High Performance Computing on ARM

C. Steinhaus C. Wedding

christian.{wedding, steinhaus}@rwth-aachen.de

February 12, 2015

Overview

Matrix matrix multiplikation

- Break down Matrix into smaller calculations
- Optimize these calculations
- Run them in parallel
- BLIS breaks GEMM down to $(4 \times 4) \cdot (4 \times 4)$
- NEON implements $(4 \times 4) \cdot (4 \times 4)$

Matrix matrix multiplikation as implemented in NEON

$$
\begin{matrix} x0 & x4 & x8 & xC \\ x1 & x5 & x9 & xD \\ x2 & x6 & xA & xE \\ x3 & x7 & xB & xF \end{matrix}
\times
\begin{matrix} y0 & y4 & y8 & yC \\ y1 & y5 & y9 & yD \\ y2 & y6 & yA & yE \\ y3 & y7 & yB & yF \end{matrix}
=
\begin{matrix} x0y0+x4y1+x8y2+xCy3 & x0y4+... \\ x1y0+x5y1+x9y2+xDy3 & x1y4+... \\ x2y0+x6y1+xAy2+xEy3 & x2y4+... \\ x3y0+x7y1+xBy2+xFy3 & x3y4+... \end{matrix}
$$

Table 1: NEON implementation of matrix matrix multiplikation

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0425/ch04s06s05.html

Matrix matrix multiplikation as implemented in NEON

$$\begin{array}{cccc} x0 & x4 & x8 & xC \\ x1 & x5 & x9 & xD \\ x2 & x6 & xA & xE \\ x3 & x7 & xB & xF \end{array} \times \begin{array}{cccc} y0 & y4 & y8 & yC \\ y1 & y5 & y9 & yD \\ y2 & y6 & yA & yE \\ y3 & y7 & yB & yF \end{array} = \begin{array}{cc} x0y0+x4y1+x8y2+xCy3 & x0y4+... \\ x1y0+x5y1+x9y2+xDy3 & x1y4+... \\ x2y0+x6y1+xAy2+xEy3 & x2y4+... \\ x3y0+x7y1+xBy2+xFy3 & x3y4+... \end{array}$$

Table 2: NEON implementation of matrix matrix multiplikation

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0425/ch04s06s05.html

Matrix matrix multiplikation as implemented in NEON

$$
\begin{array}{cccc} x0 & x4 & x8 & xC \\ x1 & x5 & x9 & xD \\ x2 & x6 & xA & xE \\ x3 & x7 & xB & xF \end{array}
\times
\begin{array}{cccc} y0 & y4 & y8 & yC \\ y1 & y5 & y9 & yD \\ y2 & y6 & yA & yE \\ y3 & y7 & yB & yF \end{array}
=
\begin{array}{cc} x0y0+x4y1+x8y2+xCy3 & x0y4+... \\ x1y0+x5y1+x9y2+xDy3 & x1y4+... \\ x2y0+x6y1+xAy2+xEy3 & x2y4+... \\ x3y0+x7y1+xBy2+xFy3 & x3y4+... \end{array}
$$

Table 3: NEON implementation of matrix matrix multiplikation

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0425/ch04s06s05.html

Matrix matrix multiplikation as implemented in NEON

$$\begin{matrix} x0 & x4 & x8 & xC \\ x1 & x5 & x9 & xD \\ x2 & x6 & xA & xE \\ x3 & x7 & xB & xF \end{matrix} \times \begin{matrix} y0 & y4 & y8 & yC \\ y1 & y5 & y9 & yD \\ y2 & y6 & yA & yE \\ y3 & y7 & yB & yF \end{matrix} = \begin{matrix} x0y0+x4y1+x8y2+xCy3 & x0y4+... \\ x1y0+x5y1+x9y2+xDy3 & x1y4+... \\ x2y0+x6y1+xAy2+xEy3 & x2y4+... \\ x3y0+x7y1+xBy2+xFy3 & x3y4+... \end{matrix}$$

Table 4: NEON implementation of matrix matrix multiplikation

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0425/ch04s06s05.html

Matrix matrix multiplikation as implemented in NEON

x0	x4	x8	xC		y0	y4	y8	yC		x0y0+x4y1+x8y2+xCy3	x0y4+...
x1	x5	x9	xD	×	y1	y5	y9	yD	=	x1y0+x5y1+x9y2+xDy3	x1y4+...
x2	x6	xA	xE		y2	y6	yA	yE		x2y0+x6y1+xAy2+xEy3	x2y4+...
x3	x7	xB	xF		y3	y7	yB	yF		x3y0+x7y1+xBy2+xFy3	x3y4+...

