Evaluating the Performance of Network Protocol Processing on Multi-core Systems

AINA 2009

Matthew Faulkner Andrew Brampton Stephen Pink

Computing Department
Lancaster University
Lancaster

May 25, 2009

- Introduction
  - Terminology
- Methodology
  - How to use Multiple Cores
  - What System is Used for the Evaulation
  - How are the multiple cores used?
  - Experimental Details
- Results
  - Single Connection Results
  - Multiple Connection Results
- Conclusions
  - Discussion and Implications on Related Work
  - Concluding Remarks
- Thank-you

Introduction

This presentation shows a series of experiments that evaluate how many architectures affect network protocol processing. Why?

- Network speeds are increasing, e.g. 10Mbps to 1GbE to 10GbE to 40GbE and 100GbE?
- Advances in clock speeds of microprocessors is slowing down
    - Transistors are just to small!
- Many core systems are being introduced by the microprocessor architecture as the new way to keep with Moorse Law

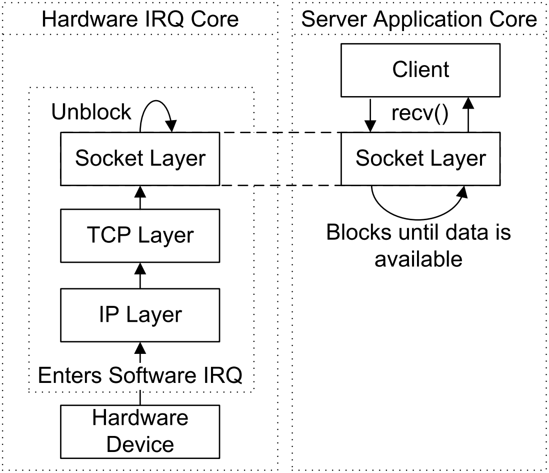
Terminology

Throughout this presentation (and the associated paper) the follow terms are used

- Processor - a single physical entity which includes one or more multiple dies
- Die - an integrated circuit upon which at least one core may be placed alongside other components, e.g. cache, bus interface etc
- Core - a computational unit with the ability to execute instructions.

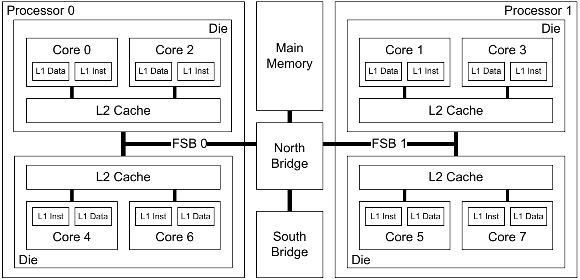
How can multiple cores be used?

Assuming packet level parallelism



What System is Used for the Evaluation

Two of the following (one client and one server)



Access Speeds (ns):

- L1 - 1.5
- L2 - 7.1
- Main memory - 107.3 (sequential) or 187.8 (non-sequential)

How are the multiple cores used in this system?

This paper identifies four possible scenarios

- **Same Core** - Both the network protocol processing and application processing take place on the same core, e.g. core 0
- **Same Die** - Network protocol processing and application processing take place on different cores, albeit they are on the same die, i.e. they share a L2 cache
- **Same Processor** - Network protocol processing and application processing take place on the same processor, i.e. they share the same FSB
- **Same Computer** - Network protocol processing takes place on a core of processor 0 and application processing takes place on a core of processor 1

How are the multiple cores used in this system? (2)

	Shared Resources
Same Core	CPU cycles, both caches, FSB & Northbridge
Same Die	L2 Cache, FSB and Northbridge
Same Processor	FSB and Northbridge
Same Computer	North bridge

Experiential Setups

All experiments

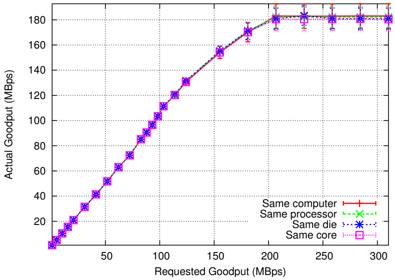
- Used a custom built tool which allowed the application processing affinity to be set
- Had a fixed network protocol processing affinity
- Setup TCP connection(s) between the client and server on a link with a 1500byte MTU
- Sent 1448 byte packets at a specified bit rate
- Recorded different metrics using several different tools (see paper for details)

