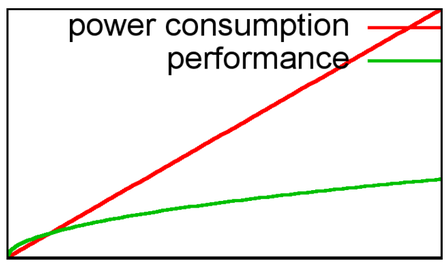
SDN controller on a Many Core Architecture

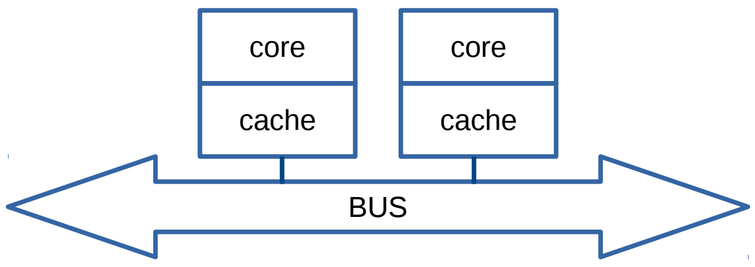
Stephen Mallon
Dr Vincent Gramoli
School of Information Technologies
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

INTRODUCTION

Multi-core Architectures

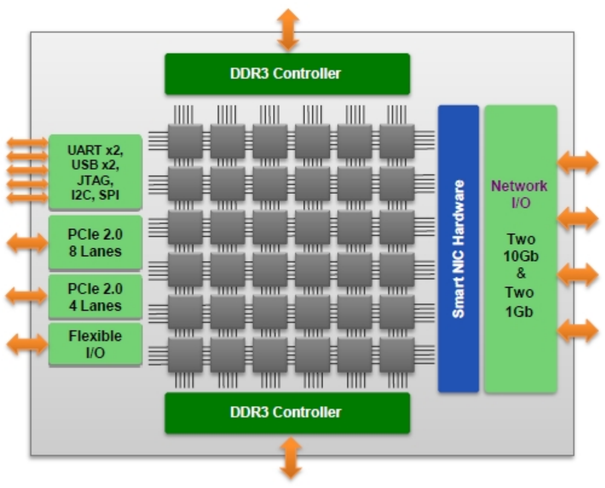
- Small number of complex cores connected by a bus
- Communicate using shared memory and cache coherence
- Pollack's Rule – Performance$\propto\sqrt{}$ Complexity, power consumption linearly proportional
- Diminishing returns for increasing core complexity





Many-core Architectures

- Many simpler cores, connected by a Network-on-Chip (NoC)
- Communicate using message passing
- Each core has lower performance
  promises aggregate core performance to be higher
- Scales to higher core count

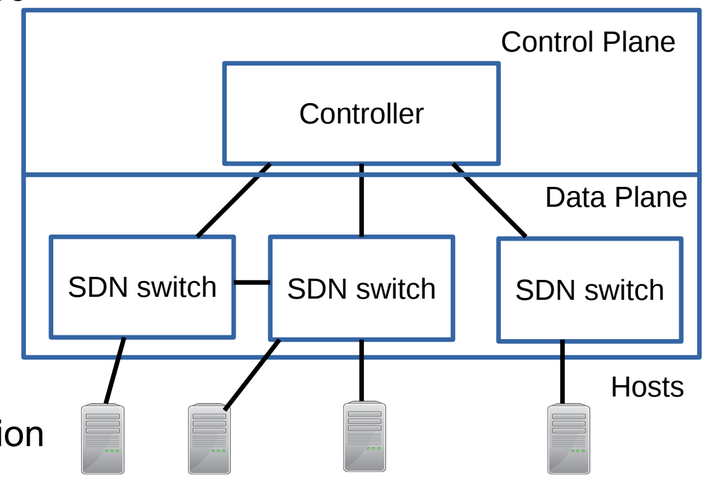


TileGX processor

- Many Core architecture, 36 Cores in 2D mesh, 1.2GHz
- Hardware message passing between cores
- Page level memory control over caching and coherence
- Previous work: Memcached [1]
  67% higher throughput, 3x transactions per watt vs x86

Software Defined Networking (SDN)

- Decouple the control plane (SDN Controllers),
  from the Data plane (SDN switches)
- Controllers make decisions on flows
  push rules to switches
- Switches operate at line rate
- Network can be dynamically adjusted by controllers
  for priority, security or improve link utilisation
- Deployed by Google [2] – 2-3x increase in link utilisation



Our Design

Implementing an efficient SDN Controller

Optimising Memory Accesses

- Multiple private heaps per worker thread
- Fine grained control over caching
- Private data homed on local cache
- Disable caching of output buffers
- Huge Pages to reduce TLB misses
- Zero copy packet buffers - Minimise data copying
- Bind worker memory to closest memory controller

Multi-threaded design

- Single acceptor thread, establishes and verifies new switch connections
- Assign switches to workers on round-robin basis
- Multiple worker threads, pinned to a core
- Subset of cores dedicated to network processing
- Minimise system calls by read/write batching