Table 5: NEON implementation of matrix matrix multiplikation

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dai0425/ch04s06s05.html

Paper 1

Design and Analysis of a 32-bit Embedded High-Performance Cluster Optimized for Energy and Performance

Michael F. Cloutier, Chad Paradis and Vincent M. Weaver

Model	Processor Family	Cores	Speed
Raspberry Pi Model B+	ARM1176	1	700MHz
Chromebook	ARM Cortex A15	2	1.7GHz
ODROID-xU	ARM Cortex A7/A15	4(big) 4(little)	1.6GHz 1.2GHz
AMD Opteron 6376		16	2.3GHz
Intel Sandybridge-EP		12	2.3GHz

Table 6: Specification of relevant hardware for DLA Paper 1

Performance evaluation
Different ARM boards

Machine	N	Total Energy	MFLOPS per Watt	MFLOPS per $
Raspberry Pi Model B	4000	801.80	53.2	5.06
Raspberry Pi Model B+	4000	494.05	86.4	5.06
Gumstix Overo	4000	2853.39	15.0	0.20
Beagleboard-xm	4000	3109.50	13.7	0.36
Beaglebone Black	4000	1679.52	25.4	1.48
Pandaboard	4000	344.24	12.4	2.69
Trimslice	4000	n/a	n/a	n/a
Cubieboard2	4000	418.37	10.2	5.18
Chromebook	10,000	2630.54	253.0	14.18
ODROID-xU	10,000	1933.52	345.0	14.73

Figure 1: Comparison ARM architecture

- High-performance Linpack (HPL)
- ATLAS as BLAS
- MPI for message-passing
- Scaled problems for stronger processors

Performance evaluation
ARM and x86_64

Machine	N	Total Energy	MFLOPS per Watt	MFLOPS per $
Raspberry Pi Model B	4000	801.80	53.2	5.06
Raspberry Pi Model B+	4000	494.05	86.4	5.06
Gumstix Overo	4000	2853.39	15.0	0.20
Beagleboard-xm	4000	3109.50	13.7	0.36
Beaglebone Black	4000	1679.52	25.4	1.48
Pandaboard	4000	344.24	12.4	2.69
Trimslice	4000	n/a	n/a	n/a
Cubieboard2	4000	418.37	10.2	5.18
Chromebook	10,000	2630.54	253.0	14.18
ODROID-xU	10,000	1933.52	345.0	14.73
2 core Intel Atom S1260	15,000	20,468	112.3	4.25
16 core AMD Opteron 6376	40,000	171,904	247.2	21.60
12 core Intel Sandybridge-EP	40,000	123,321	346.0	21.30

Figure 2: Comparison ARM vs x86_64 processors

- Scaled problems for stronger processors
- Relative data provides objective results
- Stronger ARM processors can compete with x86
- Power per watt comparable
- ODROID expensive because specific

Paper 2

Evaluating Energy Efficient HPC Clusters for Scientific Workloads

Jahanzeb Maqbool, Sangyoon Oh and Geoffrey C. Fox

	ARM SoC	Intel Server
Processor	Samsung Exynos 4412	Intel Xeon x3430
Processor Family	ARM Cortex A9	Intel Nehalem
L1/L2/L3	32K(i) 32K(d) / 1M / None	32K / 256K / 4M
# of cores	4	4
Clock Speed	1.4 GHz	2.40 GHz
Instruction Set	32-bit	64-bit

Table 7: Specification of the compared ARM and Intel processors

Performance evaluation
Paper 2

- R_{max} : maximum amount of GFLOPS
- $\bar{P}(R_{max})$: average powerconsumption

Testbed	Build	$R_{max}(GFLOPS)$	$\bar{P}(R_{max})$	PPW(MFLOPS/watt)
Weiser	ARM Cortex-A9	24.86	79.13	321.70
Intel x86	Xeon x3430	26.91	138.72	198.64

Table 8: Energy Efficiency of Intel x86 server and Weiser cluster running HPL benchmark

Conclusion Dense Linar Algebra

- ARM can compare to x86 in Power/Watt
- Nonstandard hardware results in high acquisation costs
- Small cache size limits ARM when computing larger problems
- ARM is currently in the ascendent

