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
Electrical and Computer Engineering
University of Maine
{michael.f.cloutier,chad.paradis,vincent.weaver}@maine.edu

***Abstract*—A growing number of supercomputers are being built using processors with low-power embedded ancestry, rather than traditional high-performance cores. In order to evaluate this approach we investigate the energy and performance tradeoffs found with ten different 32-bit ARM development boards while running the HPL Linpack and STREAM benchmarks.**

Based on these results (and other practical concerns) we chose the Raspberry Pi as a basis for a power-aware embedded cluster computing testbed. Each node of the cluster is instrumented with power measurement circuitry so that detailed cluster-wide power measurements can be obtained, enabling power / performance co-design experiments.

While our cluster lags recent x86 machines in performance, the power, visualization, and thermal features make it an excellent low-cost platform for education and experimentation.

I. Introduction

Embedded systems and supercomputers are at opposite ends of the computing spectrum, yet they share common design constraints. As the number of cores in large computers increases, the per-core power usage and cost becomes increasingly important. Luckily there is a class of processors that have already have been optimized for power and cost: those found in embedded systems. The use of embedded processors in supercomputers is not new; the various BlueGene [1], [2], [3] machines use embedded-derived PowerPC chips. There is an ongoing push to continue this trend by taking ARM processors, such as those found in cellphones, and using them in supercomputing applications.

The uptake of ARM processors in supercomputers has started slowly, as vendors were waiting for the release of 64-bit processors with HPC friendly features (such as high-speed interconnects and fast memory hierarchies).

In the meantime it is possible to take readily-available commodity 32-bit ARM boards and use them to build computing clusters. We look at the performance and power characteristics of ten different 32-bit ARM development boards. Our goal is to find a low-cost, low-power, yet high-performance board for use in constructing an educational compute cluster.

After weighing the various tradeoffs, we chose the Raspberry Pi as the basis for our cluster. We built a prototype 32-node Raspberry Pi cluster with per-node energy measurement capabilities, and compare the power and performance tradeoffs with various x86 machines. Our cluster is flexible and can be easily expanded to 64 (or more) nodes.

II. Board Comparison

We measure the power and performance tradeoffs of ten different commodity 32-bit ARM boards, as listed in Table I. The boards, all running Linux, span a wide variety of speeds, cost, and processor types.

A. Experimental Setup

During the experiments all machines are placed in physical states that best simulate their role as a compute node in a cluster. No extraneous devices (keyboards, mice, monitors, external drives) are attached during testing; the only connections are the power supplies and network cables (with the exception of the Chromebook, which has a wireless network connection and a laptop screen).

1) Benchmarking Programs: Choosing a benchmark that properly characterizes a HPC system is difficult. We use two benchmarks commonly used in the HPC area: Linpack and STREAM.

High-performance Linpack (HPL) [4] is a portable version of the Linpack linear algebra benchmark for distributed-memory computers. It is commonly used to measure performance of supercomputers worldwide, including the twice-a-year Top500 Supercomputer list [5]. The program tests the performance of a machine by solving complex linear systems through use of Basic Linear Algebra Subprograms (BLAS) and the Message-Passing Interface (MPI).

For our experiment, mpich2 [6] was installed on each machine to provide a message passing interface (MPI) and the Automatically Tuned Linear Algebra Software (ATLAS) [7] library was installed on each machine to serve as the BLAS.

The second benchmark we use is STREAM [8] which tests a machine's memory performance. STREAM performs operations such as copying bytes in memory, adding values together, and scaling values by another number. The program completes these operations and reports the time it took as well as the speed of the operations.

2) Power Measurement: The power consumed by each machine was measured and logged using a WattsUp Pro [9] power meter. The meter was configured to log the power at its maximum sampling speed of once per second.

978-1-4799-7564-8/14 $31.00 © 2014 IEEE
DOI 10.1109/Co-HPC.2014.7

TABLE I. THE TEN 32-BIT ARM BOARDS EXAMINED IN THIS WORK.

Type	Processor Family	Processor Type	Process	Cores	Speed	CPU Design	BrPred	Network	Cost
Raspberry Pi Model B	ARM1176	Broadcom 2835	40nm	1	700MHz	InOrder 1-issue	YES	10/100 USB	$35
Raspberry Pi Model B+	ARM1176	Broadcom 2835	40nm	1	700MHz	InOrder 1-issue	YES	10/100 USB	$35
Gumstix Overo	ARM Cortex A8	TI OMAP3530	65nm	1	600MHz	InOrder 2-issue	YES	10/100	$199
Beagleboard-xm	ARM Cortex A8	TI DM3730	45nm	1	1GHz	InOrder 2-issue	YES	10/100	$149
Beaglebone Black	ARM Cortex A8	TI AM3358/9	45nm	1	1GHz	InOrder 2-issue	YES	10/100	$45
Pandaboard ES	ARM Cortex A9	TI OMAP4460	45nm	2	1.2GHz	OutOfOrder	YES	10/100	$199
Trimslice	ARM Cortex A9	NVIDIA Tegra2	40nm	2	1GHz	OutOfOrder	YES	10/100/1000	$99
Cubieboard2	ARM Cortex A7	AllWinner A20	40nm	2	912MHz	InOrder Partl-2-Issue	YES	10/100	$60
Chromebook	ARM Cortex A15	Exynos 5 Dual	32nm	2	1.7GHz	OutOfOrder	YES	Wireless	$184
ODROID-xU	ARM Cortex A7/A15	Exynos 5 Octa	28nm	4 (big) 4 (LITTLE)	1.6GHz 1.2GHz	OutOfOrder InOrder	YES	10/100	$169

TABLE II. FLOATING POINT AND GPU CONFIGURATIONS OF THE BOARDS.

