Wire Speed Programmable Networks

[Extended Abstract]

Sandeep Pathivada
School of Computer Science
Telecommunication Research Centre
Trinity College Dublin
pathivas@tcd.ie

Prof. Donal O'Mahony
School of Computer Science
Telecommunication Research Centre
Trinity College Dublin
donal.omahony@cs.tcd.ie

ABSTRACT

This paper is a work in progress report for developing programmable wire speed networks using hybrid processor architectures. There is a lot interest in the networking research community to create faster software routers for experimental purposes and for developing open network infrastructure. Some of these solutions include using coprocessors on network interface cards; accelerating packet processing operations on FPGA's, GPU's etc. In this research project we are investigating software engineering and performance issues around packet processing on multicore and hybrid architectures. We would like to leverage the architectural features of IBM PowerEN processor for developing highly programmable wirespeed network devices.

Keywords

multicore network scalability, network processors,hybrid processor architectures, software routers, software defined networking

1. OPEN NETWORKING

Developing high performance network devices typically requires special hardware like ASICs, FPGA's or network processors to deliver required throughput. Existing network devices offer little or no programmability which prevents researchers in conducting networking experiments on Internet scale. Some solutions proposed by the networking community to address these challenges include:

- OpenFlow [10] is an innovation from Stanford which lets researches use slices of existing networking infrastructure for experiments. OpenFlow comes with a protocol specification which is implemented by most of the existing network vendors and provides a simple mechanism to program flow tables on these network devices using a central controller.
- Click [8] is a modular software router which can be used to build complex packet processing systems. Click scripts are modelled as directed graphs of various elements each performing small operations (eg decrementing TTL, computing checksum etc). Click makes use of push and pull mechanisms to direct packet flow. SMP-Click [1] is a multi-threaded variant of click which runs on multiple cores.
- NetFPGA [9] provides a platform for developing 1/10G line rate switches and routers using verilog code. NetFPGA is a custom built PCI card which contains a Xilinx FPGA, network interfaces and memory. There are attempts to compile click scripts on NetFPGA [12] for faster prototyping.

2. DISCUSSION

Greater computational power can achieved by using multiple processors or cores in parallel. There still remains an open ended question on how best we can harness this parallelism for processing network intensive workloads. Existing research focuses mostly using operating systems running in SMP mode. Another possible solution is to run separate embedded stacks on multiple cores for data and control flow(AMP mode) (Eg: WindRiver Packet Acceleration Solution [15]).

Strategies like Receive Packet Steering [2], Receive Flow Steering [4] and Microsoft's version of Receive-Side Scaling [11] spread incoming network packets across multiple cores. These strategies employ a hash on incoming packets to queue them for processing packet flows on CPU cores maximising cache and thread affinity.

Routebricks [3] has demonstrated that it is possible to build high performance software routers using multiple servers powered by intel multicore processors. PacketShader [7] is a 40Gbps software router which makes use of two Intel Xeon X550 Quad core processors and two Nvidia GTX285 connected in NUMA mode. The prototype server has 4x10GbE intel NIC's and a total of 12G DDR3 memory. The software framework has an optimised NUMA aware Packet I/O engine using batch packet processing and a pipelined packet processing framework which nicely scales across all CPU and GPU cores. A recent survey on obtaining 10G line rate packet processing on commodity hardware is done in [14]. API's like NetMap, PF_RING DNA etc. which provide fast network access to user space programs are explained in this paper.

In our research we are looking at an alternative approach for developing wirespeed programmable networks using hybrid processor architectures. Traditionally network interface cards are treated as second-class citizens which attach themselves to a peripheral I/O bus like PCI-Express. Even with the latest multicore cpu's with fast interconnects such as Intel QuickPath Interconnect(QPI) or AMD HyperTransport researchers have hit a limit on the maximum scalability software based routers can achieve [6]. Also most of these proposed solutions are designed with 10G interfaces in mind. In the future with 40G and 100G interfaces available on the network edge the networking community should also investigate about scalable hardware architectures for performing packet processing operations at such high speeds. Packet processing on such hybrid hardware architectures incur several software engineering and throughput related issues which will be addressed as a part of this project.