MapReduce

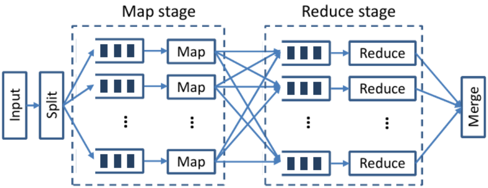

Figure 3: Mapreduce model

- Programming model for processing large datasets on clusters
- Composition of map and reduce procedures
- Used to compute word count, string match, histogram and more

Paper 1

Comparing the Performance and Power Usage of GPU and ARM Clusters for Map-Reduce

Vivian Delplace and Pierre Manneback

Hardware	Cores	CPU clock	Maximum Power
Nvidia M2090	512	1.3Ghz	225W
Viridis ARM cluster(Cortex A9)	192	1.4GHz	300W

Table 9: Specification of the compared ARM and GPU hardware

	WC	SM
Mars	172	172
Disco	32	31

Table 10: Lines of code on GPU (Mars) and ARM (Disco)

Evaluation Paper 1

Word Count (map+reduce)

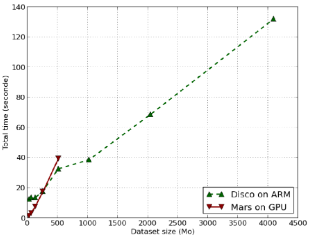

Figure 4: Total time

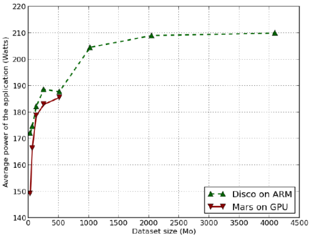

Figure 5: Power average

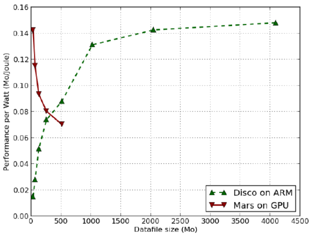

Figure 6: Performance/Watt

Evaluation Paper 1

Stringmatch (only map)

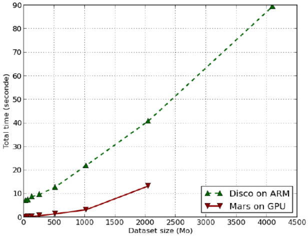

Figure 7: Total time

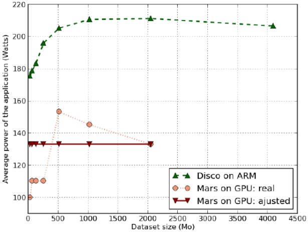

Figure 8: Power average

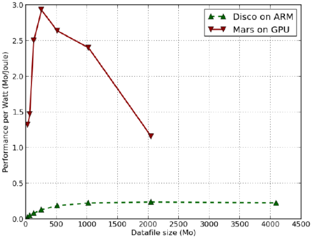

Figure 9: Performance/Watt

Application	input size	perf/W ARM cluster	perf/W GPU	ratio GPU/ARM cluster
WC	512 MB	0.088008	0.070254	0.80
SM	2048 MB	0.238806	1.158083	4.80

Table 11: Performance per watt per application for the largest input

Mars (GPU)	Disco (ARM)
C++/CUDA	Erlang and Python
global memory directly accessible	local disks
small inputs	large inputs
almost at full potential	already good still improvable

Table 12: Direct comparison

Paper2

Performance Evaluation of Embedded Processor in MapReduce Cloud Computing Applications

Christoforos Kachris, Georgios Sirakoulis and Dimitrios Soudris

	HP-GPP	LP-GPP	EP
Processor	i7-2600	U5400	OMAP4430
# of Cores	4	2	2
Cores	Intel i7	Intel Pentium	ARM Cortex A9
Frequency	3.4 GHz	1.2 GHz	1 GHz
L1 Cache	64 KB (I), 64 KB (D)	64 KB (I), 64 KB (D)	32 KB (I), 32 KB (D)
L2 Cache	256 KB per core	256 KB per core	1 MB (shared)
L3 Cache	8 MB	3 MB	-
Instruction Set	64-bits	64-bits	32-bits

