효율적인 패킷 병렬처리를 위한 멀티코어 기반 라우터에 대한 고찰

최대진, 한진영, 임영빈, 권태경*, 최양희*
서울대학교 컴퓨터 공학부
{djchoi, jyhan, ybim}@mmlab.snu.ac.kr, *{tkkwon, yhchoi}@snu.ac.kr

A Study on the Multi-core based Router for Efficient Parallel Packet Processing

Daejin Choi, Jinyoung Han, Youngbin Im, Ted "Taekyong" Kwon, Yanghee Choi
School of Computer Science and Engineering, Seoul National University

요 약

최근 인터넷 트래픽이 폭발적으로 증가함에 따라 수많은 패킷들을 라우터가 효과적으로 전달하는 것에 대한 중요성이 더욱 커져가고 있다. 이에 따라, 많은 패킷에 대한 병렬 처리를 통해 라우터의 전달 성능을 높이고자 하는 멀티코어 기반의 라우터 개발이 주목을 받고 있으며, 대표적으로 GPU 기반의 멀티코어 라우터와 CPU 기반의 멀티코어 라우터가 제안되었다. GPU 기반의 멀티코어 라우터는 각 패킷에 대한 다음 라우터(next hop) 검색을 GPU의 많은 코어를 이용해서 병렬적으로 수행할 수 있지만, PCIe bus를 통하여 CPU, NIC (Network Interface Card), 그리고 메모리간에 많은 데이터를 이동시키기 때문에 패킷수가 많아짐에 따라 성능에 한계가 있다. 반면 CPU 기반의 멀티코어 라우터는 GPU에 비해서 PCIe bus 사용량이 적기 때문에 코어 개수가 증가할수록 포워딩 성능이 확장가능(scalable) 하지만, NIC과의 데이터 교환시에 PCIe bus를 통하여 동작하기 때문에 결국 성능에 한계점을 가지고 있다. 본 논문에서는 두 시스템의 특징과 한계점을 명시하고, 이를 해결하기 위한 방안을 제시한다. 본 연구의 고찰은 인터넷 트래픽 폭증 문제를 라우터 관점에서 해결하기 위한 연구들의 초석이 될 것으로 기대된다.

Ⅰ. 서 론

급증하는 인터넷 사용자와 트래픽으로 인해 패킷 전달을 담당하는 라우터의 전달 성능 향상에 대한 중요성이 꾸준히 강조되고 있다. 라우터의 전달 성능향상을 위해서 다음 두 가지의 접근법으로 연구가 진행되고 있다: (1) FIB (Forwarding Information Base)의 검색 (Lookup) 알고리즘 속도 향상을 통한 단일 패킷의 프로세싱의 성능 향상 방법 [1], (2) 많은 패킷들을 동시에 효율적으로 처리하기 위한 병렬 처리(Parallel Processing) 방법[2][3].

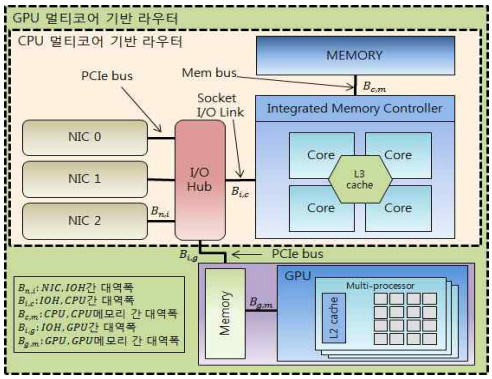
그 중에서도 처리할 패킷이 증가함에 따라 라우터의 계산능력이 중요해진다는 연구 결과[3]와 함께 다수의 코어를 이용한 병렬처리에 대한 가능성이 부각 되고 있고, 이와 함께 GPU 기반의 멀티코어 라우터와 CPU 기반의 멀티코어 라우터가 제안되었다[2][3]. 하지만 제시된 두 시스템 모두 패킷 프로세싱을 위한 데이터 이동 경로(PCIe)에서의 병목현상으로 인해 일정 수준 이상으로 코어의 이용률을 높이지 못하는 한계점을 가지고 있고, 프로세서의 증가 추세에 따른 라우터의 전달 성능에 대한 확장성 (Scalability)에서 문제가 있다.

이러한 문제를 해결하기 위해 본 논문에서는 제안된 두 시스템의 특징과 성능을 최근 관련 연구를 기반으로 비교하고 두 시스템의 구조적 한계를 보완하여 병렬 처리 성능을 높일 수 있는 새로운 시스템을 제안한다. 멀티코어 기반의 라우터의 디자인에 대한 본 연구의 논의를 통해 증가하는 트래픽 문제를 해결하기 위한 연구들의 초석을 제공할 것이라고 기대한다.

