Are mobile processors ready for HPC?

Nikola Rajovic, Pall Carpenter, Isaac Gelado, Nikola Puzovic, Alex Ramirez

Barcelona Supercomputing Center

http://www.montblanc-project.eu
First, vector processors dominated HPC

• 1st Top500 list (June 1993) dominated by DLP architectures
  • Cray vector, 41%
  • MasPar SIMD, 11%
  • Convex/HP vector, 5%
• Fujitsu *Wind Tunnel* is #1 1993-1996, with 170 GFLOPS

*Mont-Blanc*
Then, commodity took over special purpose

- **ASCI Red, Sandia**
  - 1997, 1 TFLOPS
  - 9,298 cores @ 200 Mhz
  - Intel Pentium Pro
    - Upgraded to Pentium II Xeon,
      1999, 3.1 TFLOPS

- **ASCI White, LLNL**
  - 2001, 7.3 TFLOPS
  - 8,192 proc. @ 375 Mhz,
  - IBM Power 3

Transition from Vector parallelism to Message-Passing Programming Models
Commodity components drive HPC

- RISC processors replaced vectors
- x86 processors replaced RISC
  - Vector processors survive as (widening) SIMD extensions
The killer microprocessors

- Microprocessors killed the Vector supercomputers
  - They were not faster...
  - ...but they were significantly cheaper and greener

- Need 10 microprocessors to achieve the performance of 1 Vector CPU
  - SIMD vs. MIMD programming paradigms
The killer mobile processors™

- Microprocessors killed the Vector supercomputers
  - They were not faster ...
  - ... but they were significantly cheaper and greener

- History may be about to repeat itself …
  - Mobile processor are not faster …
  - … but they are significantly cheaper

**Graph:**
- MFLOPS
- Alpha
- Intel
- AMD
- NVIDIA Tegra
- Samsung Exynos
- 4-core ARMv8 1.5 GHz

**Legend:**
- 100
- 1,000
- 10,000
- 100,000
- 1,000,000
Mobile SoC vs Server processor

Performance

- 5.2 GFLOPS
- 153 GFLOPS
- 15.2 GFLOPS

Cost

- 21$\textsuperscript{1}
- 1500$\textsuperscript{2}
- 21$ (?)

1. Leaked Tegra3 price from the Nexus 7 Bill of Materials
2. Non-discounted List Price for the 8-core Intel E5 SandyBridge
SoC under study: CPU and Memory

NVIDIA Tegra 2
2 x ARM Cortex-A9 @ 1GHz
1 x 32-bit DDR2-333 channel
32KB L1 + 1MB L2

NVIDIA Tegra 3
4 x ARM Cortex-A9 @ 1.3GHz
2 x 32-bit DDR2-750 channels
32KB L1 + 1MB L2

Samsung Exynos 5 Dual
2 x ARM Cortex-A15 @ 1.7GHz
2 x 32-bit DDR3-800 channels
32KB L1 + 1MB L2

Intel Core i7-2760QM
4 x Intel SandyBridge @ 2.4GHz
2 x 64-bit DDR3-800 channels
32KB L1 + 1MB L2 + 6MB L3
## Evaluated kernels

<table>
<thead>
<tr>
<th>Tag</th>
<th>Full name</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>vecop</td>
<td>Vector operation</td>
<td>Common operation in numerical codes</td>
</tr>
<tr>
<td>dmmmm</td>
<td>Dense matrix-matrix multiply</td>
<td>Data reuse an compute performance</td>
</tr>
<tr>
<td>3dstdc</td>
<td>3D volume stencil</td>
<td>Strided memory accesses (7-point 3D stencil)</td>
</tr>
<tr>
<td>2dcon</td>
<td>2D convolution</td>
<td>Spatial locality</td>
</tr>
<tr>
<td>fft</td>
<td>1D FFT transform</td>
<td>Peak floating-point, variable stride accesses</td>
</tr>
<tr>
<td>red</td>
<td>Reduction operation</td>
<td>Varying levels of parallelism</td>
</tr>
<tr>
<td>hist</td>
<td>Histogram calculation</td>
<td>Local privatization and reduction stage</td>
</tr>
<tr>
<td>msort</td>
<td>Generic merge sort</td>
<td>Barrier synchronization</td>
</tr>
<tr>
<td>nbody</td>
<td>N-body calculation</td>
<td>Irregular memory accesses</td>
</tr>
<tr>
<td>amcd</td>
<td>Markov chain Monte-Carlo method</td>
<td>Embarassingly parallel</td>
</tr>
<tr>
<td>spvm</td>
<td>Sparse matrix-vector multiply</td>
<td>Load imbalance</td>
</tr>
</tbody>
</table>
Single core performance and energy

