Massive affordable computing using ARM processors in high energy physics

J W Smith, A Hamilton

Department of Physics
University of Cape Town, Rondebosch, South Africa

E-mail: `joshua.wyatt.smith.cern.ch`

Abstract. High Performance Computing is relevant in many applications around the world, particularly high energy physics. Experiments such as ATLAS, CMS, ALICE and LHCb generate huge amounts of data which need to be stored and analyzed at server farms located on site at CERN and around the world. Apart from the initial cost of setting up an effective server farm the cost of power consumption and cooling are significant. The proposed solution to reduce costs without losing performance is to make use of ARM® processors found in nearly all smartphones and tablet computers. Their low power consumption, low cost and respectable processing speed makes them an interesting choice for future large scale parallel data processing centers. Benchmarks on the Cortex™-A series of ARM® processors including the HPL and PMBW suites will be presented as well as preliminary results from the PROOF benchmark in the context of high energy physics will be analyzed.

1. Introduction

High energy physics (HEP) at the Large Hadron Collider (LHC) [1] creates an enormous amount of data that need to be stored for later analysis. Dedicated server farms have been built at CERN and Hungary (Tier 0) as well as around the world (Tier 1's and Tier 2's) and are connected through The Worldwide LHC Computing Grid (WLCG). The initial setup costs of these server farms can be immense and the costs to maintain such a server farm can be even larger. Cooling and the power required to maintain such servers at their peak performance make server farms an expensive venture. In early 2013, CERN's Tier 0 had a power capacity of 3.5MW.

The proposed solution is to make use of ARM® processors from here on referred to as ARM. These are found in smartphones and tablet computers where the combination of low power consumption and high performance is the top priority. Significant savings might be achieved if ARM processors are able to cope with the huge amount of data processed. This letter serves to explore the capabilities of ARM processors through parallel processing benchmarks.

2. ARM Processors

The ARM processor was designed to perform only a few instructions at once. This reduces the need for hardware such as transistors and thus minimizes power consumption. This letter addresses three system on chip (SoC) setups in the Cortex™-A range, namely the A7 MPCore™, A9 MPCore™ and A15 MPCore™ [2, 3, 4], from here on referred to as the A7, A9 and A15 respectively. Table 1 summarizes the different processors used. An obvious advantage to the A9 is the number of cores, however this is irrelevant because the number of cores is an intrinsic property of the SoC. The hardware *Floating-Point Unit* (FPU) generates results for the speed at which multiplication and addition operations are carried out. v3/4 refers to the respective FPU version. A traditional Intel® computer is also used (Hep405). This serves to provide some reference to the reader. Power measurements for the A7, A9 and A15 were taken using a Fluke 289 Digital Multimeter. Measurements on Hep405 were taken using the Intel® Power Gadget. However, if the power usage characteristics and temperatures vary from those used in Intel's calibration then there will be errors between estimated and actual power usage. For this reason, when referring to Hep405's results the reader must take care in remembering that these values serve as more of an estimate.

Table 1. The different setups with key features.

Setup	Processor	Cores	RAM	Cache	FPU	OS
Cubietruck	AllWinner A20, 1.2GHz	A7 dual core	2GiB DDR3	512 KiB L2	VFPv4	Archlinux, hard float
Wandboard-Quad	Freescale i.MX6 Quad, 996MHz	A9 quad core	2GiB DDR3	32KiB L1, 1 MiB L2	VFPv3	Archlinux, hard float
ArndaleBoard-K	Samsung Exynos 5250, 1.7GHz	A15 dual core	2GiB DDR3	32 KiB L1, 1MiB L2	VFPv4	Fedora 19, hard float
Hep405	Intel® Core i7-2600, 3.4GHz	quad core	16GiB DDR3	256KiB L1, 1MiB L2, 8MiB L3	-	Scientific Linux 6