Table 13: Processor architecture characteristics

Evaluation Paper 2

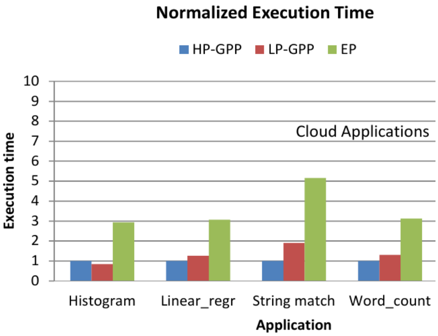

Figure 10: Normalized execution time

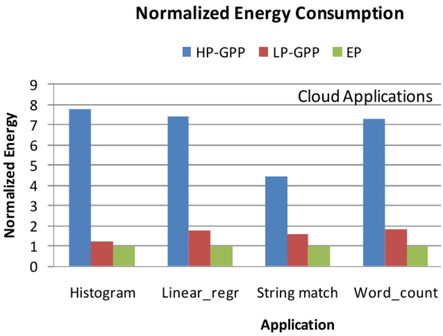

Figure 11: Normalized energy comsumption

Conclusion

- ARM consumes 7.8x less energy
- GPP requires more energy due to:
  - CISC
  - advanced branch prediction scheme
  - larger caches
- Tradeoff between executiontime and energy consumption

Paper

Time-to-Solution and Energy-to-Solution: A Comparison Between ARM and Xeon

Edson Padoin, Daniel de Oliveira, Pedro Velho, Philippe Navaux

Processor	Nehalem	Nehalem	ARM Cortex A9
Processor Model	Xeon E5530	Xeon X7550	OMAP4430
# of Processors	2	4	1
Cores/Processors	4	8	2
Threads/Core	2	2	2
Frequency	2.40 GHz	2.00 GHz	1.00 GHz
TDP (W)	80	130	0.25

Table 14: Specification of the compared hardware

Evaluation

NAS Parallel Benchmark (Fast Fourier Transform)

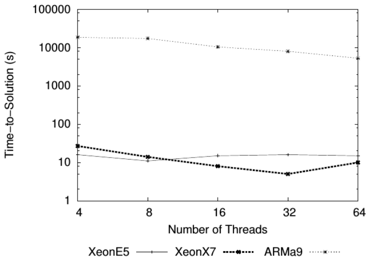

Figure 12: Time-to-Solution as a function of the # threads

Benchmark	XeonE5	XeonX7	ARMa9	$\frac{ARMa9}{XeonE5}$	$\frac{ARMa9}{XeonX7}$
FT	14.6	12.8	11899.6	815.0	929.7

Table 15: Average time-to-solution (in seconds) and ratio of ARMa9 to XeonE5/XeonX7

Evaluation

NAS Parallel Benchmark (Fast Fourier Transform)

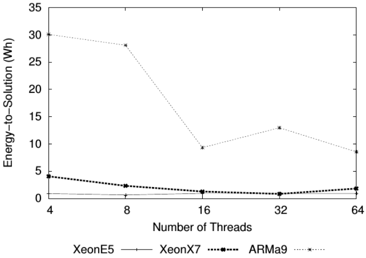

Figure 13: Energy-to-Solution as a function of the # threads

Benchmark	XeonE5	XeonX7	ARMa9	$\frac{XeonE5}{ARMa9}$	$\frac{XeonX7}{ARMa9}$
FT	0.875	2.070	17.800	20.35	8.60

Table 16: Average energy-to-solution (in WH) and ratio with Xeon as reference

Conclusion
as given in the paper

- Xeon has better hardwaresupport for floating point operations
- Xeon outperformes ARM in every NAS Parallel Benchmark (Fast Fourier Transform)
- ARM for HPC still questionable
  - Consumes less energy
  - Requires much more execution time

Paper

Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster

Dominik Göddeke, Dimitri Komatitsch, Markus Geveler, Dirk Ribbrock, Nikola Rajovic, Nikola Puzovic, Alex Ramirez

	ARM	x86
Processor	Cortex A9	Intel Xeon X5550
Clockspeed	2x 1.0GHz	2x 4x2.66GHz
L1 Cache	i/d 32KB	i/d 64KB
L2 Cache	2MB	2MB
L3 Cache		8MB
Memory	896 MB DDR2 (low power)	16 GB DDR3
	does not support NEON the integrated GPUs are not used	no hyperthreading