- Tegra3 is 1.4x faster than Tegra2
  - Higher clock frequency
- Exynos 5 is 1.7x faster than Tegra3
  - Better frequency, memory bandwidth, and core microarchitecture
- Intel Core i7 is ~3x better than ARM Cortex-A15 at maximum frequency
- ARM platforms more energy-efficient than Intel platform
Multicore performance and energy

- Tegra3 is as fast as Exynos 5, a bit more energy efficient
  - 4-core vs. 2-core
- ARM multicores as efficient as Intel at the same frequency
- Intel still more energy efficient at highest performance
  - ARM CPU is not the major power sink in the platform
• Exynos 5 improves dramatically over Tegra (4.5x)
  • Dual-channel DDR3
  • ARM Cortex-A15 sustains more in-flight cache misses
Tibidabo: The first ARM HPC multicore cluster

- **Q7 Tegra 2**
  - 2 x Cortex-A9 @ 1GHz
  - 2 GFLOPS
  - 5 Watts (?)
  - 0.4 GFLOPS / W

- **Q7 carrier board**
  - 2 x Cortex-A9
  - 2 GFLOPS
  - 1 GbE + 100 MbE
  - 7 Watts
  - 0.3 GFLOPS / W

- **1U Rackable blade**
  - 8 nodes
  - 16 GFLOPS
  - 65 Watts
  - 0.25 GFLOPS / W

- **2 Racks**
  - 32 blade containers
  - 256 nodes
  - 512 cores
  - 9x 48-port 1GbE switch
  - 512 GFLOPS
  - 3.4 Kwatt
  - 0.15 GFLOPS / W

- **Proof of concept**
  - It is possible to deploy a cluster of smartphone processors

- **Enable software stack development**
HPC System software stack on ARM

- Open source system software stack
  - Ubuntu Linux OS
  - GNU compilers
    - gcc, g++, gfortran
  - Scientific libraries
    - ATLAS, FFTW, HDF5,...
  - Slurm cluster management

- Runtime libraries
  - MPICH2, OpenMPI
  - **OmpSs toolchain**

- Performance analysis tools
  - Paraver, Scalasca

- Allinea DDT 3.1 debugger
  - Ported to ARM
Parallel scalability

- HPC applications scale well on Tegra2 cluster
  - Capable of exploiting enough nodes to compensate for lower node performance
SoC under study: Interconnection

- **NVIDIA Tegra 2**
  - 1 GbE (PCIe)
  - 100 Mbit (USB 2.0)

- **NVIDIA Tegra 3**
  - 1 GbE (PCIe)
  - 100 Mbit (USB 2.0)

- **Samsung Exynos 5 Dual**
  - 1 GbE (USB3.0)
  - 100 Mbit (USB 2.0)

- **Intel Core i7-2760QM**
  - 1 GbE (PCIe)
  - QDR Infiniband (PCIe)
• TCP/IP adds a lot of CPU overhead
• OpenMX driver interfaces directly to the Ethernet NIC
• USB stack adds extra latency on top of network stack

Thanks to Gabor Dozsa and Chris Adeniyi-Jones for their OpenMX results
Interconnection network: Bandwidth

- TCP/IP overhead prevents Cortex-A9 CPU from achieving full bandwidth
- USB stack overheads prevent Exynos 5 from achieving full bandwidth, even on OpenMX