Ⅱ. 본론

2.1. CPU 및 GPU 기반의 멀티코어 라우터

멀티코어 시스템은 다수의 프로세서를 이용하여 많은 연산을 분산 및 병렬처리 할 수 있는 시스템이다. 라우팅 연산은 모든 패킷들에 독립적으로 적용되기 때문에, 멀티코어 기반 라우터는 효과적으로 패킷 병렬 처리를 할 수 있도록 연구되어 왔고, CPU 및 GPU 기반의 멀티코어 라우터가 최근 주목받기 시작하였다[2][3]. GPU는 그래픽 카드에 존재하는 프로세서로써 많은 양의 산술연산을 필요로 하는 물리 엔진의 동작이나 영상처리 등을 효과적으로 처리할 수 있도록 설계된 하드웨어이다. GPU가 많은 데이터의 산술연산을 빠른 시간에 처리할 수 있다는 장점 때문에, 물리, 기계, 생물학 등 다양한 분야에서 활용되고 있다. 본 장에서는 CPU 및 GPU 기반 라우터의 구조 및 특징을 설명한다.



〔그림 1〕 CPU 및 GPU 멀티코어 기반 라우터 구조

그림 1에서 CPU 및 GPU 기반의 라우터의 구조를 묘사하고 있다. NICs (Network Interface Cards)로부터 패킷을 받아 실제 데이터 (Payload)를 IOH (IO Hub)를 통해 메모리에 저장시키고, 동시에 라우팅 수행을 위한 별도의 패킷 헤더를 생성한다. 이후 생성된 헤더 정보의 처리는 CPU 및 GPU 기반의 멀티코어 라우터가 각각 다르게 처리한다.

CPU 멀티코어 라우터에서는 생성된 헤더 정보를 NICs로부터 IOH로 PCIe bus(대역폭: $B_{n,i}$)를 통해 이동시키며 (시간 비용: T_1), IOH로부터 각 CPU로 소켓 I/O 링크(대역폭: $B_{i,c}$)를 통하여 전달된다 (시간 비용: T_2). 이후에 전달된 데이터를 이용하여 라우팅 알고리즘을 수행한 후에,

다시 IOH와 NIC를 거쳐서 패킷을 다음 라우터로 이동시킨다.

GPU 기반의 멀티코어 라우터에서 패킷 프로세싱 과정은 CPU의 경우와 유사하지만, 그림 1의 바깥쪽 사각형에 나타난 것처럼, 그래픽카드로 라우팅 알고리즘 수행을 위해 패킷 헤더가 전달되어야하기 때문에 IOH와 GPU 사이의 PCIe bus (대역폭: $B_{i,g}$)를 통한 데이터 이동에 대한 시간 비용(T_3)이 추가적으로 빈번하게 발생한다.

2.2. CPU 및 GPU 기반의 멀티코어 라우터 비교

표 1에서 보여주듯이, GPU 기반의 멀티코어 라우터의 경우 CPU보다 가격 당 코어 개수가 더 많으며, 그래픽 카드 내에 있는 메모리를 추가적으로 이용할 수 있다는 장점이 있지만, 수행에 필요한 데이터들이 PCIe bus를 통해서 그래픽 카드로 전송되어야하는 추가적인 비용이 발생한다. 이러한 비용은 처리되는 패킷의 처리량 (Throughput)을 줄일 뿐만 아니라 프로세서 이용률의 최대화, 그리고 증가하는 패킷 양에 대한 병렬 처리 효율 향상을 저해하는 병목 요인이 된다.

	가격 당 코어 개수	가용 메모리	패킷 프로세싱 소요시간
CPU	적음	DRAM	$T_1 + T_2 + T_{cpu}$
GPU	많음	DRAM + 그래픽 카드 메모리	$T_1 + T_2 + T_3 + T_{gpu}$

〔표 1〕 CPU와 GPU 멀티코어 기반 라우터 비교

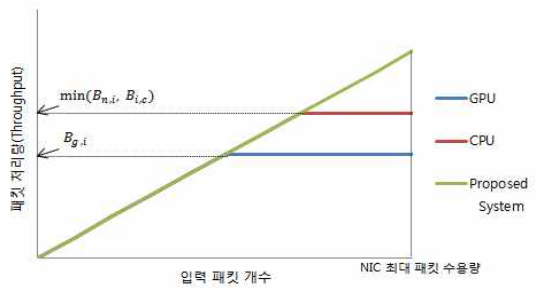
두 시스템의 입력(Incoming) 패킷 개수의 증가에 대한 확장성 반영 정도를 파악하기 위해 입력 패킷 개수와 패킷 처리량의 관계를 분석한다. 라우터의 패킷 처리량은 각 패킷 이동 경로의 대역폭과 코어들의 라우팅 연산 처리량에 비례하는데, CPU 멀티코어 기반 라우터에서는 데이터가 NICs, IOH, CPU, 메모리에서만 이동하는 반면, GPU 멀티코어 기반 라우터에서는 라우팅 연산을 위한 패킷의 메타(Meta) 데이터가 GPU와 IOH 간의 PCIe bus를 통해 이동하는 비용이 추가적으로 고려되어야 한다.