Table 17: Specifications of the processors

x86 cluster configurations

Tibidabo		Configuration 1		Configuration 2		Configuration 3		Configuration 4	
Cores	Nodes	Nodes	Cores/node	Nodes	Cores/node	Nodes	Cores/node	Nodes	Cores/node
6	3	1	6	1	8	1	8	1	8
12	6	2	6	2	8	1	8	1	8
24	12	4	6	4	8	1	8	2	8
48	24	8	6	8	8	2	8	4	8
96	48	16	6	16	8	3	8	6	8
192	96	32	6	32	8	6	8	12	8
4	2	1	4			1	8	1	8
8	4	1	8			1	8	1	8
12	6	2	6			1	8	1	8
24	12	4	6			1	8	2	6
48	24	8	6			2	8	3	8
96	48	16	6			3	8	6	8
192	96	32	6			6	8	12	8

Figure 14: Details of the mapping onto various x86-Cluster Nodes for FEAST(top) and SPECFEM3D_GLOBE(bottom)

- Config 1: same load per core as on the ARM cluster
- Config 2: use all 8 cores per node (not possible with SPECFEM3D_GLOBE)
- Config 3: use as few nodes as possible
- Config 4: use twice the amount of nodes as in config 3

Application: FEAST(Finite-Element Analysis and Solution Tools)

SBBLAS and MPI

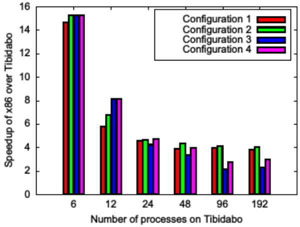

Figure 15: Time to solution

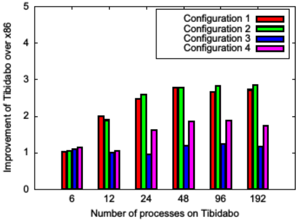

Figure 16: Energy to solution

- FEAST is a Finite Element based solver toolkit for the simulation of PDE problems
- FEAST is rather memory-bound
- ARM Cluster is more energy efficient for all problem sizes and configs

Application: SPECFEM3D_GLOBE
MPI

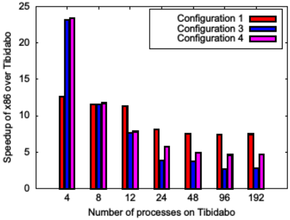

Figure 17: Time to solution

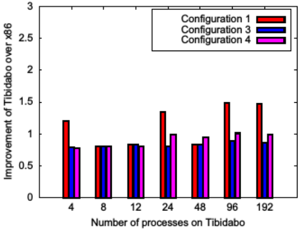

Figure 18: Energy to solution

- SPECFEM3D_GLOBE simulates global and regional seismic wave propagation
- SPECFEM3D_GLOBE is rather compute-bound
- Big difference between peak floating point performance
  $\rightarrow$ no gain in energy efficiency

Conclusion Structured Grids

- Paper was not about Structured Grids in particular but about PDEs
- Again: tradeoff between energy and speed
- Moderate slowdowns but substantial reductions of energy to solutions are possible
- "How much slower can the simulation afford be for certain energy savings"

Conclusion

- Green computing
- Exascale computing achievable
- Exascale computing
  - 20 Megawatt Exascale system by 2018-2019
  - 50 GFLOPS per watt
- 64-Bit
- Cache size

References

- **Design and analysis of a 32-bit embedded high-performance cluster optimized for energy and performance**
  Michael F. Cloutier, Chad Paradis and Vincent M. Weaver
- **Evaluating Energy Efficient HPC Clusters for Scientific Workloads**
  Jahanzeb Maqbool, Sangyoon Oh and Geoffrey C. Fox
- **Comparing the Performance and Power Usage of GPU and ARM Clusters for Map-Reduce**
  Vivian Delplace and Pierre Manneback
- **Performance Evaluation of Embedded Processor in MapReduce Cloud Computing Applications**
  Christoforos Kachris, Georgios Sirakoulis and Dimitrios Soudris
- **Time-to-solution and energy-to-solution: a comparison between ARM and Xeon**
  Edson Padoin, Daniel de Oliveira, Pedro Velho, Philippe Navaux
- **Energy efficiency vs. performance of the numerical solution of PDEs: an application study on a low-power ARM-based cluster**
  Dominik Göddeke, Dimitri Komatitsch, Markus Geveler, Dirk Ribbrock, Nikola Rajovic, Nikola Puzovic, Alex Ramirez

Workbalance

- no individual work
- from papersearch to presentation teamwork