of reactor design and deployment
  - Improving the efficiency of combustion energy sources

- **National Nuclear Security:** Maintaining a safe, secure and reliable nuclear stockpile
  - Stockpile certification
  - Predictive scientific challenges
  - Real-time evaluation of urban nuclear detonation

Accomplishing these missions requires exascale resources.
Power is a Driving Issue

- Power per flop
- Power per byte
- Power per byte/sec
- Power for infrastructure
- POWER POWER POWER
Memory is a Big Problem

- Power Consumption with standard Technology Roadmap
- Power Consumption with Investment in Advanced Memory Technology

**Cost**

- Cost in $M (8 gigabit modules)
- Cost in $M (16 Gigabit modules)
- 1/2 of $200M system

**Power**

- 70 Megawatts total
- 20 Megawatts total

- FPU
- Memory Interconnect
Parallelism will be Massive

How much parallelism must be handled by the program?
From Peter Kogge (on behalf of Exascale Working Group), "Architectural Challenges at the Exascale Frontier", June 20, 2008

Need 1Million-way parallelism to reach an Exaflop . . .
And possibly another 100x just to hide latency
Technology Roadmap

- Improved hardware and software reliability
- New IO technology
- 3D chip-level integration
- New programming model
- Improved interconnect technology
- Demonstrate > 3X power efficiency gain
- SW scalability to 100M threads
- Latency tolerant algorithms
- Demonstrate > 3X power efficiency gain over 2015
- 10X memory BW
- Improved resilience through local recovery and migration
- Application scalability to 1B threads

Exascale Science

Net Throughput
It is a Complicated Trade Space

- 20 MW power envelope
- Bytes/core envelope
- $200M cost envelope
- Exascale Performance envelope

nodes

memory

feasible system
Reliability will be Difficult

- Industry must maintain constant FIT rate per node
  - ~1000 failures in time
- Moore’s law gets us 100x improvement
  - But still have to increase node count by 10x
- So we will own 10x worse FIT rate
  - MTTI 1 week to 1 day
  - MTTI 1 day to 1 hour

Figure 2. Failures in billions of hours of operation.2-5
<table>
<thead>
<tr>
<th>System attributes</th>
<th>2010</th>
<th>“2015”</th>
<th>“2018”</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Peta</td>
<td>200 Petaflop/sec</td>
<td>1 Exaflop/sec</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>15 MW</td>
<td>20 MW</td>
</tr>
<tr>
<td>System memory</td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>32-64 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>125 GF</td>
<td>0.5 TF</td>
<td>7 TF</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>25 GB/s</td>
<td>0.1 TB/sec</td>
<td>1 TB/sec</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>12</td>
<td>O(100)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>System size</td>
<td>18,700</td>
<td>50,000</td>
<td>5,000</td>
</tr>
<tr>
<td>(nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Node</td>
<td>1.5 GB/s</td>
<td>20 GB/sec</td>
<td></td>
</tr>
<tr>
<td>Interconnect BW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTTI</td>
<td>days</td>
<td>O(1day)</td>
<td>O(1 day)</td>
</tr>
</tbody>
</table>
Gloom and Doom from 2006/

• Petascale computing is coming
  – Orders of magnitude more components
  – Orders of magnitude more failures

• Need raw data for better understanding of failures
Past and Future Assumptions

• Past
  – All disk
  – Constant ratio of total $ to IO infra $
  – Machines wont accelerate their reliability per flop

• Future
  – Not necessarily all disk
  – Not necessarily same % but close
  – Machines may make accelerate progress on reliability/flop due to integration and industry desire to have constant reliability per socket
Can we do defensive IO at Exascale?

