Comparison and evaluation of High-Performance Computing Network Interconnects used for Lattice QCD Simulations

Konstantin Boyanov
DV Zeuthen
Zeuthen, 18.01.2011
Overview

- Motivation and goals
- QPACE parallel computer
- InfiniBand based clusters
- Micro-benchmarks for performance measurements
- Latency and Bandwidth Models
- Results
- Conclusion and outlook
Lattice Quantum Chromodynamics

- Studies the strong interactions between the building blocks of matter
- Formulation of the theory on a discrete 4D space-time lattice of points
- Mapping to 3D lattice of points
- Quark fields $\Psi(x)$ on lattice points
- Gluons $U(x,\mu)$ on lattice links
- Latice size up to 64x64x64x192 “sites”/points

From [4]
Motivation

- How is the investigation of network performance related to lattice QCD?
- Challenge of typical QCD simulations
  - Use of computation and communication intensive algorithms
  - Parallelisation on a large number number of nodes/cores
- Very good communication hardware needed
  - Low latency of order ~1 microsecond between two nodes
  - High bandwidth of order ~1GB/sec
- To parallelise a problem of given size a strong-scaling architecture is needed
  - Good utilization = good performance pro Euro ratio
Goals

- Investigate possible ways of comparing different platforms for the same application domain
  - QPACE parallel computer
  - Leading commodity cluster technology (Intel CPUs + InfiniBand network)
- To evaluate custom designed network interconnect
  - Discover advantages and disadvantages compared to leading market technologies
- Provide consistent set of tools for low-level performance measurements
  - Discover what affects network operation behaviour
  - Construct models which best describe and predict performance measurements
QPACE Parallel Computer

- QPACE: “QCD Parallel Computer based on Cell Processors”
- Specially developed for QCD numerical simulations
- Based on IBM PowerXCell 8i Processor
- Cell CPUs are interconnected through a custom torus network

From [5]
- QPACE rack = 8 backlanes x 32 node cards
- Liquid cooling makes high performance density possible
- 2 installations with 4 racks each

From [1]
QPACE Network Processor (1)

- Developed by M. Pivanti, F. Schifano, H. Simma
- Implemented on a Field Programmable Gate Array (FPGA)
  - Reprogrammable logic blocks connected by reconfigurable interconnects
  - Ready-to-use circuit modules (Ethernet MAC, PCIe cores, memory)
- QPACE Network Processor is a southbridge with various tasks
  - 2 links to the Cell processor
  - 6 torus network links
  - Nearest neighbour communication
  - On each of the six links up to 8 virtual channels
    - simultaneous use of single physical link by all 8 cores
QPACE Network Processor (2)

IBM interface logic to the Cell processor

FlexIO, 2 byte @ 2.5GHz = 5 GB/s

DCR Master

Master Interface

Slave Interface

128 bit @ 200 MHz

IWC

OWC

Application Logic

UART

To Service Processor

UART

To Root Card

Global Signals

To global signal tree

Torus Network Logic

32 bit @ 250 MHz

PHY 0 PHY 1 PHY 2 PHY 3 PHY 4 PHY 5

6x 4 bit @ 2.5GHz

Ethernet

To Ethernet Transceiver

Network Interconnects for HPC LQCD | Konstantin Boyanov | Technisches Seminar

01/18/11
QPACE Network Processor (3)

- Custom two-sided communication protocol
- Messages composed of multiple 128 byte packets
  - Messages contain multiple packets, max. 2048 bytes (2KB)
- Send operation tnw_put()
- Receive operation tnw_credit()
Off-the-shelf HPC Clusters

- Easy to obtain and start exploitation
  - Processors - Intel-based server CPUs
  - Networking – InfinBand
- Support for
  - Broad spectrum of applications
  - Different communication patterns (MPI-based)
- The performance of Intel-based processors becomes interesting for LQCD simulations with newer and powerful Intel CPUs
- A comparison of the QPACE custom network with the leading InfiniBand network technology is interesting for future developments
**PAX cluster**

- At DESY – the PAX clusters
  - 9 chassis with 16 node cards, each node has two quad core processors
  - Summing up to 32 CPUs / 128 cores per chassis

From [8,9]

Pictures from dell.com
InfiniBand Switched Fabric

- Host Channel Adapters (HCAs) interconnect nodes for data exchange
- On PAX – the ConnectX HCA
  - Dual-port QDR interface, 4 channels per port
InfiniBand Communication

- Queue Pairs = Send Queue + Receive Queue
  - Completion Queues
- Connection Semantics
  - Send/receive
  - Inter-node Direct Memory Access