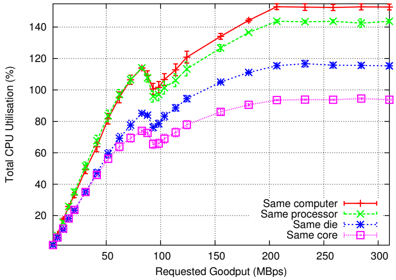
Experiential Setups (2)

However, two different setups were used

- Single connection - Two 1GbE were bonded together to form a single 2GbE link
- Multiple connections - Four 1GbE nics were placed in each machine and a TCP connection established between each pair

Goodput and CPU utilisation for the different scenarios

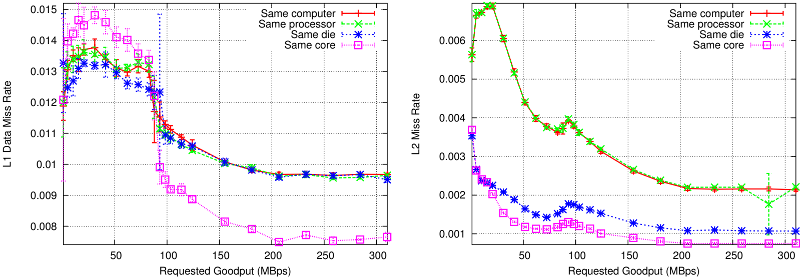




All scenarios achieve the same rate. However, each scenario uses a different amounts of CPU resources.

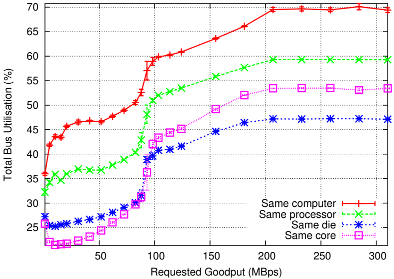
The drop in CPU utilisation at around 75MBps is due to Linux's interrupt mitigation API known as NAPI. See the paper for further details.

CPU Caches



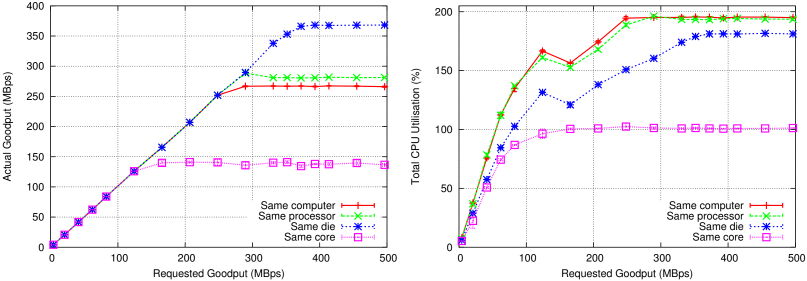
The “Miss Rate” is the number of cache lines fetched in to that level of cache divided by the number of complete instructions. Thus the higher this rate, the more time is spent fetching from a higher level in the cache hierarchy

Front Side Bus



The Northbridge contains a snoop filter which stops adverse traffic going on to the wrong bus

Goodput and CPU utilisation for the different scenarios



Now, the scenarios have a different maximum throughput. Same core is limited by raw CPU cycles.

Multiconnection Experiments

General Comments

- Difference in performance of each scenario can be attributed to (a) caches (b) fsb
- Performance of the Same Core scenario is 15% less in these tests
  - The network buffer could not be emptied fast enough (see the paper for further details)

Some General Comments

- The results can be generalised
  - We have done similar studies using a 10GbE NIC and the hardware architecture has the same impact
- Vastly important area of research especially with the introduction of NUMA many core architectures
- Analysis could be used along with Receive Side Scaling type technologies
- Very useful for Asymmetric Multiprocessors, for example packet processing engines *etc*

Conclusions

- Network protocol processing and application processing are affected by the resources they share
- A performance decrease of over 40% occurs if the wrong cores are used
- When an application only requires small throughput consider using the same core all processing
- When an application requires larger throughput use two cores which share a L2 cache

Network architectures now need to consider microprocessor design

Evaluating the Performance of Network Protocol Processing on Multi-core Systems

AINA 2009

Matthew Faulkner Andrew Brampton Stephen Pink

Computing Department
Lancaster University
Lancaster

May 25, 2009