The MIST, a local, secure cloud context and 802.11s testbed

G. Dempsey [1], R. Feher [1], L. Gordon [2], K. Keville [3]
[1] Department of Systems Engineering, USMA, West Point, NY
Gregory.Dempsey@usma.edu, Ronald.Feher@usma.edu
[2] Department of Physics, USMA, West Point, NY
Lindsay.Gordon@usma.edu
[3] Institute for Soldier Nanotechnologies, MIT, Cambridge, MA
KLK@mit.edu

Abstract

This poster describes the evolution, use, and utility of a low-level cloud (a "MIST") built out of MIPS and ARM-based embedded Linux devices. Most of these devices started out as Wireless Access Points but have found a new niche as infrastructure support processors. They are cheap and low-powered and can therefore be fielded in places to do supercomputing support for previously unsupported applications.

Introduction

There is a class of HPC problems for which small "underpowered" clusters are ideally suited. The descriptor "underpowered" is inappropriate in this case; the CPU is balanced to the chokepoints of RAM and disk subsystems due not only to the low clock speeds and meager functionality of the CPU but also to the inadequacies of matching memory access speeds from which even high-end systems suffer. These HPC problems include those that require considerable I/O but little processing. In May 2009, researchers at Carnegie Mellon University demonstrated a system entitled FAWN (a "Flexible Array of Wimpy Nodes"), which represented a proof of the concept of efficient management of scan-bound and seek-bound workloads [1]. We have constructed a cluster following some of the tenets established with this new class of low-power designs, creating a scalable mesh of purpose-built nodes to test throughput and processing of short messages. As the next generation of low-power clusters demonstrate their value in managing certain workloads, we endeavour to optimize the hardware designs and queuing systems. Part of the impetus of this project was to determine the class of problems for which low-power clouds are best suited; that is to say, whether they could hold their own and compete on an equal footing with standard HPC clusters in raw OP/s, OP/s/$, or OP/s per Watt (or Joule) for certain applications.

Why MIPS or Arm?

There have been many methods developed over the years to mask the disparity in speeds and bus widths between the processor and the memory subsystems. Backus [2], for instance, coined the expression 'von Neumann bottleneck" to describe the serial nature of communication between the host processor and main memory. Methods such as pre-fetching and branch-prediction are two of the better known remedies for this I/O misalignment. These are ways to get incrementally better performance out of a processor by getting more operations per cycle that, in the aggregate, will reduce overall cycle use for a given process. Low-power processors solve this problem by not engaging it. Here again, in the aggregate, a higher node count makes up for the performance inadequacies of the individual node. Note the details of the processors in our test setups described below. The Netgear WGT634u was popular with the OpenWRT programming community because of its relatively large RAM sizes but otherwise does not compare favorably with embedded Linux devices.

Table 1: Mesh Test Setups

	Platform	
	Netgear WGT634u	**Marvell SheevaPlug**
CPU	200Mhz Broadcom BCM3302	1.2Ghz Feroceon 88FR131
BogoMIPS	198.65	1192.75
Flash	32M	512M
RAM	8M	512M
NIC	FE	GE
Nodes	8 - 35	10

IEEE 802.11s Mesh

Our initial benchmarking applications had no math component. We wanted to observe the performance of a large (large enough to generate sufficient traffic to create a "noisy" environment) wireless mesh utilizing the nascent 802.11s protocol as a primary wireless protocol, although 802.11g would be running as well. We ran enough traffic over the network to create a considerably noisy environment and thereby had a substantial number of packet collisions to force the Mesh to adapt to the environment.

The Test Suite

Prior work has been done on testing apps that mimic the operations of Facebook's memcached or Google's Hadoop / MapReduce scheme. Our work stayed within the confines of network testing; standard TCP test using nttcp, and netpipe over mpi. Performance numbers were predictably disparate. A 2.6.24 kernel compile took 369m27s. An identical kernel compile on a single SheevaPlug took 25m16s. Note that the (emulated) floating point performance on the MIPS machines was substantially worse that those on the ARM9 based boxes. The Broadcom MIPS processors are "wimpy" indeed. While there is a choice of 3 floating-point emulators on the SheevaPlug (NWFPE, VPE, and the new FastFPE) you are generally stuck with the Algorithmics/MIPS FPU Emulator on the

WGT634u. Ethernet tests were encouraging on the SheevaPlug. The SheevaPlug often got better than half their specified upper end. Nttcp and netpipe indicated better than 500 Mbps in back-to-back and better than 450 Mbps in switched fabric tests. Netpipe numbers across mesh connections, however, were extraordinarily bad. Even when we forced the connections into the highest available rate (54Mbps), we rarely got above 6Mbps peak in pairwise tests. Iperf tests met with similar performance. We plan on building a considerably larger WGT634u cluster (on the order of 200 nodes) in July 2010 to try and improve our mesh throughput at which time we will upgrade the wireless drivers and compat-wireless libraries if necessary.