Thanks to Gabor Dozsa and Chris Adeniyi-Jones for their OpenMX results
### Interconnect vs. Performance ratio

<table>
<thead>
<tr>
<th></th>
<th>Peak IN bytes / FLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Gb/s</td>
</tr>
<tr>
<td>Tegra2</td>
<td>0.06</td>
</tr>
<tr>
<td>Tegra3</td>
<td>0.02</td>
</tr>
<tr>
<td>Exynos 5250</td>
<td>0.02</td>
</tr>
<tr>
<td>Intel i7</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- Mobile SoC have low-bandwidth interconnect …
  - 1 GbE or USB 3.0 (6Gb/s)
- … but ratio to performance is similar to high-end
  - 40 Gb/s Infiniband
Limitations of current mobile processors for HPC

- 32-bit memory controller
  - Even if ARM Cortex-A15 offers 40-bit address space
- No ECC protection in memory
  - Limited scalability, errors will appear beyond a certain number of nodes
- No standard server I/O interfaces
  - Do NOT provide native Ethernet or PCI Express
  - Provide USB 3.0 and SATA (required for tablets)
- No network protocol off-load engine
  - TCP/IP, OpenMX, USB protocol stacks run on the CPU
- Thermal package not designed for sustained full-power operation

- All these are implementation decisions, not unsolvable problems
  - Only need a business case to justify the cost of including the new features … such as the HPC and server markets
<table>
<thead>
<tr>
<th>Per-node figure</th>
<th>Server chips</th>
<th>Mobile chips</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intel SandyBridge (E5-2670)</td>
<td>AppliedMicro X-Gene</td>
<td>Calxeda EnergyCore (“Midway”)</td>
<td>TI Keystone II</td>
</tr>
<tr>
<td>#cores</td>
<td>8</td>
<td>16-32</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CPU</td>
<td>Sandy Bridge</td>
<td>Custom ARMv8</td>
<td>Cortex-A15</td>
<td>Cortex-A15</td>
</tr>
<tr>
<td>Technology</td>
<td>32nm</td>
<td>40nm</td>
<td>28nm</td>
<td>28nm</td>
</tr>
<tr>
<td>Clock speed</td>
<td>2.6GHz</td>
<td>3GHz</td>
<td>2GHz</td>
<td>1.9GHz</td>
</tr>
<tr>
<td>Memory size</td>
<td>750GB</td>
<td>?</td>
<td>4GB</td>
<td>4GB</td>
</tr>
<tr>
<td>Memory bandwidth</td>
<td>51.2GB/s</td>
<td>80 GB/s</td>
<td>12.8 GB/s</td>
<td>12.8 GB/s</td>
</tr>
<tr>
<td>ECC in DRAM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>I/O bandwidth</td>
<td>80GB/s</td>
<td>?</td>
<td>4 x 10 Gb/s</td>
<td>10 Gb/s</td>
</tr>
<tr>
<td>I/O interface</td>
<td>PCIe</td>
<td>Integrated</td>
<td>Integrated</td>
<td>Integrated</td>
</tr>
<tr>
<td>Protocol offload</td>
<td>(in the NIC)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Nvidia Tegra4 | Samsung Exynos 5 Octa

- Cortex-A15
- Cortex-A15 + Cortex-A7
- 28nm
- 28nm
- 1.9GHz
- 1.8GHz
- 4GB
- 4GB
- 12.8 GB/s
- 12.8 GB/s
- Yes
- Yes
- Yes
- No
- No
- 6 Gb/s *
- 6 Gb/s *
- USB 3.0
- USB 3.0
- No
- No
Conclusions

• Mobile processors have qualities that make them interesting for HPC
  • FP64 capability
  • Performance increasing rapidly
  • Large market, many providers, competition, low cost
  • Embedded GPU accelerator

• Current limitations due to target market conditions
  • Not real technical challenges

• A whole set of ARM server chips is coming
  • Solving most of the limitations identified

• Get ready for the change, before it happens …