Type	Processor Type	FP Support	NEON	GPU	DSP	Offload Engine
Raspberry Pi Model B	Broadcom 2835	VFPv2	no	VideoCore IV (24 GFLOPS)	yes	n/a
Raspberry Pi Model B+	Broadcom 2835	VFPv2	no	VideoCore IV (24 GFLOPS)	yes	n/a
Gumstix Overo	TI OMAP3530	VFPv3 (lite)	YES	PowerVR SGX530 (1.6 GFLOPS)	n/a	n/a
Beagleboard-xm	TI DM3730	VFPv3 (lite)	YES	PowerVR SGX530 (1.6 GFLOPS)	TMS320C64x+	n/a
Beaglebone Black	TI AM3358/9	VFPv3 (lite)	YES	PowerVR SGX530 (1.6 GFLOPS)	n/a	n/a
Pandaboard ES	TI OMAP4460	VFPv3	YES	PowerVR SGX540 (3.2 GFLOPS)	IVA3 HW Accel	2 x Cortex-M3 Codec
Trimslice	NVIDIA Tegra2	VFPv3, VFPv3d16	no	8-core GeForce ULP GPU	n/a	n/a
Cubieboard2	AllWinner A20	VFPv4	YES	Mali-400MP2 (10 GFLOPS)	n/a	n/a
Chromebook	Exynos 5250 Dual	VFPv4	YES	Mali-T604MP4 (68 GFLOPS)	Image Processor	n/a
ODROID-xU	Exynos 5410 Octa	VFPv4	YES	PowerVR SGX544MP3 (21 GFLOPS)	n/a	n/a

TABLE III. MEMORY HIERARCHY DETAILS FOR THE BOARDS.

Type	Processor Type	RAM	L1-I	L1-D	L2	Prefetch
Raspberry Pi Model B	Broadcom 2835	512MB	16k,4-way,VIPT,32B	16k,4-way,VIPT,32B	128k*	no
Raspberry Pi Model B+	Broadcom 2835	512MB	16k,4-way,VIPT,32B	16k,4-way,VIPT,32B	128k*	no
Gumstix Overo	TI OMAP3530	256MB DDR	16k,4-way,VIPT	16k,4-way,VIPT	256k	no
Beagleboard-xm	TI DM3730	512MB DDR2	32k,4-way,64B	32k,4-way,64B	256k,64B	no
Beaglebone Black	TI AM3358/9	512MB DDR3	32k,4-way,64B	42k,4-way,64B	256k,64B	no
Pandaboard ES	TI OMAP4460	1GB LPDDR2 Dual	32k,4-way,VIPT,32B	32k,4-way,VIPT,32B	1MB (external)	yes
Trimslice	NVIDIA Tegra2	1GB LPDDR2 Single	32k	32k	1MB	yes
Cubieboard2	AllWinner A20	1GB DDR3	32k	32k	256k shared	yes
Chromebook	Exynos 5 Dual	2GB LPDDR3/2 Dual Channel	32k	32k	1M	yes
ODROID-xU	Exynos 5 Octa	2GB LPDDR3 Dual	32k	32k	512k/2MB	yes

* By default the L2 on the Pi belongs to the GPU, but Raspbian reconfigures it for CPU use.

B. HPL FLOPS Results

Table IV summarizes the floating point operations per second (FLOPS) results when running HPL. The results for all of the boards were gathered with N=4000 for consistency; these are not the peak observed FLOPS values. The two machines using the more advanced Cortex-A15 CPUs (Chromebook and Odroid) were instead benchmarked at N=10000 due to their larger memory and faster processors; with the smaller problem size they finished too quickly to get power measurements.

The FLOPS value is unexpectedly low on the Cortex-A8 machines; the much less advanced ARM-1176 Raspberry-Pi obtains better results. This is most likely due to the "VFP-lite" floating point unit found in the Cortex-A8 which takes 10 cycles per operation rather than just one. It may be possible to improve these results by changing the gcc compiler options; by default strict IEEE-FP correctness is chosen over raw speed.

The Cortex-A9 Pandaboard shows a nice boost in performance, although much of that is due to the availability of two cores. We do not have numbers for the Trimslice; this machine lacks NEON support (which is optional on the Cortex-A9) which meant the default pre-compiled version of ATLAS used on the other machines would not run. All attempts to hand-compile a custom version of ATLAS without NEON support failed during the build process.

The Cortex-A15 machines have an even greater boost in FLOPS, although they still are over an order of magnitude less than the two example x86 server boards (but around the same as the x86 Atom machine).

C. HPL FLOPS per Watt Results

Table IV also shows the FLOPS per average power results. This is shown graphically in Figure 1, where an ideal system

TABLE IV. FLOPS SUMMARY. THE TRIMSLICE RESULTS ARE MISSING DUE TO PROBLEMS COMPILING A NEON-LESS VERSION OF ATLAS.

