ExaFlash:
An HPC system for 2015

Allan Snavely
ExaFlash Baseline

32 nm “aggressive silicon strawman” from:

- **ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems** Peter Kogge, Bill Dally, Shekar Borkhar, Thomas Sterling, Allan Snavely et al

- **We start with an FPU (along with its registers)**
  - 4 FPUs and L1 forms a **Core** 742 such cores yields a 4.5TFlops, 150W active power (215W total) chip. This chip along with 16 memory slots forms a **Node** with 16GB of DRAM. 12 nodes plus routing and (mystery component) forms a **Group**, 32 Groups are packaged into a **1.7 Pflops 116KW Cabinet**.
  - Our total system is then 8 racks i.e. **13 Pflops in a 1 MW**
Exaflash Cabinet

1 Cabinet Contains 32 Groups on 12 Networks
## Summary characteristics

<table>
<thead>
<tr>
<th>Level</th>
<th>What</th>
<th>Perf</th>
<th>Power</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPU</td>
<td>FPU, regs., Instruction-memory</td>
<td>1.5 Gflops</td>
<td>30mW</td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>4FPUs, L1</td>
<td>6 Gflops</td>
<td>141mW</td>
<td></td>
</tr>
<tr>
<td>Processor Chip</td>
<td>742 Cores, L2/L3, Interconnect</td>
<td>4.5 Tflops</td>
<td>214W</td>
<td></td>
</tr>
<tr>
<td>Node</td>
<td>Processor Chip, DRAM</td>
<td>4.5 Tflops</td>
<td>230W</td>
<td>16GB</td>
</tr>
<tr>
<td>Group</td>
<td>12 Processor Chips, routers</td>
<td>54 Tflops</td>
<td>3.5KW</td>
<td>192GB</td>
</tr>
<tr>
<td>rack</td>
<td>32 Groups</td>
<td>1.7 Pflops</td>
<td>116KW</td>
<td>6.1 TB</td>
</tr>
<tr>
<td>System</td>
<td>583 racks</td>
<td>1 Eflops</td>
<td>67.7MW</td>
<td>3.6PB</td>
</tr>
</tbody>
</table>
Mystery component

- To provide checkpoint storage, scratch space, and archival storage 16 NAND flash drives are associated with each group. These drives are located in the rack and attached via high-speed serial channels (the evolution of SATA) to the routers in the group. They are accessible via the network from any processor in the system. Using projections for 2014 flash the 16 drives will provide an aggregate of 64TB of storage, 64GB/s of bandwidth, 1 M IOPS and dissipate about 50W.
More “Special Sauce”
I claim all the below should be commodity in 2015

1. Dynamic power management to stay under 1 MW
   • The system can devote all its energy budget to the level of memory hierarchy that is the performance bottleneck

2. Reconfigurable memory hierarchy
   • The system does not move 1 unneeded byte and no data further than required. Leverages FULL-EMPTY BITS
   • Saves a lot of power!

3. Integrated hardware software stack
   • The two above require a hardware/software integrated stack that cooperates to detect and exploit locality
## Power Budget and Bandwidth Tapers

<table>
<thead>
<tr>
<th>Item</th>
<th>Percent</th>
<th>Watts</th>
<th>Units</th>
<th>BW (GW/s)</th>
<th>Taper</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPU's</td>
<td>59.0%</td>
<td>88.5</td>
<td>2968</td>
<td>4495</td>
<td>1</td>
<td>Includes 3-port reg and I-mem</td>
</tr>
<tr>
<td>L1 Data</td>
<td>10.9%</td>
<td>16.4</td>
<td>742</td>
<td>1124</td>
<td>4</td>
<td>64KB per 4 FPU's</td>
</tr>
<tr>
<td>L2</td>
<td>6.9%</td>
<td>10.4</td>
<td>371</td>
<td>562</td>
<td>8</td>
<td>256KB per 2 L1s</td>
</tr>
<tr>
<td>L3</td>
<td>7.5%</td>
<td>11.3</td>
<td>189</td>
<td>286</td>
<td>16</td>
<td>Global access to L2s</td>
</tr>
<tr>
<td>DRAM</td>
<td>10.0%</td>
<td>15.0</td>
<td>59</td>
<td>89</td>
<td>50</td>
<td>Attached to this chip</td>
</tr>
<tr>
<td>Network</td>
<td>5.6%</td>
<td>8.4</td>
<td>13</td>
<td>27</td>
<td>164</td>
<td>Global access</td>
</tr>
</tbody>
</table>

Taper = number of flops executed per access

1 MW
Increase tapers and stay within power budget

- To stay under 1MW, monitor overall power use (e.g. by counting the number of accesses at each level of the hierarchy) and throttle instruction issue.
- An application that demands very high DRAM bandwidth and little floating-point can use all of its power on DRAM bandwidth.
- This widens DRAM interface by an order of magnitude to a bandwidth of nearly 900GWords/s for full system.
Dynamic memory reconfiguration

- Smart controller less than typical FPGA complexity and cache can be
  - Scratchpad
  - Read-only
  - Specified line length and associativity
  - Prefetched or not prefetched
  - Synchronized on full empty-bits
Compiler:
Inputs: locality descriptions (programmer) Outputs: memory footprint, shared resource demands

PL
Inputs: Locality information (programmer etc.)
Outputs: Locality representation

Access patterns re: data structures, memory footprints, coo. patterns
Symbiosis information

Runtime
Inputs: symbiosis info., memory access pattern info.
Outputs: instructions to prefetch, cache, hardware resource allocations

Cacheing and prefetching instructions, schedule

Hardware
Inputs: instructions to cache or not, thread bindings etc.
Outputs: hardware counters, events, address stream

Global Data Motion optimization viewed as a “feedback and control” problem to be solved by cooperation among layers of the hardware/software stack.

Locality information

Performance tools
Inputs: address stream, communications and I/O patterns, etc. Outputs: locality info.

Optimal event reorderings
Physical address stream
• **FFT**
  - Use 3D FFT due to Andrew Canning, 75% of peak (9+ Pflops)

• **Graph Problem**
  - Fill memory runs at (effectively) true GUPS rate i.e. about $10^{14}$ memops

• **Social Network database queries**
  - Fill SSD system runs at 1 million IOPS per Group, or $32 \times 8 = 256$ million random IOPS
Dash
Exaflash prototype winner of SC09 Data Challenge

- Prototype of *Cost-effective Data-performance* for a *Data-Intensive Supercomputer (Gordon)*
- Move scientific data 10x closer to compute
- Random IOPS > 10x higher than spinning disk for random-walks through scientific databases
- Provide a balanced system for data with ratio addressable memory bytes to flops > 1
  - 5.2 Tflops, 3 TB DRAM, 4 TB flash
- Ease of shared memory programming and outreach to emerging scientific communities
Questions?