Evaluation

Benchmarking Setup

- 1 TileGX processor running the SDN Controller
- 2 TileGX processors running cbench benchmark
- Connected by 2x 10 Gb SFP+ cables

Problem Statement

Controller Throughput

- Data centres have thousands of virtual machines
  each making new connections (flows) every second
- SDN Controllers need to have high flow setup rates to handle this
- Many Core Architectures promise higher throughput and energy efficiency
- Evaluate performance of SDN controllers on the TileGX
- Implement a our controller to take advantage of TileGX hardware features
- Benchmark using Cbench [3]
  simulates switches which generate large numbers of flow requests, large numbers of new connections

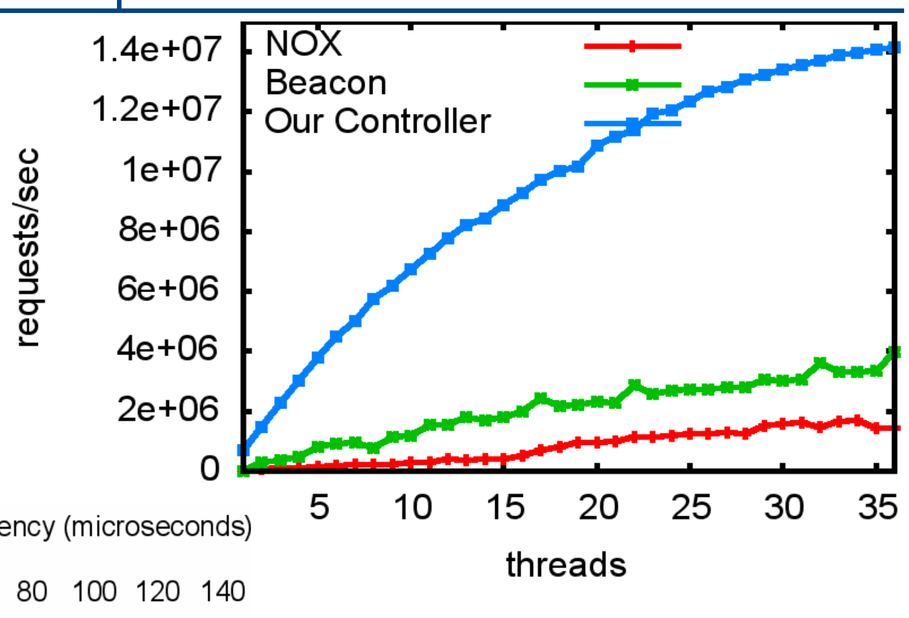
Existing Controllers

- Loop back test on a 16 Core (2x Xeon E5-2670) [4]
- NOX (C++) – 5.3 M requests/sec
- Beacon (Java) – 12.8 M requests/sec

Results

Throughput

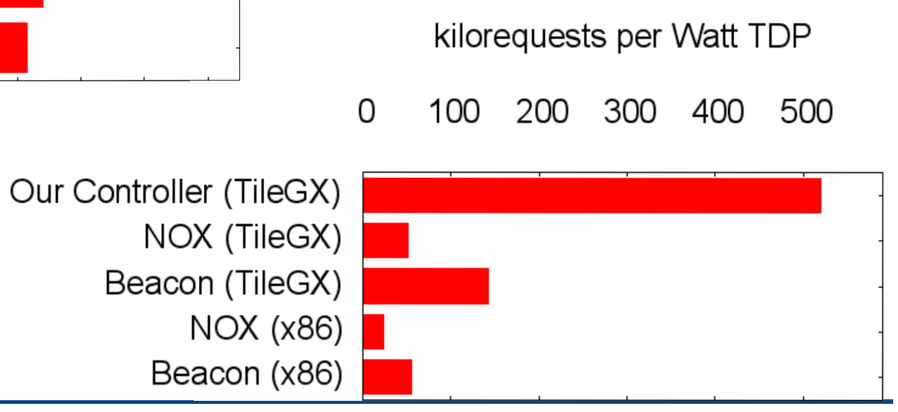
- On TileGX
- NOX – 1.7M req/s
- Beacon – 4M req/s
- Our Controller
  14.6 M req/s



Latency

Energy Efficiency

- Estimate using TDP
- TileGX 28W TDP
- X86 (2 x 115W)
  = 230W TDP



References

- [1] *M. Berezecki et al.*, “Many-core key-value store”. In Green Computing Conference and Workshops (IGCC), 2011
- [2] B4: Experience with a globally-deployed software defined WAN, In Proceedings of the ACM SIGCOMM 2013 conference
- [3] *A. Tootoonchian et al.*, and R. Sherwood. “On controller performance in software-defined networks”. In USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2012
- [4] *D. Erickson*. “The Beacon openflow controller” Proceedings of the 2nd ACM SIGCOMM Workshop on Hot topics in Software Defined Networking, 2013