Marrying Many-core Accelerators and InfiniBand for a New Commodity Processor

Konstantin S. Solnushkin* <konstantin@solnushkin.org>

Yuichi Tsujita <yuichi_tsujita@fw.ipsj.or.jp>

*Your speaker today

International Conference on Computational Science (ICCS 2013)
Barcelona, Spain
June 5 - June 7, 2013
About the speaker

• PhD candidate, topic is “Automated Design of Computer Clusters”

• Main findings:
  – Continuity is more important than sporadic improvements
  – Ease of use is more important than energy efficiency (or any other trendy topic)
    • Although “ease of use” is harder to measure than “energy efficiency”
About the speaker

• Main finding of my PhD research, in one sentence:

Economics is important
About the speaker

• Main finding of my PhD research, in one sentence:

**Economics is important**
Why is economics important?

• New devices bring:
  benefits: energy efficiency, etc.
  disadvantages: you need to learn how to use them

• New device (GPGPUs, Intel Xeon Phi) → need to rewrite code (e.g., learn NVIDIA’s CUDA)

• Learning new devices means time, and therefore money

• Vendor lock-in also translates into money (your knowledge of CUDA can become “sunk costs”)
  – And you will need to learn something new when NVIDIA retires CUDA
Economics

- Mass-produced devices rule the world
- We use commodity, off-the-shelf components to build clusters because they are cheap, not because they are the best
- Future exascale computers are going to contain $O(10^5)$ or even $O(10^6)$ individual CPU chips [1]

1. J. Dongarra et al., “The international exascale software project roadmap”
Economics

• $10^6$ CPUs is a game-changer
• It becomes economically viable to build your own CPUs
• CPU design project costs around $100M [1]. This is just $100 per one future CPU
• Costs of processing a 300mm wafer were $5000-$6000 in 2010 [2]. Such a wafer has ~23 good 435 mm$^2$ CPU dies (at 18% yield). That is $260 per one CPU.
• We don’t need any marketing costs
• For comparison: Intel® Xeon® Processor E5-2680 – $1723

1. Wikipedia: “CPU design”
2. John L. Hennessy and David A. Patterson, “Computer Architecture: A Quantitative Approach”
Economics

• Why buy ready-made chips when you can design your own?

• Japan and China are already doing this.

• Besides, money spent inside the country doesn’t leave the economy
  – Government contracts “trickle down” as salaries, then return back as taxes
  – Especially good in view of the financial crisis
Economics

• Japan: “K Computer”. Based on Fujitsu SPARC64 VIIIfx CPUs – government-funded

• China:
  – “Tianhe-1A” computer: 2,048 “FeiTeng-1000” processors for service nodes – government-funded
  – “Loongson” processor: MIPS architecture; public-private partnership

• It’s all about technological independence
Economics

• What we propose is to design:
  – A new open-source CPU
  – Based on existing architecture such as OpenSPARC
  – With multi-core and many-core parts
    • Multi-core part feeds data to many-core part
  – Integrating InfiniBand network controller on the chip
  – Not very expensive
  – Capable of being produced in millions
  – Designed for HPC, *but equally useful for data centres and desktops*
  – As much open-source as possible
Economics

• The idea of public funding is not new...

• Many Silicon Valley companies are the result of generous government funding
  — (Learn more: Steve Blank, Stanford University: “Secret History of Silicon Valley” [1])

• For example, SPARC architecture grew out of the “Berkeley RISC” project, which was funded by DARPA [2]

2. Wikipedia: “Berkeley RISC”
Economics

• General-purpose architectures can be energy-efficient
• For example, “K Computer” ranked 6th on the June 2011 Green500 list
• Besides, we can reuse heat with water cooling for many purposes (including agricultural) [1]

1. Fruits of Computing: Redefining “Green” in HPC Energy Usage (link).
What’s wrong with the current approaches?

- GPGPUs are not suitable for generic data centre workloads
  - Will remain as “computing”, not “data-processing” engines
  - An important milestone in computing history, but...
What’s wrong with the current approaches?

• Intel Xeon Phi:
  – Needs to be plugged in into a host computer
  – Based on the outdated and proprietary X86 architecture
  – No real compatibility: existing software still needs to be recompiled
What’s wrong with the current approaches?