데이터 이동 bus의 대역폭은 제품마다 다르기 때문에 직접적인 비교가 어렵지만 비슷한 가격의 CPU Intel Xeon X5550과 GTX480 그래픽 카드를 기준으로 볼 때, $B_{n,i}$는 4GB/s*NIC 개수(n_{nic}), $B_{i,g}$는 8GB/s, $B_{i,c}$는 25.6GB/s, $B_{c,m}$는 32GB/s, 그리고 $B_{g,m}$은 177.4GB/s의 대역폭을 가진다.

2.3. 멀티코어 시스템 기반 라우터의 발전 방향

앞장에서 분석하였듯이, GPU 기반의 멀티코어 라우터에서는 입력 패킷의 양이 증가하더라도 처리되는 패킷의 양이 한계점에 도달한 후 더 이상 증가하지 않는다. 이것은 주변장치와의 데이터 이동에 이용되는 PCIe bus (대역폭: $B_{i,g}$) 부분에서 병목현상이 발생하기 때문인데, 이는 네 가지 이유 때문인 것으로 추측된다: (1) 작은 대역폭 크기 (2) GPU의 라우팅 연산 시 RAM으로부터 빈번한 데이터 요청 (3) 추가적인 메타 데이터의 이동 (4) 다른 주변장치들과의 대역폭 공유. 현재까지의 연구된 시스템에서 코어의 이용률은 이 부분으로 인해 제약되었고, 따라서 라우터의 성능향상을 위해서는 구조적 해결책이 필요하다.

CPU 기반의 멀티코어 라우터도 한계점이 존재한다. GPU의 경우와 유사하게 주변장치인 NICs와 코어간의 PCIe bus(대역폭: $B_{n,i}$)에 병목현상이 발생하기 때문인데, GPU보다 CPU의 한계점이 더 높은 이유는 일반적으로 여러 NICs를 이용할 수 있어 $B_{i,g}$보다 $B_{n,i}$값이 더 크기 때문이다.



〔그림 2〕 입력패킷 개수 증가에 따른 패킷처리량 예측그래프

결론적으로, 폭증하는 트래픽을 효과적으로 감당하기 위한 새로운 멀티코어 기반의 라우터는 (1) 늘어나는 입력 패킷 개수에 대한 확장성 (Scalability)을 반영할 수 있어야 하며, (2) 주변장치와의 데이터 이동을 최소화하고, (3) 각 코어가 라우팅 알고리즘만을 빠르게 수행할 수 있도록 패킷 입출력 등 패킷 프로세싱에 공통된 작업들을 따로 처리해 줄 수 있는 추가적인 하드웨어를 가지고 있는 형태로 발전해야 한다. CPU 기반, GPU 기반, 그리고 제안한 시스템의 라우터에 대해 입력패킷 개수가 증가할수록 패킷 처리량이 어떻게 될지를 예측한 결과가 그림 2에 나타나 있다. 그림 2에서 볼 수 있듯이, 입력 패킷 개수가 증가하더라도 일정 입력 패킷 개수에 도달하면 패킷처리량이 수렴하게 되지만, 제안된 시스템의 라우터는 이러한 병목 지점을 극복할 수 있어서, 향후 폭증하는 패킷을 효과적으로 처리할 수 있을 것으로 기대된다.

Ⅲ. 결론

본 논문에서는 많은 양의 패킷들을 효율적으로 처리하기 위한 CPU 및 GPU 기반의 멀티코어 라우터의 특징에 대해 비교 및 분석하였고, 성능향상을 제약하는 구조적 한계점을 극복하기 위해 새로운 시스템의 요구사항을 제시하였다. CPU 및 GPU 기반의 시스템 모두 주변장치와의 데이터 교환이 병목이 되기 때문에 입력 패킷 개수에 따른 확장성을 제대로 반영하지 못할 것으로 예측되며, 고성능 라우터의 구현을 위해서는 주변장치와의 통신의 최소화가 요구된다. 이러한 요구사항을 반영하여 최근에는 프로세서 개수의 증가와 동시에 별도의 패킷 입출력 하드웨어를 추가하는 등 위에서 제시된 기존 시스템의 한계를 극복하려는 연구들이 진행되고 있고[4], 본 연구팀에서는 이 시스템을 바탕으로 하는 라우터의 구현 및 성능향상에 대한 연구를 진행하고 있다.

ACKNOWLEDGMENT

본 연구는 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업 (No.C0018176)의 연구수행으로 인한 결과물임을 밝힙니다.

참 고 문 헌

[1] K. Levchenko, G. M. Voelker, R. Paturi, S. Savage. XL:An Efficient Network Routing Algorithm. In ACM SIGCOMM, 2008

[2] M.Dobrescu, N.Egi, K. Argyraki, B.-G. Chun, K.Fall, G.Iannaccone, A.Knies, M.Manesh, and S.Ratnasamy. RouteBricks:Exploiting parallelism to scale software routers. In SOSP, 2009.

[3] S.-J. Han, K. Jang, K.-S. Park, S. Moon. PacketShader:a GPU-Accelerated Software Router. ACM SIGCOMM, 2010.

[4] http://www.tilera.com/technology.