Machine	N	Operations	Time	MFLOPS	Idle Power	Avg Load Power	Total Energy	MFLOPS per Watt	MFLOPS per $
Raspberry Pi Model B	4000	42.7B	240.78	177	2.9	3.33	801.80	53.2	5.06
Raspberry Pi Model B+	4000	42.7B	241.0	177	1.8	2.05	494.05	86.4	5.06
Gumstix Overo	4000	42.7B	1060.74	40	2.0	2.69	2853.39	15.0	0.20
Beagleboard-xm	4000	42.7B	799.28	53	3.6	3.89	3109.50	13.7	0.36
Beaglebone Black	4000	42.7B	640.36	67	1.9	2.62	1679.52	25.4	1.48
Pandaboard	4000	42.7B	79.87	535	2.8	4.31	344.24	12.4	2.69
Trimslice	4000	42.7B	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Cubieboard2	4000	42.7B	137.17	311	2.2	3.05	418.37	10.2	5.18
Chromebook	10,000	667B	255.64	2610	5.8	10.29	2630.54	253.0	14.18
ODROID-xU	10,000	667B	267.43	2490	2.7	7.23	1933.52	345.0	14.73
2 core Intel Atom S1260	15,000	2.2T	907.4	2550	18.9	22.7	20,468	112.3	4.25
16 core AMD Opteron 6376	40,000	42.7T	500.41	85,300	162.1	343.53	171,904	247.2	21.60
12 core Intel Sandybridge-EP	40,000	42.7T	501.99	85,000	93.0	245.49	123,321	346.0	21.30

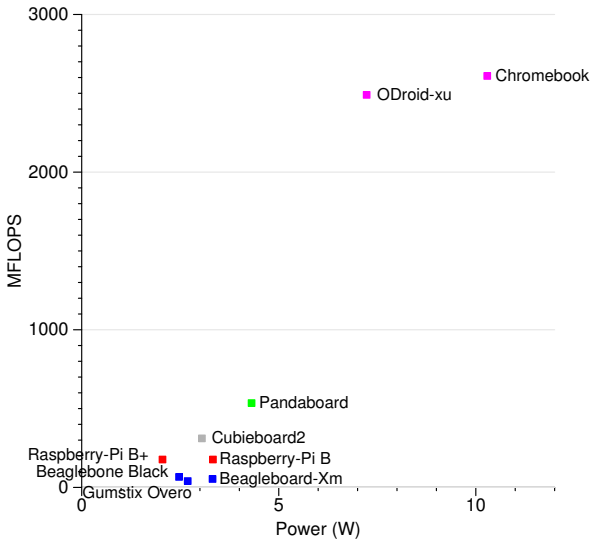

Fig. 1. MFLOPS compared to average power. Upper left is the best.

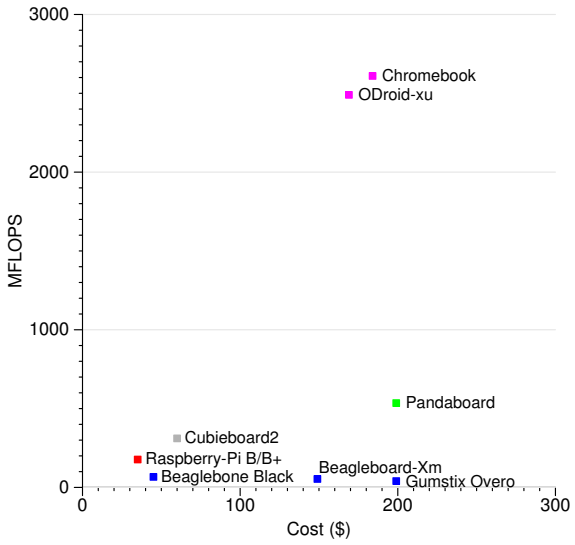

Fig. 2. MFLOPS compared to Cost. Upper left is the best.

optimizing both metrics would have points in the upper left.

In this metric the Cortex-A15 machines are best by a large margin. It is interesting to note that these machines have similar values to those found on high-end x86 servers. For the remaining boards, the recently released Raspberry-Pi B+ (which has the same core as the model B but with more efficient power circuitry) is the clear winner.

D. HPL FLOPS per Cost Results

Table IV also shows the FLOPS per dollar cost (purchase price) of the system (higher is better); this is also shown in Figure 2 where an ideal system optimizing both would have points in the upper left.

Again the Cortex-A15 systems are the clear winners, with the Cortex-A7 Cubieboard and the Raspberry Pi systems making strong showings as well. The x86 servers still win on this metric despite their much higher cost.

E. STREAM Results

We ran Version 5.10 of the STREAM benchmark on all of the machines with the default array size of 10 million (except for the Gumstix Overo, which only has 256MB of RAM so a problem size of 9 million was used). Figure 3 shows a graph of the performance of each benchmark. The more advanced Cortex-A15 chips have much better memory performance than the other boards, most likely due to the use of dual-channel DDR3 memory.

The number of DRAM channels does matter, as the Trimslice vs Pandaboard comparison shows. The Trimslice only has a single channel to memory, while the Pandaboard has two, and the memory performance is correspondingly better.

F. Summary

Results of both the HPL and STREAM benchmarks show the two Cortex-A15 machines to be the clear winners for 32-

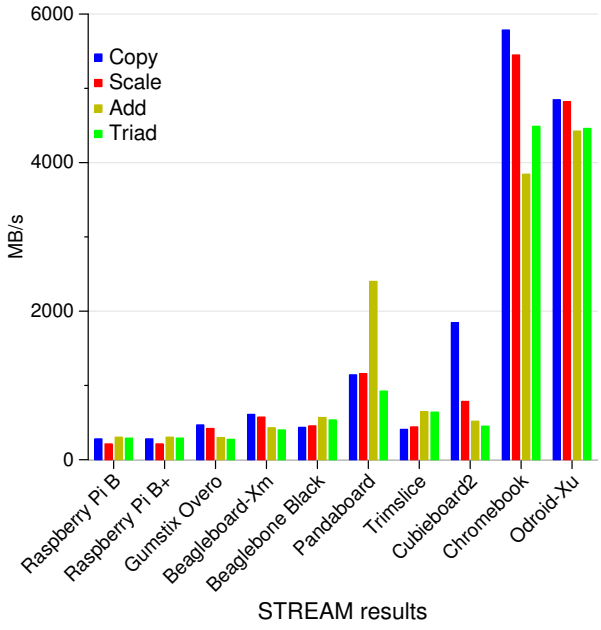

Fig. 3. STREAM benchmark results.

