PacketShader

Massively-Parallel Packet Processing with GPUs to Accelerate Software Routers

Sangjin Han[1], Keon Jang[1], KyoungSoo Park[2], and Sue Moon[1]

[1]Department of Computer Science, KAIST, {sangjin, keonjang}@an.kaist.ac.kr, sbmoon@kaist.edu

[2]Department of Electrical Engineering, KAIST, kyoungsoo@ee.kaist.ac.kr

PacketShader

- The fastest software router in the world
  - » 4.5x faster than RouteBricks [SOSP '09]
- Eliminates the CPU bottleneck
  - » with massively-parallel GPU computation
- Highly scalable Packet I/O
  - » 40 Gbps throughput even with 64B packets
- Flexible architecture
  - » Runs in user-mode on commodity x86 servers
- Implementations: IPv4, IPv6, OpenFlow, and IPsec

Motivation

PC-based Software Routers

- Easy to fix problems
- Ready for additional functionalities
- Cheap & Ubiquitous ($1,000 ~ $10, 000)
- Low Performance: 1-10 Gbps today

General-Purpose Computation on GPUs

- GPUs are widely used for data-intensive workloads
  - » E.g. Medical imaging, bioinformatics, finance, etc.
- High performance with massively-parallel processing

	Price	# of cores	# of HW threads	Peak performance
CPU (Intel Core i7 920)	$260	4	8	43 GFLOPS
GPU (NVIDIA GTX280)	$260	240	30,720	933 GFLOPS

Our key insight on GPU packet processing:

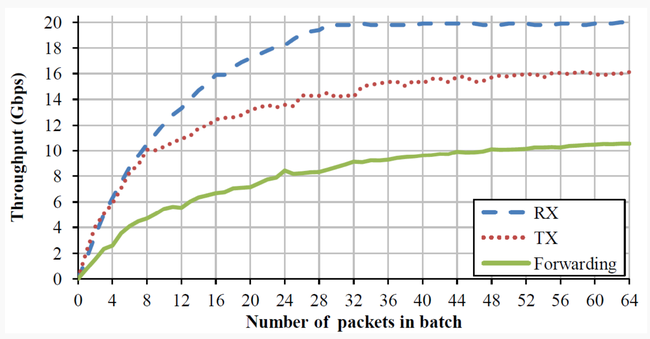
"The massive core array of GPUs matches the inherent parallelism in stateless packet processing."

Design

Data-path in a software router = Packet I/O + Packet Processing

Our Packet I/O Engine

- Huge packet buffer
  - » instead of per-packet buffer
- Batch packet processing
  - » effectively amortizes per-peacket cost

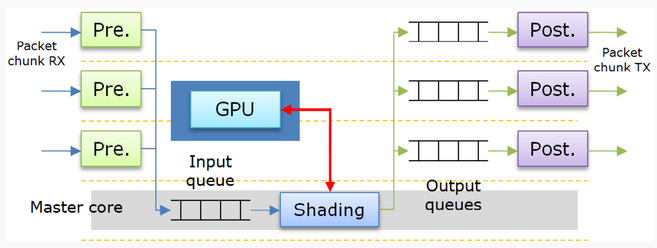


I/O batch performance with a single CPU core

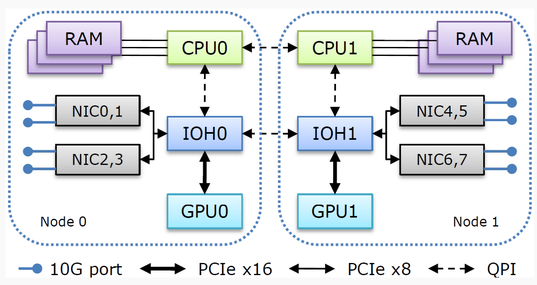
- Multi-core/NUMA scalable
  - » RSS and NUMA-aware data placement
- Efficient user-level interface
  - » supports multiple queues
  - » resistant to receive livelock

GPU Processing Framework

- General framework
  - » Provides a convenient way for parallel packet processing
  - » Applications (e.g. IPv4 routing) are implemented on it
- Highly scalable
  - » with multiple CPU cores, NUMA nodes, NICs, and GPUs
- Optimization supports
  - » Pipelining in CPU-GPU collaboration
  - » Gather/Scatter for more parallelism
- 3-stage workflow



Hardware Setup



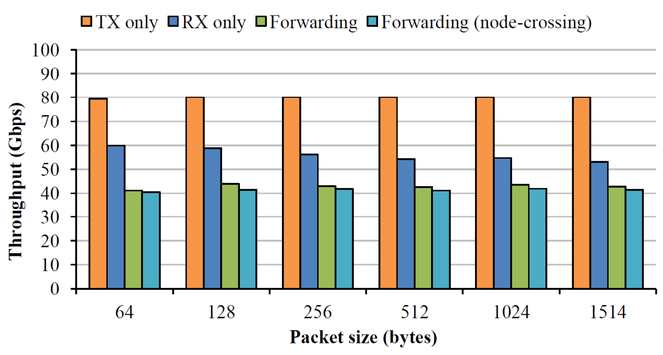
Item	Specification	Qty
CPU	Xeon X5550 (quad-core 2.66GHz)	2
RAM	DDR3 ECC FBDIMM 2GB 1,333Mhz	6
Motherboard	Super Micro X8DAH+	1
Graphics card	NVIDIA GTX285 (240 cores, 1GB DDR3)	2
NIC	Intel X520-DA2 (dual-port 10GbE)	4

The interior view of our prototype server (equipped with 2 quad-core CPUs, 2 GPUs, 8×10GbE ports)

Our prototype costs less than $7,000, which is 10+ times cheaper than hardware-based commercial routers with similar performance.

Performance

- Basic Packet I/O performance
  - » 40 Gbps forwarding for any packet sizes



- We implemented 4 applications on top of PacketShader and measured performance.

Application	What we offloaded	Type	CPU-only[1]	CPU+GPU[1]
IPv4 routing[2]	IPv4 longest prefix matching	Simple	28.8 Gbps	**38.1 Gbps**[4]
IPv6 routing[2]	IPv6 longest prefix matching	Memory-intensive	8.1 Gbps	**31.4 Gbps**
OpenFlow switch[3]	Wildcard matching	Configuration dependent	9.6 Gbps	**24.6 Gbps**
IPsec gateway	Encryption and HMAC	Compute-intensive	2.8 Gbps	**5.3 Gbps**

1) All throughputs are measured with **min-sized (64B) packets**.
2) Configured with 300K IPv4 prefixes and 200K IPv6 prefixes
3) Configured with 64K exact-match flows and 64 wildcard-match flows.
4) This performance is **4.5x higher than the fastest software router** previously reported (RouteBricks in SOSP '09)

This work was supported by NAP of Korea Research Council of Fundamental Science & Technology (KRCF).