- IBM BlueGene:
  - Makes a bet on *lots* of low-power cores
    - you need to have algorithms with lots of parallelism,
    - or very large datasets for weak scaling
  - Architecture is very good, but very proprietary:
    - Money gone to IBM never return back
What’s good with our proposal?

• Based on open-source OpenSPARC architecture
  – CPUs in “K Computer” are also based on it
• Includes industry-standard, field-proven InfiniBand interconnect with good performance and clear roadmap
• Low risk: we are not starting from scratch
What’s good with our proposal?
(There was so much it didn’t fit on one slide)

• Thanks to being open-source, any aspect of the system can be changed in the future

• Will boost innovation in Europe (or other region that decides to undertake this project)

• Will cause more competition for giant market players
Related work

• “DEEP” project (EU-funded) [1]:
  – Cluster of traditional servers
  – Cluster of Intel Xeon Phi accelerators
  – Two clusters connected via InfiniBand

• Heterogeneous architecture

• Doesn’t cater to data centre workloads

1. DEEP Project, Dynamic exascale entry platform, http://www.deep-project.eu
Related work

• “Mont-Blanc” project (EU-funded) [1]:
  – Based on mass-produced CPUs with ARM architecture
  – Data centre workloads possible
  – Mass-produced means low cost
  – CPUs are like those used in mobile phones: low-power, low-capability

1. Mont-Blanc Project, http://www.montblanc-project.eu/
Related work

• NVIDIA:
  – Project “Denver” [1]
    • General-purpose ARM-based cores + CUDA-programmed GPU cores
  – “Project Echelon” [2]
    • general-purpose system suitable for data-intensive and HPC workloads
    • based on long instruction word (LIW) architecture

1. B. Dally, “Project Denver" processor to usher in new era of computing, http://blogs.nvidia.com/2011/01/project-denver-processor-to-usher-in-new-era-of-computing/
2. S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, D. Glasco, GPUs and the future of parallel computing, Micro, IEEE 31 (5) (2011) 7-17
Related work

• Intel:
  – Project “Runnemede” [1]
  – employs a large number of simple cores called “execution engines”
  – controlled by a smaller number of general-purpose cores called “control engines”
  – borrows ideas from dataflow architectures

1. N. P. Carter et al., “Runnemede: An architecture for ubiquitous high-performance computing, in: High Performance Computer Architecture (HPCA), 2013 IEEE 19th International Symposium on, 2013”
Related work

• Lots of “orthogonal” EU-funded projects that can mutually benefit from our proposal:
  – DAL Project [1]
  – Fit4Green [2]
  – etc.
The chip

• Multi-core and many-core parts

• Both are based on the same architecture
  – Multi-core: several powerful cores useful for everything
  – Many-core: lots of simpler cores targeted at floating-point operations

• Multi-core part feeds data into the many-core part
  – And takes care of collective communications

• InfiniBand controller integrated on the chip to connect to the outside world
The chip

- Three integration scenarios:
  - Three chips on the same board: multi-core, many-core, network chip
  - Multi-core and many-core parts on the same chip; separate network chip
  - All three parts on a single chip, resulting in a SoC
The chip

- Dark silicon [1]:
  - we can place many cores onto the same chip
  - but we cannot use them all at once (because of the power limitation)
  - therefore we place many heterogeneous cores with different functions
  - and use those which are required by a particular workload

M. B. Taylor, “Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse”, in: Proceedings of the 49th Annual Design Automation Conference, ACM, 2012, pp. 1131-1136.
The chip

- What cores do we suggest to put on the chip?
  
  - Graphics processing unit (video adaptor) – for use of the chip for desktop computers
  
  - Encryption units
  
  - FPGA unit for even more programming flexibility
  
  - Do you have proposals? We want to hear from you!
Use cases

• How all this stuff will be used?

• Three possible uses:
  – HPC
  – Server
  – Desktop
Use case: HPC

- General-purpose multi-core processing unit
- Interfaces (network, DRAM controller, ...)
- Many-core processing unit
- Custom cores (encryption, FPGA, ...)
- GPU

- I/O and MPI delegation, feeding data into the many-core unit
- Main floating-point intensive computational work
Use case: servers

- General-purpose multi-core processing unit
- GPU
- Interfaces (network, DRAM controller, ...)
- Many-core processing unit
- Custom cores (encryption, FPGA, ...)

Both units running data centre workloads (data mining, e-mail, anti-virus, web, proxy, ...)