bit ARM systems in all metrics, from performance, performance per Watt, and performance per dollar. One would think this would make them the clear choice for creating a high-performance ARM cluster. In the end we chose the Raspberry Pi for our cluster. There are various reasons for this, mostly logistical. The Chromebook is in a laptop form factor and has no wired ethernet, making it an awkward choice for creating a large cluster. The Odroid-xU would seem to be the next logical choice, however at the time of writing they could only be purchased directly from a Korean company and it would not have been easy to source a large number of them through our university procurement system.

Therefore we chose the Raspberry Pi for use in our cluster, as it has excellent MFLOPS/Watt behavior, a high MFLOPS/dollar cost, and was easy to source in large numbers.

III. Cluster

Based on the analysis in Section II we chose the Raspberry Pi Model B as the basis of a 32-bit ARM cluster. The Raspberry Pi is a good candidate because of its small size, low cost, low power consumption, easy access to GPIO pins for external devices. Figure 4 shows a prototype version of the cluster in action. The core part of the 32-node cluster (compute nodes plus network switch) costs roughly $2200, power measurement adds roughly $200, and the visualization display cost an additional $700.

A. Node Installation And Software

Each node in the cluster consists of a Raspberry Pi Model B or B+ with its own 4GB SD card. Each node has an installation of the Raspbian operating system, which is based on Debian Linux and designed specifically for the Raspberry Pi. One node will be designated the head node and will provide outside Ethernet connectivity as well as additional disk storage.

Fig. 4. The prototype cluster.

The head node contains a file system which will be shared via a Network File System (NFS) server and subsequently mounted by the sub-nodes. Using NFS allows programs, packages, and features to be installed on a single file system and then shared throughout the network, which is faster and easier to maintain than manually copying files and programs around. Passwordless SSH (Secure Shell) allows easily running commands on the sub-nodes.

A Message Passing Interface (MPI) is installed on the cluster so that programs (such as the HPL benchmark) may run on multiple nodes at the same time. The MPI implementation used for this cluster is MPICH2, a free MPI distribution written for UNIX-like operating systems.

The nodes are connected by 100MB ethernet, consisting of a 48-port 10/100 network switch which draws approximately 20 Watts of power.

B. Node Arrangement and Construction

The initial cluster has 32 nodes but is designed so that expansion to 64 nodes and beyond is possible.

A Corsair CX430 ATX power supply powers the 32 nodes in the cluster. This power supply is capable of supplying 5V DC up to 20A. At maximum load, a single Raspberry Pi model B should draw no more than 700mA of current (and a model B+ will draw even less). We found it necessary to draw power from both the 5V and 12V lines of the power supply, otherwise the voltages provided would become unstable. To do this we utilize two 12V to 5V DC/DC converters to provide power to some of the nodes.

Power can be supplied to a Raspberry Pi in two ways, through the micro USB connector or through the GPIO pins. For this cluster, power will be distributed from the ATX power supply by connecting 5V and ground wires from the power supply's distribution wires to male micro USB connectors. The decision to use the micro USB sockets over the GPIO pins was made on account of the protective circuitry between the

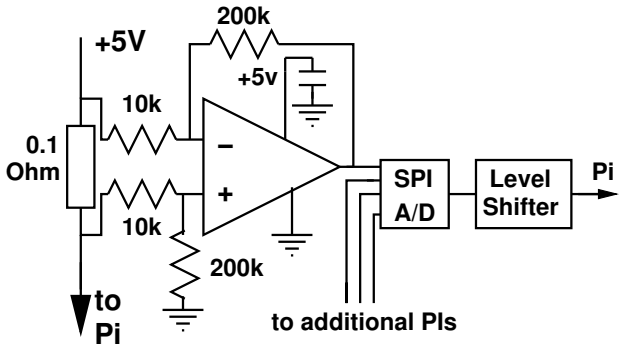

Fig. 5. The circuit used to measure power. An op-amp provides a gain of 20 to the voltage drop across a sense resistor. This can be used to calculate current and then power. The values from four Pis are fed to a measurement node using an SPI A/D converter.

micro USB socket and the rest of the Raspberry Pi, including fuses and smoothing capacitors. The GPIO pins have no such protection, and are essentially designed for supplying power to external devices.

The boards are connected via aluminum standoffs in stacks of eight. A large server case houses the cluster in one unit.

C. Visualization Displays

Two main external display devices will act as a front end for the cluster.

The first is a series of 1.2" bi-color 8x8 LED matrix displays attached to each node's GPIO ribbon cable. These LED displays will be programmed and controlled using the Raspberry Pi nodes' i2c interfaces. These per-node displays can be controlled in parallel via MPI programs. This will not only allow interesting visualization and per-node system information, but will provide the possibility for a more interactive and visual introduction for students learning MPI programming.

The second piece of the front end will be an LCD-PI32 3.2" LCD touchscreen that will be programmed and controlled using the head node's SPI interface. This screen will enable a user to view power and performance information as well as programs that are currently running and the status of all nodes in the cluster.