3. Benchmarks

3.1. HPL

The High-Performance LINPACK (HPL) benchmark is the parallel version of the LINPACK benchmark [5] which is synthetic and CPU intensive. *Flops* of a system are calculated by splitting the large matrix into blocks that are then solved on different cores. Figure 1 shows the relationship between matrix size, block size, *Gflops*, power consumption and *Gflops*/watt for the HPL benchmark. The grey area is an "envelope function" which splits the data into blocks along the time axis and then finds the average power value for that window. It gives a representation of the average power consumption which is indicated by the blue line. A trend we see for the A7 and A9 is that as the size of the matrix block that is passed to the separate cores increases, the average power consumption decreases. This is because slower computation means that less work in the form of communication has to happen between the processors. The average power consumption remains fairly constant with the A15 because of the faster clock speed of 1.7GHz. The *Gflops*/watt is calculated by taking the average power consumption for the time window where there is a HPL measurement. We can see that the A9 is the most efficient provided the block size is relatively large. The A7 does not have a fast enough processor and the A15 uses to much power for the output it delivers. The A9's *Gflops*/watt is comparable to the output of Hep405.

3.2. PMBW

The Parallel Memory Bandwidth Benchmark (PMBW) [6] is a suite that measures bandwidth capabilities of a multi-core computer. This is an important test because more cores result in the floating point performance increasing in a linear fashion. However, if the memory bandwidth

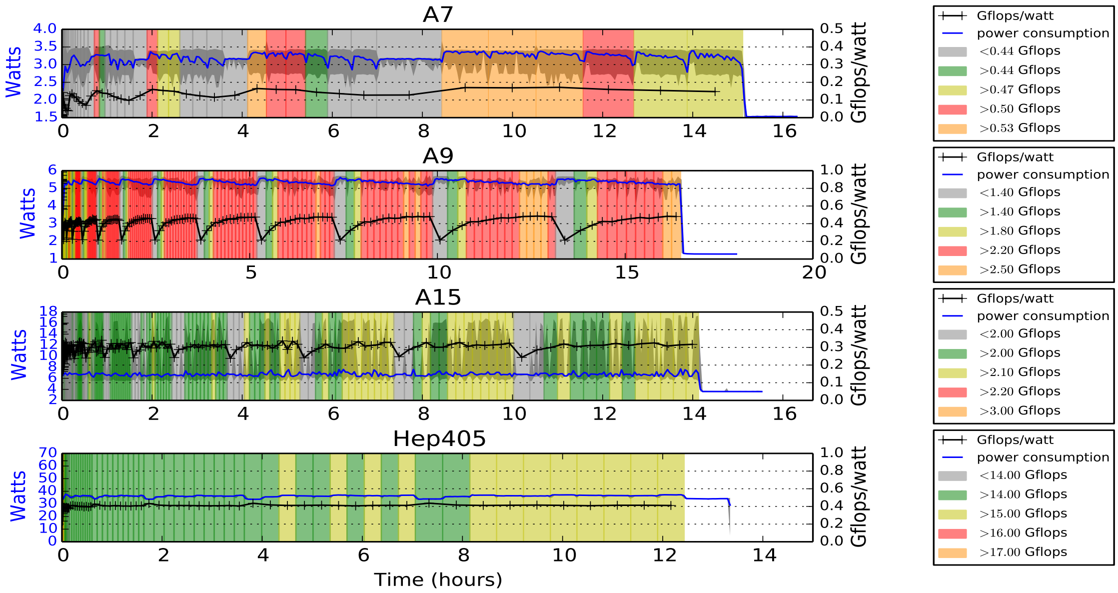

Figure 1. The HPL benchmark applied to the 3 different ARM processors and 1 Intel processor.

is not capable of processing the data fast enough those processors will stall. Unlike floating point units the memory bandwidth does not scale with the number of cores running in parallel. The code was developed in assembler language which means compile flags for the SoC become unimportant, thus there are no optimizations. The code uses two general synthetic access patterns, namely sequential scanning and pure random access. The benchmark outputs an enormous amount of data and so only the results for multiple threads, read/write scan operations with each clock-cycle transferring 32-bits (ARM) and 256 bits (HEP405) are shown in Table 2. It is accurate to say that none of these ARM processors perform very well when it comes to memory bandwidth.