3. COLLABORATION WITH IBM RESEARCH

IBM has recently unveiled PowerEN wirespeed network processor[5] which is a hybrid network server processor. It has 16 cores, 64 hardware threads and accelerator engines for performing compute intensive operations like encryption, compression, regex lookups and for processing XML streams. The processor also features a Host Ethernet Adapter (HEA) interfacing with four 10GigE ports. There are 2 DDR3 controllers in the processor which can host upto a maximum of 64Gbytes of memory. PowerEN processor's HEA can be run either in network mode(switch/router) or in end point mode(server) supporting features like multiple hardware queues; checksum assist etc. Two/Four PowerEN processors can be connected in SMP mode offering a total of 128/256 hardware threads suitable for processing highly parallel network workloads.

From software point of view this processor exposes a uniform memory addressing scheme spanning all software threads and accelerators. The development model is familiar to existing linux application development owing to POSIX API compatibility. This provides a significant advantage in development over existing network processors which use proprietary languages to access underlying hardware capabilities (e.g. Intel microcode, Agere Scripting Language etc). Access to accelerators and networking operations can be performed directly from user space which is highly useful for rapid prototyping networking applications in the areas of network security, data centre networking research etc.

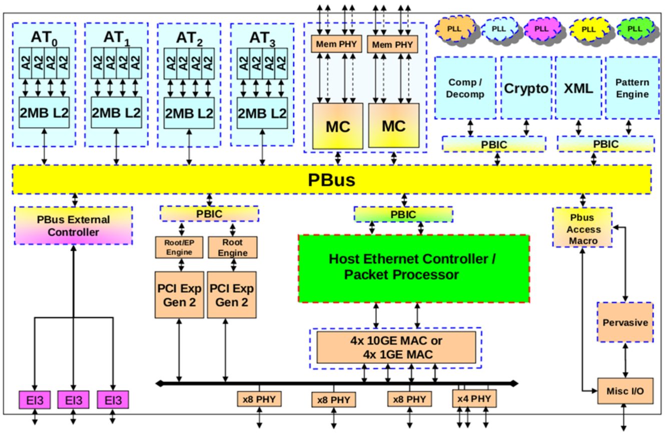

Figure 1: PowerEN Chip Diagram [13]

4. ACKNOWLEDGEMENTS

This project is funded by IRCSET & IBM Research under Enterprise Partnership Scheme.

5. REFERENCES

[1] B. Chen and R. Morris. Flexible Control of Parallelism in a Multiprocessor PC Router. pages 333–346, June 2001.

[2] J. Corbet. Receive packet steering [LWN.net]. `http://lwn.net/Articles/362339/`.

[3] M. Dobrescu, N. Egi, K. Argyraki, B. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, and S. Ratnasamy. RouteBricks: Exploiting parallelism to scale software routers. In *ACM SOSP*, pages 15–28. Citeseer, 2009.

[4] J. Edge. Receive flow steering [LWN.net]. `http://lwn.net/Articles/382428/`.

[5] H. Franke, J. Xenidis, C. Basso, B. M. Bass, S. S. Woodward, J. D. Brown, and C. L. Johnson. Introduction to the wire-speed processor and architecture. *IBM Journal of Research and Development*, 54(1):1–11, Jan. 2010.

[6] S. Han, K. Jang, K. Park, and S. Moon. Building a single-box 100 gbps software router. In *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on*, pages 1–4. IEEE, 2010.

[7] S. Han, K. Jang, K. Park, and S. Moon. Packetshader: a gpu-accelerated software router. *ACM SIGCOMM Computer Communication Review*, 40(4):195–206, 2010.

[8] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, Aug. 2000.

[9] J. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo. Netfpga–an open platform for gigabit-rate network switching and routing. 2007.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[11] Microsoft. Receive-Side Scaling. `http://technet.microsoft.com/en-us/network/dd277646`.

[12] P. Nikander, B. Nyman, T. Rinta-aho, S. Sahasrabuddhe, and J. Kempf. Towards software-defined silicon: Experiences in compiling click to netfpga. In *1st European NetFPGA Developers Workshop, Cambridge, UK*, 2010.

[13] I. Research. A Wire-Speed Power Processor: 2.3GHz 45nm SOI with 16 Cores and 64 Threads Ű Presentation. `http://www.power.org/events/2010_ISSCC/Wire_Speed_Presentation_5.5_-_Final4.pdf`.

[14] L. Rizzo, L. Deri, and A. Cardigliano. 10 gbit/s line rate packet processing using commodity hardware: Survey and new proposals.

[15] WindRiver. Packet Acceleration Solution. `http://www.windriver.com/solutions/network-equipment/packet.html`. [Online; accessed 22-July-2011].