D. Power Measurement

Each node will have detailed power measurement provided by a circuit as shown in 5. The current consumed by each node will be calculated from the voltage drop across a 0.1 Ohm sense resistor which is amplified by 20 with an op-amp and then measured with an SPI A/D converter. Multiplying overall voltage by the calculated current will give the instantaneous power being consumed. The power values for four Pis will be fed into the SPI interface of a controlling Pi responsible for power measurement.

An example power measurement of a single-node HPL run is shown in Figures 6. Figure 7 shows the same results but with increased 100Hz sampling frequency; by having higher sample rates available more detailed power analysis can be conducted.

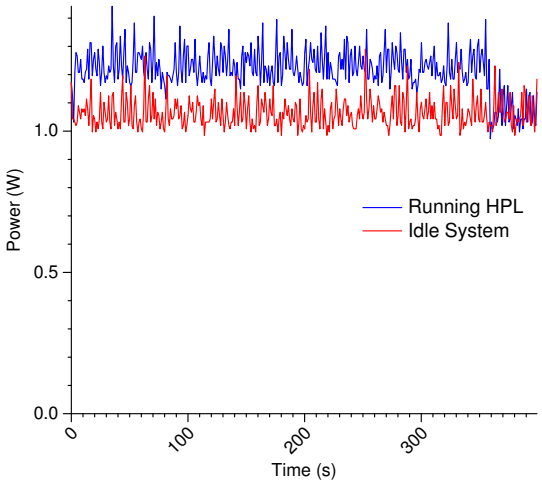

Fig. 6. Sample power measurement showing increase when running HPL on individual node.

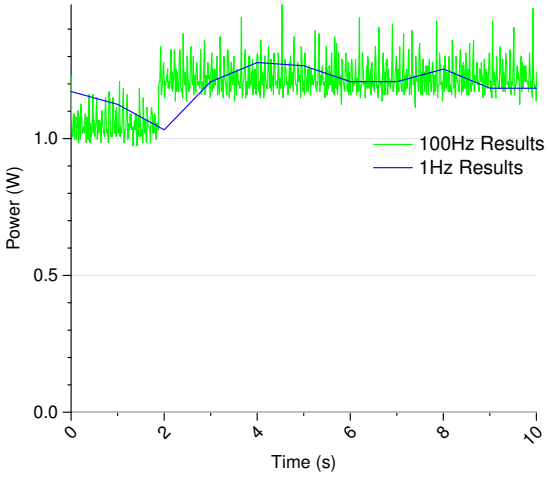

Fig. 7. Comparison of 1Hz vs 100Hz power measurement resolution.

IV. Cluster Results

We ran the HPL benchmark on our Raspberry Pi cluster while measuring the power consumption in order to calculate FLOPS per Watt metrics.

A. Peak FLOPS results

Table V shows the peak performance of our Pi cluster running with 32 boards (16 Model B, 16 Model B+). We find a peak of 4.37 GFLOPS while using 93.0W for a MFLOPS/W rating of 47.0. If we overclock the cluster to run at 1GHz we bump the peak performance to 6.25 GFLOPS while increasing the power to 112.1W, for a MFLOPS/W rating of 55.8 (we need to add cooling fans when overclocked; the extra power for those is included). As can be seen in the table, the MFLOPS/W is much less than that found on x86 servers. Part of this inefficiency is due to the 20W overhead of the network switch, which is amortized as more nodes are added. An additional

TABLE V. PEAK FLOPS CLUSTER COMPARISON.

Type	Nodes	Cores (Threads)	Frequency	Memory	Peak MFLOPS	N	Idle Power	Busy Power	MFLOPS per Watt	MFLOPS per $
Pi Cluster	32	32	700MHz	16 GB	4,370	35,000	86.8	93.0	47.0	1.99
Pi Cluster Overclocked	32	32	1GHz	16 GB	6,250	37,000	94.5	112.1	55.8	2.84
AMD Opteron 6376	1	16	2.3GHz	16 GB	85,300	40,000	162.1	343.53	247.2	21.60
Intel Sandybridge-EP	1	12	2.3GHz	16 GB	85,000	40,000	93.0	245.49	346.0	21.30

boost in the MFLOPS/W rating of our cluster would occur if we moved to use entirely the more efficient Model B+ boards.

B. Cluster Scaling

Figure 8 shows the performance results of our underclocked Modem B cluster scaling as more nodes are added. Adding nodes continues to increase performance in an almost linear fashion, this gives hope that we can continue to improve performance by adding more nodes to the cluster.

Figure 9 shows the performance per Watt numbers scaling as more nodes are added. This value is still increasing as nodes are added, but at a much lower level than pure performance. This is expected, as all of the boards have the same core MFLOPS/W value, so adding more is simply mitigating the static overhead power rather than making the cluster more efficient.

C. Summary

Our cluster shows decent floating point performance despite its low cost, and should serve as an excellent educational tool for conducting detailed ARM power/performance code optimization. Despite the benefits, if raw performance, performance per Watt, or performance per cost are the metrics of interest, then our cluster is easily defeated by x86 servers.

V. RELATED WORK

We perform a price, performance, and cost comparison of a large number of 32-bit ARM boards. We use those results to guide the design for a large, per-node instrumented, cluster for power and performance co-design. There is previous work in all of these areas, though not combined into one encompassing project.

A. Cluster Power Measurement

Other work has been done on gathering fine-grained cluster power measurement; usually the cluster in question runs x86 processors. Powerpack [10] is one such instrumented x86 cluster. The PowerMon2 [11] project describes small boards that can be used to instrument a large x86 cluster. Hackenberg et al. [12] describe many other techniques that can be used in such cases, including RAPL, but again primarily looking at x86 devices.