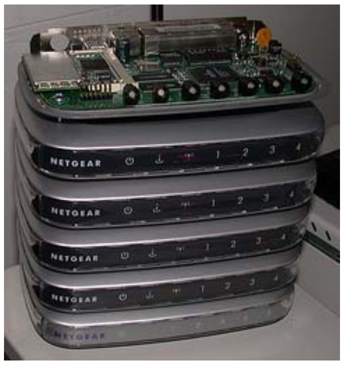
Figure 1: "Microclusters" of the WGT634u. One of our test setups created a tree of 35 nodes; seven sub-clusters of five nodes each. This was so we could fully involve an 8 port switch (counting uplink to the head node).

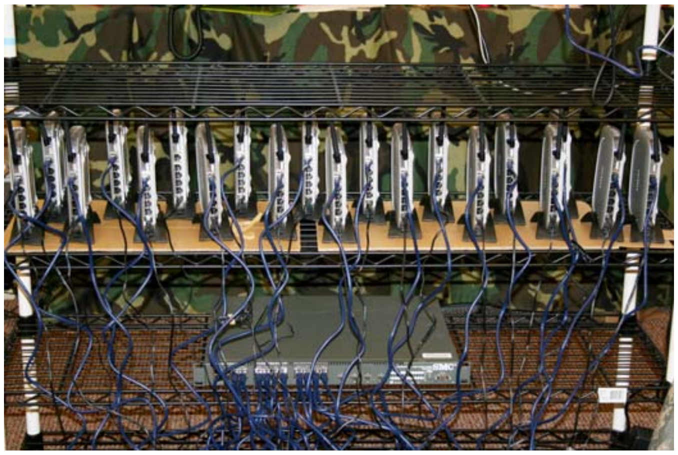
Figure 2: The 24 WGT634u node test setup on a switched fabric.

Solar-Powered Supercomputing

We have run these clusters a number of times off of a live Solar Panel for demonstration purposes. It is unclear that this would ever be of any utility, but it does point out the low powered aspects of this approach. We have certainly tested the same hardware in the field running their native OS and application (Open-Mesh with 802.11s or OLSR routing). In fact, the platform of choice for wireless network neighborhoods nowadays is the Accton board running a processor from the Atheros SoC family. The AR5312 is a popular processor in the OpenWRT community. In June 2009, we demonstrated 16 WGT634u running off of a 200W solar panel and later that month we recreated the demo with 10 SheevaPlugs. Average usage was 77W on the SheevaPlug cluster. Power usage was a little higher than we predicted on WGT634us. The WGT634us were easier to work with since they were a 12v design. The SheevaPlugs are 5v so we had to design an unusual circuit / charge controller for them. We anticipate testing the Tensilica and TI OMAP processor under a similar methodology in the near future.

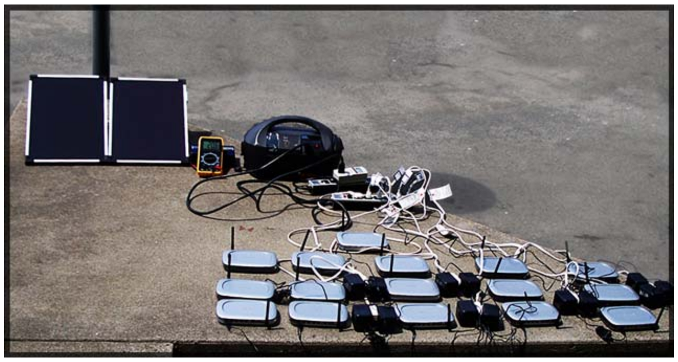
Figure 3: 16 of theWGT634u intercommunicating via 802.11g and 802.11s. This was with the primary bandwidth test device, wlan0, which becomes mesh0 under 802.11s.

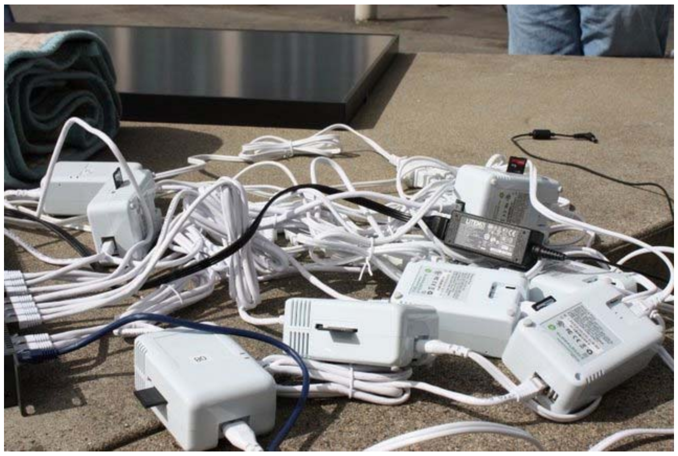
Figure 4: Pile of 10 SheevaPlugs powered by a solar panel in the background. Even with a power-hungry switch this ran at well under 100W.

References

[1] Vijay Vasudevan, Jason Franklin, David Andersen, Amar Phanishayee, Lawrence Tan, Michael Kaminsky, and Iulian Moraru, FAWNdamentally Power-efficient Clusters, *Proc. 12th Workshop on Hot Topics in Operating Systems (HotOS XII)*, Monte Verita, May 2009.

[2] Backus, J. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Communications of the ACM* 21, 8, (August 1978), 613-641.