ClusterDesign.org
Use case: desktops

General-purpose multi-core processing unit

Interfaces (network, DRAM controller, ...)

Many-core processing unit

Custom cores (encryption, FPGA, ...)

Legacy workloads with low thread count (text processing, web browsing)

Audio/image processing, video rendering, games
The chip

• We don’t know how the semiconductor industry will advance
  – all forecasts are based on projections and extrapolation

• We cannot decide right now:
  – how many cores we will have,
  – how much cache memory, etc.

• Time will show!
The chip

- Things to appear on the chip:
  - Multi-core part: with a possibility of increased clock rate when many-core part is not engaged
  - Many-core part: not too many cores to guarantee ample bandwidth for each of them
  - Cache memory: use eDRAM technology (see IBM’s success story [1])
    - With an opportunity to configure it as a user-controlled scratch pad
  - Flash storage chip to create a new level of memory hierarchy (first featured in “Gordon” supercomputer)

- 1. R. Kalla, B. Sinharoy, W. J. Starke, M. Floyd, POWER7: IBM’s next-generation server processor, IEEE Micro 30 (2) (2010) 7-15.
Packaging

• On the proposed board, the InfiniBand controller is integrated on the chip
• There exist ready-made InfiniBand “switch” chips with 36 ports
• We can unite 18 boards via a backplane and connect them to the 1st level InfiniBand switch chip
• Other 18 ports of the “switch” chip will go to the outside world
• InfiniBand supports torus and fat-tree topologies out of the box
Cooling

• For desktop computing, we anticipate traditional air cooling
  – the chip will run in “slower” mode, automatically tuning itself to available cooling

• For HPC and server uses, we suggest liquid cooling:
  – Via a hot-plate or heat spreaders attached to the board (like in “SuperMUC”)
  – Immersive cooling (see http://www.allied-control.com)

• With both types of liquid cooling, heat can be easily reused
Software ecosystem

• Operating system:
  – SPARC is already supported under Linux:
  – Linux kernel includes contributions from Afara Websystems
  – Existing Linux distributions for SPARC: Debian, Gentoo

• Compilers and similar tools (profilers, debuggers):
  – Workable, but needs improvement to handle the many-core part

• Numerical libraries and MPI implementations:
  – Needs improvement, but on the other hand we are not starting from scratch
Software ecosystem

• **Scientific applications:**
  - The goal is: recompile them, and they “automagically” work on the new platform
  - Easier said than done

• **Server software:**
  - Things like Apache (web server), Squid (proxy server), SQL server, PHP & Python etc. will hopefully run just fine, making use of available parallelism (including simultaneous multi-threading):
  - Independent processes will run on both the multi-core and many-core parts
Software ecosystem

• **Desktop software:**
  – Mozilla, OpenOffice, GIMP – hopefully will just work too
  – They will run mostly on the multi-core part
  – Things like audio/image/video processing that require lots of computations will offload work to the many-core part

• Let’s be realistic: for desktop users, the system is unusable unless they can “browse funny cat videos on Facebook”
  – We have a long way to go to overthrow the domination of existing proprietary architectures 😊
MPI_Allgather optimisation (Multi-core and many-core hybrid system, with delegation)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation (Multi-core *and* many-core hybrid system, with delegation)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation (Multi-core and many-core hybrid system, with delegation)

Leader process on each node is managing inter-node communications.
 MPI_Allgatter optimisation (Multi-core and many-core hybrid system, with delegation)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation
(Many-core only system with delegation to leader processes)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation
(Many-core only system with delegation to leader processes)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation
(Many-core only system with delegation to leader processes)

Leader process on each node is managing inter-node communications.
MPI_Allggather optimisation
(Many-core only system with delegation to leader processes)

Leader process on each node is managing inter-node communications.
MPI_Allgather optimisation
(Many-core only system with delegation to leader processes)

Leader process on each node is managing inter-node communications.
• Collective communications should take into account hardware affinity

• Responses from remote compute nodes can be cached for later reuse
Conclusions

• Making a non-X86 CPU for HPC is possible
  – Japan and China are already doing this
  – projects are government-funded
• We propose a CPU designed for HPC but suitable for server and desktop markets as well
• Broader market → larger production → lower prices
• Open-source with no vendor lock-in
That’s all, folks!
Time for questions!

Image: “Monty Python's The Meaning of Life” [1, 2]