B. ARM HPC Performance Comparisons

Dongarra and Luszczek [13] were one of the first groups attempting to optimize for HPC performance on small boards; they created an iPad2 (Cortex A9) Linpack app showing performance was on par with early Cray supercomputers.

Aroca et al. [14] compare Pandaboard, Beagleboard, and various x86 boards with FLOPS and FLOPS/W. Their Pandaboard and Beagleboard performance numbers are much lower than the ones we measure. Jarus et al. [15] compare the power and energy efficiency of Cortex-A8 systems with x86 systems. Blem et al. [16] compare Pandaboard, Beagleboard and x86. Stanley-Marbell and Cabezas [17] compare Beagleboard, PowerPC, and x86 low-power systems for thermal and power. Pinto et al. [18] compare Atom x86 vs Cortex A9. Padoin et al. [19], [20], [21] compare various Cortex A8 and Cortex A9 boards. Pleiter and Richter [22] compare Pandaboard vs Tegra2. Laurenzano et al. [23] compare Cortex A9, Cortex A15 and Intel Sandybridge and measure power and performance on a wide variety of HPC benchmarks.

Our ARM comparisons are different from the previously mentioned work primarily by how many different boards (10) that we investigated.

C. ARM Cluster Building

There are many documented cases of compute clusters built from commodity 32-bit ARM boards. Many are just brief descriptions found online; we concentrate on those that include writeups with power and HPL performance numbers.

Rajovic et al. [24], [25] describe creating the Tibidabo cluster out of 128 Tegra 2 boards. They obtain 97 GFLOPS when running HPL on 96 nodes. Göddecke et al. [26] use this cluster on a wide variety of scientific applications and find the energy use compares favorably with an x86 cluster.

Sukaridhoto et al. [27] create a cluster out of 16 Pandaboard-ES boards. They run STREAM and HPL on it but do not take power measurements. Their STREAM results are much lower than ours, but the HPL FLOPS values are close.

Balakrishnan [28] investigates a 6-node Pandaboard cluster as well as a 2-node Raspberry Pi cluster. He uses a WattsUpPro like we do, but only runs HPL on the Pandaboard. He finds lower results than we do with STREAM, but his HPL results on Pandaboard are much higher than ours.

Ou et al. [29] create a 4-board Pandaboard cluster and measure energy and cost efficiency of web workloads on ARM compared to x86 servers.

Fürlinger et al. [30] build a cluster out of 4 AppleTV devices with Cortex A8 processors. They find 16MFlop/W. Their single-node HPL measurements are close to ours.

D. Raspberry Pi Clusters

Various groups have built Raspberry Pi clusters, we focus here on ones that were reported with HPL as a benchmark,

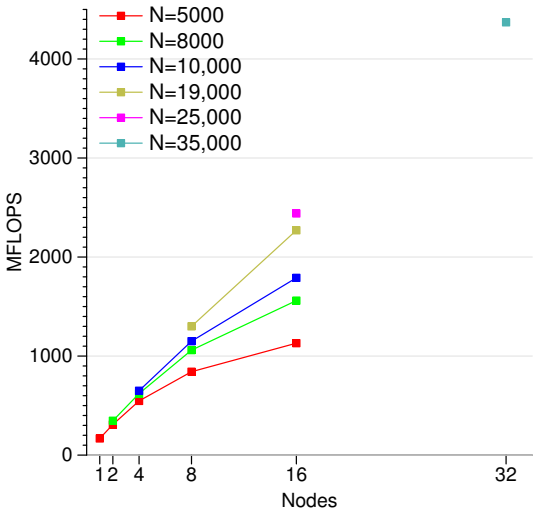

Fig. 8. MFLOPS with number of nodes for Model B cluster, no overclocking.

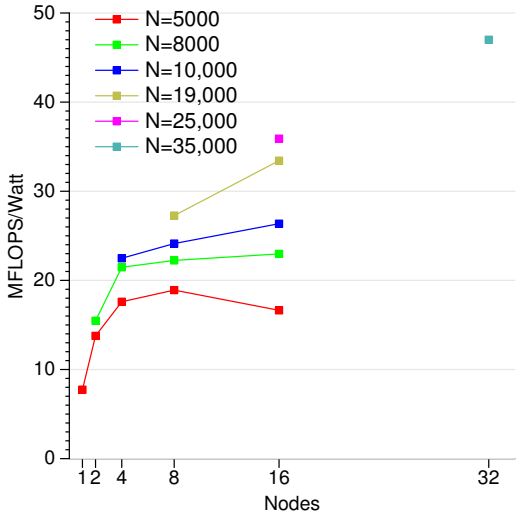

Fig. 9. MFLOPS per Watt with number of nodes for Model B cluster, no overclocking.

or else have large numbers of nodes. None of them are instrumented for per-node power measurements like ours is. Pfalzgraf and Driscoll [31] create a a 25-node Raspberry Pi cluster, but do not provide power or FLOPS results. Kiepert [32] builds a 32-node Raspberry Pi cluster. He includes total power usage of the cluster, but does not include floating point performance results. Tso et al. [33] build a 56-node Raspberry Pi "cloud" cluster. Cox et al. [34] construct a 64-node Raspberry Pi cluster. The obtain a peak performance of 1.14 GFLOPS, which is much less than we find with 32 nodes on our cluster. Abrahamsson et al. [35] built a 300-node Raspberry Pi cluster.