• If we loosen assumptions?
• If we can do it can we afford to do it?
I wanted to know – what miracles will we need and to get past what problems.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>EF</th>
<th>2010</th>
<th>2012</th>
<th>2014</th>
<th>2016</th>
<th>2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td></td>
<td>1.00</td>
<td>20.00</td>
<td>200.00</td>
<td>400.00</td>
<td>1000.00</td>
<td></td>
</tr>
<tr>
<td>mem low PB</td>
<td></td>
<td>0.004</td>
<td>0.07</td>
<td>0.72</td>
<td>1.44</td>
<td>3.60</td>
<td></td>
</tr>
<tr>
<td>mem med PB</td>
<td></td>
<td>0.020</td>
<td>0.40</td>
<td>4.00</td>
<td>8.00</td>
<td>20.00</td>
<td></td>
</tr>
<tr>
<td>mem high PB</td>
<td></td>
<td>0.300</td>
<td>6.00</td>
<td>60.00</td>
<td>120.00</td>
<td>300.00</td>
<td></td>
</tr>
<tr>
<td>Num Full Mem Cap</td>
<td></td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Size Scratch PB low</td>
<td></td>
<td>0.108</td>
<td>2.16</td>
<td>21.60</td>
<td>43.20</td>
<td>108.00</td>
<td></td>
</tr>
<tr>
<td>Size Scratch PB med</td>
<td></td>
<td>0.600</td>
<td>12.00</td>
<td>120.00</td>
<td>240.00</td>
<td>600.00</td>
<td></td>
</tr>
<tr>
<td>Size Scratch PB high</td>
<td></td>
<td>9.000</td>
<td>180.00</td>
<td>1800.00</td>
<td>3600.00</td>
<td>9000.00</td>
<td></td>
</tr>
<tr>
<td>Time to dump Secs</td>
<td></td>
<td>1200.00</td>
<td>800.00</td>
<td>600.00</td>
<td>400.00</td>
<td>300.00</td>
<td></td>
</tr>
<tr>
<td>Ckpt BW low TB/s</td>
<td></td>
<td>0.003</td>
<td>0.09</td>
<td>1.20</td>
<td>3.60</td>
<td>12.00</td>
<td></td>
</tr>
<tr>
<td>Ckpt BW med TB/s</td>
<td></td>
<td>0.017</td>
<td>0.50</td>
<td>6.67</td>
<td>20.00</td>
<td>66.67</td>
<td></td>
</tr>
<tr>
<td>Ckpt BW high TB/s</td>
<td></td>
<td>0.250</td>
<td>7.50</td>
<td>100.00</td>
<td>300.00</td>
<td>1000.00</td>
<td></td>
</tr>
<tr>
<td>Disk Capacity TB</td>
<td></td>
<td>2.000</td>
<td>3.92</td>
<td>7.68</td>
<td>15.06</td>
<td>29.52</td>
<td></td>
</tr>
<tr>
<td>Disk Speed MB/s</td>
<td></td>
<td>100</td>
<td>100.00</td>
<td>140.00</td>
<td>196.00</td>
<td>274.40</td>
<td>384.16</td>
</tr>
<tr>
<td>IO node thrput GB/s</td>
<td></td>
<td>100</td>
<td>1.000</td>
<td>2.000</td>
<td>4.000</td>
<td>8.000</td>
<td>16.000</td>
</tr>
</tbody>
</table>

New Assumptions

- DARPA Exa Study for machine sizes, mtti, etc. except 20 PB med mem machine and 30 dumps in scratch
- Seagate Disk Capacity/Size/Pricing/Power (not shown)
- Micron Flash Capacity/Size/Pricing/Power (not shown)
- 10% of mtti as dump time
Status Quo: Use Disk Based Shared Global Parallel File System to Provide Dump Space

Notice that using these modeling parameters, we finally reach the predicted cross over point of buying disk for BW and not Capacity in 2012

- Buying disk for capacity is reasonably priced but buying disk for bandwidth gets expensive fast!

2018 medium memory machine
- **4166** IO nodes, 175k disks
- File System sees 50-100k way parallelism (assumes IOFSL)
- **$225M** pessimistic purchase (assumes no technologies pushing disk other than Flash)
- Power **1.5MWatts**

Miracle Needed!
Use MLC Based Shared Global Parallel File System to Provide Dump Space

Notice that buying MLC for capacity is expensive but buying it for Bandwidth is cheaper

2018 medium memory machine
• 4166 IO nodes
• File System sees 50-100k way parallelism (assumes IOFSL)
• $625M pessimistic purchase (assumes no technologies pushing disk other than Flash)
• Power 2.5MWatts (have to buy so much to get capacity)
Lets Try to Buy Disks for Capacity and MLC for Bandwidth == Hybrid Model

• Twin tailed non global MLC connected to NN compute nodes N I/O Nodes, compute nodes dump to MLC at 10% MTTI time and IO nodes bleed to global disk without causing jitter at 1/10\textsuperscript{th} the dump-burst data rate or less

\begin{figure}
\centering
\includegraphics[width=\textwidth]{hybrid_model.png}
\caption{Devices Needed by Year for Low Mem Option}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{disks_needed.png}
\caption{Disks Needed by Year for Low Mem Option}
\end{figure}

• 3 memory dumps in MLC
• 30 dumps in global disk
Hybrid MLC burst / Disk Global

Hybrid wins big once we are in the regime of having to buy disk for bandwidth.

All Global disk is cheapest now because we buy capacity.

2018 med mem mach

- 416 IO Nodes, 20k disks not much of a stretch
- Disk FS sees modest parallelism assumes IOFSL/burstbuffer etc.
- $60M pessimistic purchase - worst case (all migrated to disk and tech price)
- Power 2.2MWatts
Hybrid MLC burst / Disk Global

Cost Driver Sensitivity
• More MLC BW (free – capacity driver)
• More MLC Cap (costly – capacity driver)
• Less Disk Cap (small savings (MLC capacity driver)
• Less Disk BW (small savings controllers/ION etc. (MLC capacity driver)
• Less Frequent MLC to Disk (no savings, Disk Capacity Driver)
Hybrid Disk/SSD

Must move checkpoint device closer to compute memory
- on node – has jitter issues
- at least near node is required
- Leads to Hybrid Storage model
## A Feasible Evolutionary Approach?