Table 2. Summary of the important results for the HPL and PMBW benchmarks.

	A7	A9	A15	Intel® i7
HPL: Average *Gflops*/W	0.109	0.408	0.323	0.421
PMBW: Max read (GiB/s)	9.505	24.611	21.472	298.264
PMBW: Max write (GiB/s)	9.035	21.428	21.546	163.123

3.3. Dedicated PROOF Cluster

The Parallel ROOT Facility, PROOF [7], is an extension to the well known framework, ROOT [8]. PROOF strives to introduce massive parallelism by splitting up files among workers. The dedicated PROOF cluster at the University of the Witwatersrand (Wits) in Johannesburg, South Africa uses 7 A9 boards sharing a NFS directory hosted by a traditional server. The PROOF benchmark suite [9] was used to determine performance. It is a further add-on designed to test the topology and scalability of clusters by running cycle- (CPU) and data-driven (I/O) processes. Results for vfpv3-d16, neon and default are shown. These correspond to compile flags used in the compilation of ROOT that are specific to the A9.

3.3.1. CPU benchmark The default TSelHist selector was used with the output shown in Figure 2. The unit of measurement is the number of events per unit time.We can see that the cluster scales linearly for CPU-intensive tasks. This is to be expected as the only significant constraint is the overhead produced when the master has to pool the results and create the histograms.

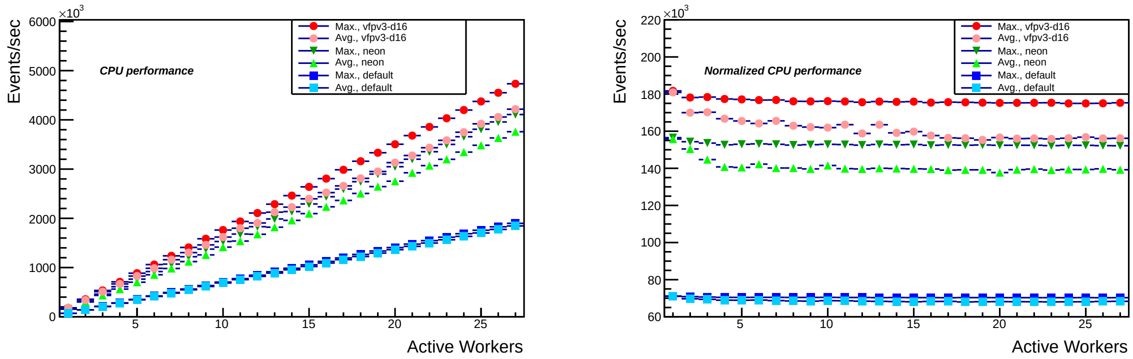

Figure 2. PROOF CPU benchmark for ARM cluster.