E. Summary

There is much existing related work; out work is different primarily in the number of boards investigated and in the per-node power measurement capabilities of the finished cluster.

One worrying trend found in the related works is the wide variation in performance measurements. For the various ARM boards the STREAM and HPL FLOPS results should be consistent, yet the various studies give widely varying results. Differences in HPL results are most likely due to different BLAS libraries being used, as well as the difficulty finding a "peak" `HPL.dat` file that gives the best performance. It is unclear why STREAM results differ so widely. Power measurement is also something that is hard to measure exactly, especially on embedded boards that use a variety of power supplies. Raspberry Pi machines in particular have no standard power supply; any USB supply (with unknown efficiency) can be used and since the total power being measured is small the efficiency of the supply can make a big difference in results.

VI. Future Work

We have much future work planned for our cluster.

- **Expand the size.** We have parts to expand to 48 nodes and can easily expand to 64 and beyond.
- **Enable hardware performance counter support.** Two of the authors have gotten Raspberry Pi performance counter support merged into upstream Linux, but this has not made it to distributions such as Raspbian yet. Having access to the counters in conjunction with the energy measurements will enable more thorough performance studies.
- **Harness the GPUs.** Table II shows the GPU capabilities available on the various boards. The Raspberry Pi has a potential 24 GFLOPS available *perf node* which is over an order of magnitude more than found on the CPU. Grasso et al. [36] use OpenCL on a Cortex A15 board with a Mali GPU and find they can get 8.7 times better performance than the CPU with 1/3 the energy. If similar work could be done to obtain GPGPU support on the Raspberry Pi our cluster could obtain a huge performance boost.
- **Perform power and performance optimization.** We now have the capability to do detailed performance and power optimizations on an ARM cluster. We need to develop new tools and methodologies to take advantage of this.

VII. Conclusion

We measure the power and performance tradeoffs found in ten different 32-bit ARM development boards. Upon careful consideration of the boards' merits, we choose the Raspberry Pi as the basis of an ARM HPC cluster. We design and build a 32-node cluster that has per-node real-time power measurement available. We then test the power and performance of the cluster; we find it to have reasonable numbers, but with lower absolute performance as well as performance per Watt when compared to an x86 server machine.

We plan to use this machine to enable advanced power and performance analysis of HPC workloads on ARM systems, both for educational and classroom use, as well as to gain experience in preparation for the coming use of ARM64 processors in server machines.

More details on the cluster can be found at our website: http://web.eece.maine.edu/~vweaver/projects/pi-cluster/

REFERENCES

[1] A. Gara, M. Blumrich, D. Chen, G.-T. Chiu, P. Coteus, M. Giampapa, R. R.A. Haring, P. Heidelberger, D. Hoenicke, G. Kopcsay, T. Liebsch, M. Ohmacht, B. Steinmacher-Burow, T. Takken, and P. Vranas, "Overview of the Blue Gene/L system architecture," *IBM Journal of Research and Development*, vol. 49, no. 2.3, pp. 195–212, 2005.

[2] G. Almasi *et al.*, "Overview of the IBM Blue Gene/P project," *IBM Journal of Research and Development*, vol. 52, no. 1.2, pp. 199–220, Jan. 2008.

[3] R. Haring, M. Ohmacht, T. Fox, M. Gschwind, P. Boyle, N. Chist, C. Kim, D. Satterfield, K. Sugavanam, P. Coteus, P. Heidelberger, M. Blumrich, R. Wisniewski, and G. Chiu, "The IBM Blue Gene/Q compute chip," *IEEE Micro*, vol. 22, pp. 48–60, Mar./Apr. 2012.

[4] A. Petitet, R. Whaley, J. Dongarra, and A. Cleary, "HPL — a portable implementation of the high-performance linpack benchmark for distributed-memory computers," Innovative Computing Laboratory, Computer Science Department, University of Tennessee, v2.0, http://www.netlib.org/benchmark/hpl/, Jan. 2008.

[5] "Top 500 supercomputing sites," http://www.top500.org/, 2014.

[6] W. Gropp, "MPICH2: A new start for MPI implementations," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Sep. 2002, p. 7.

[7] R. C. Whaley and J. Dongarra, "Automatically tuned linear algebra software," in *Proc. of Ninth SIAM Conference on Parallel Processing for Scientific Computing*, 1999.

[8] J. McCalpin, "STREAM: Sustainable memory bandwidth in high performance computers," http://www.cs.virginia.edu/stream/, 1999.

[9] Electronic Educational Devices, "Watts Up PRO," http://www.wattsupmeters.com/, May 2009.

[10] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. Cameron, "PowerPack: Energy profiling and analysis of high-performance systems and applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 6, May 2010.

[11] D. Bedard, R. Fowler, M. Linn, and A. Porterfield, "PowerMon 2: Fine-grained, integrated power measurement," Renaissance Computing Institute, Tech. Rep. TR-09-04, 2009.

[12] D. Hackenberg, T. Ilsche, R. Schoene, D. Molka, M. Schmidt, and W. E. Nagel, "Power measurement techniques on standard compute nodes: A quantitative comparison," in *Proc. IEEE International Symposium on Performance Analysis of Systems and Software*, Apr. 2013.

[13] J. Dongarra and P. Luszczek, "Anatomy of a globally recursive embedded LINPACK benchmark," in *Proc. of the 2012 IEEE High Performance Extreme Computing Conference*, Sep. 2012.

[14] R. Aroca and L. Gonçalves, "Towards green data centers: A comparison of x86 and ARM architectures power efficiency," *Journal of Parallel and Distributed Computing*, vol. 72, pp. 1770–1780, 2012.