<table>
<thead>
<tr>
<th>Summary:</th>
<th>Issue</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Probably pretty close on storage densities, bandwidths, and costs, in fact it may be a bit conservative (maybe more than a bit)</td>
<td>Continue to update model</td>
</tr>
<tr>
<td></td>
<td>Based heavily on MTTI assumptions in the DARPA study and that study indicates a pretty large per socket improvement in MTTI without good substantiation</td>
<td>Get serious about measuring and predicting this!</td>
</tr>
<tr>
<td></td>
<td>Assumes that existing techniques like RAID or other redundant techniques will keep the burst buffer working often enough to not have issues without substantiation</td>
<td>Keep our eye on Flash reliability – prospects are good given wide use</td>
</tr>
<tr>
<td></td>
<td>Assumes existing RAS techniques for file systems will be able to keep up without substantiation</td>
<td>Keep our eye on this</td>
</tr>
<tr>
<td></td>
<td>Have to have burst buffer so we will need software to manage MLC burst buffer, with bleed to global disk</td>
<td>SCR LLNL / PLFS LANL / ADIOS ORNL / MPI-IO ANL. Zest PSC, ...</td>
</tr>
<tr>
<td></td>
<td>Assumes flattening to get high % of peak on disks (like log structure)</td>
<td>PLFS LANL / ADIOS ORNL / MPI-IO ANL, Zest PSC, ...</td>
</tr>
<tr>
<td></td>
<td>Need a way to deal with large numbers of files</td>
<td>Giga+, etc.</td>
</tr>
</tbody>
</table>
Maybe we can get to Exascale with evolution only, but it would be pretty sad if we didn’t also attempt some more fundamental revolutionary approaches!
NSF PRObE:
A Systems Research Center

NMC, LANL, CMU, Utah
LANL was going to trash this!
NMC/LANL/CMU/Utah/NSF

to the Rescue

• NSF funds New Mexico Consortium for PRObE
  – Parallel Reconfigurable Observational Environment
  – New Mexico Consortium (NMC)
    • University of New Mexico, New Mexico Institute of Mining and Technology, New Mexico State University
  – Los Alamos National Laboratory
  – Carnegie Mellon University
  – University of Utah

• Large scale clusters for systems researchers
  – For dedicated use, long periods of time (days, weeks)
  – Allow replacement of any and all software
  – Maybe destructive fault injection too
Hardware Plan

• Fall 2011: Sitka (2048 cores)
  – 1024 Nodes, Dual Socket, Single Core AMD Opteron; 2 GB RAM per core; Full fat-tree Myrinet high-speed interconnect

• Fall 2012: Kodiak (2048 cores)
  – 1024 Nodes, Dual Socket, Single Core AMD Opteron; 4 GB RAM per core; Fat-tree SDR Infiniband high-speed interconnect

• Fall 2013: Nome (1600 cores)
  – 200 Node, Quad Socket, Dual Core AMD Opteron; 2 GB RAM per core; Fat-tree DDR Infiniband high-speed interconnect
Hardware Plan II

• Small (bought new) high-core-count clusters
• Summer 2011: Susitna (1728 cores)
  – 36 Nodes, Quad Socket, 12 core AMD Magny-Cours (or similar); 1-2GB RAM / core; EDR Infiniband high-speed interconnect
• Summer 2013: Matanuska (3456 cores)
  – 36 Nodes, Quad Socket, 24 core AMD Magny-Cours (or similar); 1-2GB RAM / core; 100 GigaBit Ethernet (or similar)
Two Large Machine Hardware Environment

1024 Node Myrinet Cluster

256 Dual Socket AMD64 Cluster

Front End

1024 Node Infiniband Cluster

256 Dual Socket AMD64 Cluster

Front End

Internet Gateway and public facing web/wiki presence

To Internet

128 Dual Socket AMD64 Cluster

Centralized Permanent Storage (100s of TB)

Software Environment: Emulab Software Environment enhanced for scaling to two 1024 node clusters, exploitation of Infiniband, with remote power and console management, and system partitioning capability

Ethernet Switch

10 GE

10GE

10 GE
For Systems Research Users

- NSF “who can apply” rules
  - Includes international and corporate research

PRObE Target Communities

- High End Computing (HEC) Systems Community: Supercomputing Conference
- HEC and DC Storage Systems Community: FAST Conference
- Data-intensive Computing (DC) Systems Community: OSDI Conference
PRObE Decision Making

• Committees usually about 6, selected by standard academic procedures (via BOFs)
Software

• First, “none” is allowed
  – Researchers can put any software they want onto the clusters
  – Full OpenCirrus stack possible

• Second, a well known tool managing clusters of hardware for research
  – Emulab (www.emulab.org), Flux Group, U. Utah
  – Widely used in academic systems community
  – Enhanced for PRObE hardware, scale, networks, resource partitioning policies, remote power and console, failure injection, deep instrumentation
Relationship with OpenCirrus

• Do not compete
  – Proposed uses may sometimes be more appropriate on resources of the others’ systems

• Make PRObE friendly to OpenCirrus users
  – CMU as a liaison to demonstrate OpenCirrus code on PRObE
  – OpenCirrus users wanting to move to bigger, dedicated, deeply controllable resources

• Your idea here .....