From [6]
Network Topology

- Different on both architectures
  - 3D torus with directly connected nodes
  - Switched network with intermediate routers
- Crucial when comparing bisectional bandwidth of large partitions (> 512 nodes)
  - Very important for good scaling
- Such large partitions not yet available on PAX
  - Require large and expensive 2\textsuperscript{nd} level switches to interconnect the chassis
  - Still bisectional bandwidth will be limited by bandwidth between 1\textsuperscript{st} and 2\textsuperscript{nd} level switches, no good scaling guaranteed
Network Topology and Bisection bandwidth
Micro-benchmarks

- Used to test the performance of a particular part of a complex computer system
  - Similar operations to real-world applications but not 100% representative
  - Usage of high loads that stress the hardware to its limits
  - Can show behaviour of the system that is unlikely to occur in production runs
    - therefore give interesting results for further development and advancement
- Interesting types of micro benchmarks for network interconnects
  - Measurement of base latency of the network with uni- and bidirectional communication patterns
  - Large amount of consecutive send operations for bandwidth measurements
  - Multi-channel variants of the above
Latency Measurements
Ping-Pong for different message sizes

Client (Sender)
- Recv Sync
- Enter loop
- Credit for receive pong
- Start timer
- Ping
- Wait notify for Pong
- Stop timer
- Exit loop

Server (Receiver)
- Sync
- Wait notify for Sync
- Enter loop
- Credit for receive ping
- Wait notify for Ping
- Pong
- Exit loop
Latency Measurements
Ping-Ping for different message sizes

Client (Sender)  Server (Receiver)

Recv Sync →
Enter loop →
Credit for receive ping →
Credit for receive pong →
Start timer →
Ping →
Wait notify for Ping →
Pong →
Wait notify for Pong →
Stop timer →
Sync →
Wait notify for Sync →
Enter loop →
Credit for receive ping →
Credit for receive pong →
Ping →
Wait notify for Ping →
Pong →
Wait notify for Pong →

Bandwidth Measurements for different message sizes

Client (Sender)  Server (Receiver)

- Recv Sync
- Enter loop
- Credit for receive pong
- Ping 0
- Ping i
- Ping i+1
- Ping N
- Wait notify for Pong

Sync

- Wait notify for Sync
- Enter loop
- Credit for receive ping
- Start timer
- Wait notify for Ping 0
- Wait notify for Ping i
- Wait notify for Ping i+1
- Wait notify for Ping N
- Pong
- Stop timer
Ping-Pong results on QPACE
(Uni-directional Latency)
Latency Model on QPACE

Execution time of the benchmarks described by

\[ t_i(N, N_{VC}) = \lambda_i + N_{VC}\tau_i + N_{VC}N/\beta_i \]

Total execution time given by

\[ T(N, N_{VC}) = \max(t_i) \]

- \( N \) - message size in bytes
- \( N_{vc} \) - number of virtual channels in use
- \( \lambda_i \) - Latency along the hardware datapath
- \( \beta_i \) – Observed bandwidth along the datapath between two nodes
- \( \tau_i \) – Overhead of operations started on the processor
Fit of the model on the data

Ping-Pong

\[ t_i(N, N_{VC}) = \lambda_i + N_{VC} \tau_i + N_{VC} N / \beta_i \]

- \( N = 128-2048 \)
- \( N_{vc} = 8 \)
- \( \lambda_1 = 0.766 \)
- \( \tau_1 = 0.57 \)
- \( \beta_1 = 4084.74 \)
- \( \lambda_2 = 2.76 \)
- \( \tau_2 = 0.037 \)
- \( \beta_2 = 937.8 \)
Ping-Ping results on QPACE
(Bi-directional Latency)
Fit of the model on the data

**Ping-Ping**

\[ t_i(N, N_{VC}) = \lambda_i + N_{VC}\tau_i + N_{VC}N/\beta_i \]

- \( N = 128-2048 \)
- \( N_{vc} = 8 \)
- \( \lambda_1 = 2.423 \)
- \( \tau_1 = 0.326 \)
- \( \beta_1 = 808.6 \)
- \( \lambda_2 = 1.795 \)
- \( \tau_2 = 1.209 \)
- \( \beta_2 = 1303.9 \)
Ping-Bandwidth results on QPACE (blocking)
Ping-Bandwidth results on QPACE (non-blocking)
Conclusion and outlook

- Results show interesting behaviour of the network interconnects
  - Micro-benchmarks show different behaviour for Cell-NWP and TNW communication
  - Communication operations can have relatively large overhead
    - Performance strongly depends on order of operations

- Porting existing micro-benchmarks to the InfiniBand PAX cluster ongoing
- More complex benchmarks to be developed in the near future
  - Running on different topologies and larger node numbers
  - Using lattice QCD application-specific communication patterns
Thank you for your attention!

Questions?
References


General Information and Disclaimer

Konstantin Boyanov

Deutsches Elektronen Synchrotron - DESY (Zeuthen)

Platanenallee 6, 15738 Zeuthen

Tel.:+49(33762)77178

konstantin.boyanov@desy.de

Some images used in this talk are intellectual property of other authors and may not be distributed or reused without their explicit approval!