[15] M. Jarus, S. Varette, A. Oleksiak, and P. Bouvry, "Performance evaluation and energy efficiency of high-density HPC platforms based on Intel, AMD and ARM processors," *Energy Efficiency in Large Scale Distributed Systems*, pp. 182–200, 2013.

[16] E. Blem, J. Menon, and K. Sankaralingam, "Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures," in *Proc of IEEE International Symposium on High Performance Computer Architecture*, Feb. 2013, pp. 1–12.

[17] P. Stanley-Marbell and V. Cabezas, "Performance, power, and thermal analysis of low-power processors for scale-out systems," in *Proc of IEEE International Symposium on Parallel and Distributed Processing*, May 2011, pp. 863–870.

[18] V. Pinto, A. Lorenzon, A. Beck, N. Maillard, and P. Navaux, "Energy efficiency evaluation of multi-level parallelism on low power processors," in *Proc of Congresso da Sociedade Brasileira de Computação*, 2014, pp. 1825–1836.

[19] E. Padoin, D. de Olivera, P. Velho, and P. Navaux, "Evaluating performance and energy on ARM-based clusters for high performance computing," in *International Conference on Parallel Processing Workshops*, Sep. 2012.

[20] ——, "Evaluating energy efficiency and instantaneous power on ARM plafforms," in *Proc. of 10th Workshop on Parallel and Distributed Processing*, Aug. 2012.

[21] E. Padoin, D. de Olivera, P. Velho, P. Navaux, B. Videau, A. Degomme, and J.-F. Mehaut, "Scalability and energy efficiency of HPC cluster with ARM MPSoC," in *Proc. of 11th Workshop on Parallel and Distributed Processing*, Aug. 2013.

[22] D. Pleiter and M. Richter, "Energy efficient high-performance computing using ARM Cortex-A9 cores," in *Proc. of IEEE International Conference on Green Computing and Communications*, Nov. 2012, pp. 607–610.

[23] M. Laurenzano, A. Tiwari, A. Jundt, J. Peraza, W. Ward Jr., R. Campbell, and L. Carrington, "Characterizing the performance-energy tradeoff of small ARM cores in HPC computation," in *Proc. of Euro-Par 2014*, Aug. 2014, pp. 124–137.

[24] N. Rajovic, A. Rico, J. Vipond, I. Gelado, N. Puzovic, and A. Ramirez, "Experiences with mobile processors for energy efficient HPC," in *Proc. of the Conference on Design, Automation and Test in Europe*, Mar. 2013.

[25] N. Rajovic, A. Rico, N. Puzovic, and C. Adeniyi-Jones, "Tibidabo: Making the case for an ARM-based HPC system," *Future Generation Computer Systems*, vol. 36, pp. 322–334, 2014.

[26] D. Göddeke, D. Komatitsch, M. Geveler, D. Ribbrock, N. Rajovic, N. Puzovic, and A. Ramirez, "Energy efficiency vs. performance of the numerical solution of pdes: An application study on a low-power ARM based cluster," *Journal of Computational Physics*, vol. 237, pp. 132–150, 2013.

[27] S. Sukaridhoto, A. KHalilullah, and D. Pramadihato, "Further investigation of building and benchmarking a low power embedded cluster for education," in *Proc. International Seminar on Applied Technology, Science and Art*, 2013, pp. 1 – 8.

[28] N. Balakrishnan, "Building and benchmarking a low power ARM cluster," Master's thesis, University of Edinburgh, Aug. 2012.

[29] Z. Ou, B. Pang, Y. Deng, J. Nurminen, A. Ylä-Jääski, and P. Hui, "Energy- and cost-efficiency analysis of ARM-based clusters," in *Proc. of IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 2012, pp. 115–123.

[30] K. Fürlinger, C. Klausecker, and D. Kranzmüller, "The AppleTV-cluster: Towards energy efficient parallel computing on consumer electronic devices," Ludwig-Maximilians-Universität, Tech. Rep., Apr. 2011.

[31] A. Pfalzgraf and J. Driscoll, "A low-cost computer cluster for high-performance computing education," in *Proc. IEEEE International Conference on Electro/Information Technology*, Jun. 2014, pp. 362 –366.

[32] J. Kiepert, "RPiCLUSTER: Creating a raspberry pi-based beowulf cluster," Boise State University, Tech. Rep., May 2013.

[33] F. Tso, D. White, S. Jouet, J. Singer, and D. Pezaros, "The Glasgow Raspberry Pi cloud: A scale model for cloud computing infrastructures," in *Proc. of IEEE International Conference on Distributed Computing Systems Workshops*, Jul. 2013, pp. 108–112.

[34] S. Cox, J. Cox, R. Boardman, S. Johnston, M. Scott, and N. O'Brien, "Irdis-pi: a low-cost, compact demonstration cluster," *Cluster Computing*, vol. 17, pp. 349–358, Jun. 2013.

[35] P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nocolodi, N. Preda, L. Miori, M. Angriman, J. Rikkilä, X. Wang, K. Hamily, and S. Bugoloni, "Affordable and energy-efficient cloud computing clusters: The Bolzano Raspberry Pi cloud cluster experiment," in *Proc. of the IEEE International Conference on Cloud Computing Technology and Science*, Dec. 2013, pp. 170–175.

[36] I. Grasso, P. Radojković, N. Rajović, I. Gelado, and A. Ramirez, "Energy efficient HPC on embedded SoCs: Optimization techniques for Mali GPU," May 2014, pp. 123 – 132.