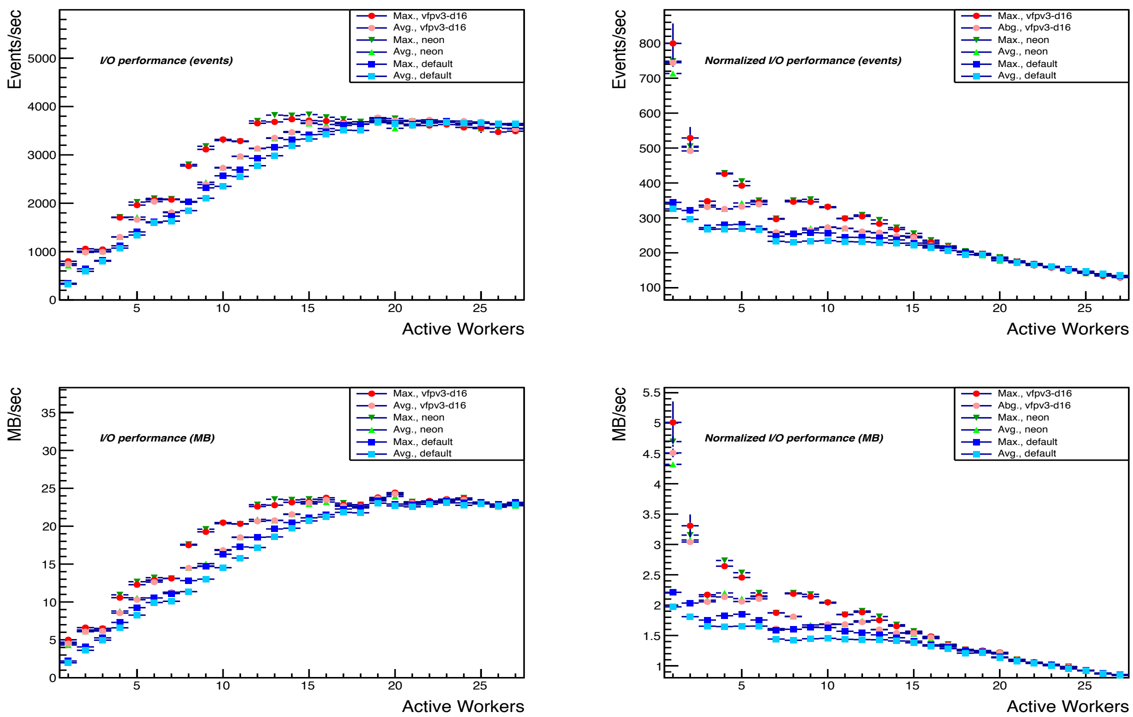

Figure 3. PROOF I/O benchmark for ARM cluster.

3.3.2. I/O benchmark The default TSelEvent selector was used with the output shown in Figure 3. This benchmark shows the bottleneck that the NFS server creates. Scalability saturates at around 12-13 workers which equates to reading events at a total of around 23 MB/sec. Its clear that a shared NFS directory reduces a lot of the advantage that one might gain from a PROOF cluster.

4. Conclusions

ARM processors use significantly less power than traditional processors. However, they are also significantly slower and so parallel computing for the right type of problem must be exploited. HEP data analysis provides tasks that are easily parallelizable and with the correct frameworks in place can be successful. PROOF provides one such framework, however communication bottlenecks have to be overcome. The performance per watt for the A7 and A15 shows poor results when compared to the A9 and Intel machines. The A9's advantage over the other ARM processors lies in the quad core SoC. This processor is still slow compared to traditional computers due in part to the age of the technology. The newer A15 quad core processors should show a dramatic improvement in overall performance and specifically performance per watt. Furthermore, the newly developed 64 bit architecture along with their respective SoCs will have a large impact on the CPU performance and memory bandwidth of these ARM processors.

Acknowledgments

The authors would like to thank the National Research Foundation of South Africa and the University of Cape Town for funding as well as the Massive Affordable Computing (MAC) team at the University of the Witwatersrand for building and providing access to the A9 ARM cluster.

References

[1] 2008 *Journal of Instrumentation* vol 3 (IOP and SISSA)
[2] ARM Ltd 2013 *Cortex^{TM}-A7 MPCore^{TM} Technical Reference Manual* r0p5 ed
[3] ARM Ltd 2012 *Cortex^{TM}-A9 MPCore^{TM} Technical Reference Manual* r4p1 ed
[4] ARM Ltd 2013 *Cortex^{TM}-A15 MPCore^{TM} Processor Technical Reference Manual* r4p0 ed
[5] Dongarra J Linpack benchmark Computer Science Technical Report CS - 89 – 85 University of Tennessee
[6] Bingmann T 2013 pmbw - Parallel Memory Bandwidth Benchmark / Measurement website
[7] Ballintijn M, Brun R, Rademakers F and Roland G 2003 *CHEP03 Intl. Conf.*
[8] Brun R, Rademakers F An object orientated analysis framework URL `http://root.cern.ch`
[9] Ryu S, G Ganis 2012 *J. Phys.Conf. Ser. 368 012020 IOPscience*