Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

Jesús Escudero-Sahuquillo
University of Castilla-La Mancha
SPAIN
jesus.escudero@uclm.es
http://www.i3a.info

José Duato
Universtitat Politècnica de València
SPAIN
jduato@disca.upv.es
http://www.gap.upv.es
People involved
Outline

• Introduction
• Topologies: Scalability, Routing and Fault-Tolerance
• Power Efficiency
• Congestion Awareness
• Conclusions
Outline

• Introduction
• Topologies: Scalability, Routing and Fault-Tolerance
• Power Efficiency
• Congestion Awareness
• Conclusions
Introduction

Requirements for Exaflops performance

• “End of the Moore’s Law and Dennard Scaling”: Transistors become smaller while the power density is constant

• Processors to achieve **more computing power** and **less power consumption**
  – Currently **ARM** products offer a good quality per watt
  – Expected **Intel, AMD** or **NVIDIA** power-efficient solutions

• The role of **GPGPUs** in heterogeneous processors

• **Big-Data** demanding more computing power and fast networks

Exascale processors to reduce their peak performance and save power, while HPC systems requiring more processors
# Introduction

## Requirements for Exaflops performance

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2018</th>
<th>Factor Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Pf/s</td>
<td>1 Ef/s</td>
<td>500</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>20 MW</td>
<td>3</td>
</tr>
<tr>
<td>System Memory</td>
<td>0.3 PB</td>
<td>10 PB</td>
<td>33</td>
</tr>
<tr>
<td>Node Performance</td>
<td>0.125 Gf/s</td>
<td>10 Tf/s</td>
<td>80</td>
</tr>
<tr>
<td>Node Memory BW</td>
<td>25 GB/s</td>
<td>400 GB/s</td>
<td>16</td>
</tr>
<tr>
<td>Node Concurrency</td>
<td>12 cpus</td>
<td>1,000 cpus</td>
<td>83</td>
</tr>
<tr>
<td>Interconnect BW</td>
<td>1.5 GB/s</td>
<td>50 GB/s</td>
<td>33</td>
</tr>
<tr>
<td>System Size (nodes)</td>
<td>20 K nodes</td>
<td>1 M nodes</td>
<td>50</td>
</tr>
<tr>
<td>Total Concurrency</td>
<td>225 K</td>
<td>1 B</td>
<td>4,444</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>300 PB</td>
<td>20</td>
</tr>
<tr>
<td>Input/Output bandwidth</td>
<td>0.2 TB/s</td>
<td>20 TB/s</td>
<td>100</td>
</tr>
</tbody>
</table>

Introduction
Deflection with the Roadmap in Power Consumption

<table>
<thead>
<tr>
<th>Green500 Rank</th>
<th>MFLOPS/W</th>
<th>Site*</th>
<th>Computer*</th>
<th>Total Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,499.44</td>
<td>National Institute for Computational Sciences/University of Tennessee</td>
<td>Beacon - Appro GreenBlade GB824M, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P</td>
<td>44.89</td>
</tr>
<tr>
<td>2</td>
<td>2,351.10</td>
<td>King Abdullah City for Science and Technology</td>
<td>SANAM - Adtech ESC4000/FDR G2, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, AMD FirePro S10000</td>
<td>179.15</td>
</tr>
<tr>
<td>3</td>
<td>2,142.77</td>
<td>DOE/SC/Oak Ridge National Laboratory</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x</td>
<td>8,209.00</td>
</tr>
<tr>
<td>4</td>
<td>2,121.71</td>
<td>Swiss Scientific Computer Centre</td>
<td>Cray - Cray XK6, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect</td>
<td>129.00</td>
</tr>
<tr>
<td>5</td>
<td>2,102.12</td>
<td>Forschungszentrum Jülich</td>
<td>IBM - PowerEdge FX2, Custom interconnect</td>
<td>1,970.00</td>
</tr>
<tr>
<td>6</td>
<td>2,101.39</td>
<td>Southern Ontario Smart Computing Innovation Consortium/University of Toronto</td>
<td>BOQdev - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect</td>
<td>41.09</td>
</tr>
<tr>
<td>7</td>
<td>2,101.39</td>
<td>DOE/NNSALLNL</td>
<td>rzuseq - BlueGene/Q, Power BQC 16C 1.60GHz, Custom</td>
<td>41.09</td>
</tr>
<tr>
<td>8</td>
<td>2,101.39</td>
<td>IBM Thomas J. Watson Research Center</td>
<td>BlueGene/Q, Power BQC 16C 1.60 GHz, Custom</td>
<td>41.09</td>
</tr>
<tr>
<td>9</td>
<td>2,101.12</td>
<td>IBM Thomas J. Watson Research Center</td>
<td>BlueGene/Q, Power BQC 16C 1.60 GHz, Custom</td>
<td>82.19</td>
</tr>
<tr>
<td>10</td>
<td>2,101.12</td>
<td>Ecole Polytechnique Federale de Lausanne</td>
<td>CADMOS BG/Q - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect</td>
<td>82.19</td>
</tr>
</tbody>
</table>

* Performance data obtained from publicly available sources including TOP500

1 ExaFLOP = 0.4 GW (400 MW)
# Introduction

Deflection with the Roadmap in Power Consumption

<table>
<thead>
<tr>
<th>Green500 Rank</th>
<th>MFLOPS/W</th>
<th>Site*</th>
<th>Computer*</th>
<th>Total Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,499.44</td>
<td>National Institute for Computational Sciences/University of Tennessee</td>
<td>Beacon - Appro GreenBlade GB824M, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P</td>
<td>44.89</td>
</tr>
<tr>
<td>2</td>
<td>2,351.10</td>
<td>King Abdulaziz City for Science and Technology</td>
<td>SANAM - Adtech ESC4000/FDR G2, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, AMD FirePro S10000</td>
<td>179.15</td>
</tr>
<tr>
<td>3</td>
<td>2,142.77</td>
<td>DOE/SC/Oak Ridge National Laboratory</td>
<td>Titan - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x</td>
<td>8,209.00</td>
</tr>
<tr>
<td>4</td>
<td>2,121.71</td>
<td>Swiss Scientific Computing Center (CSCS)</td>
<td>Todi - Cray XK7, Opteron 6272 16C 2.100GHz, Cray Gemini interconnect, NVIDIA Tesla K20 Kepler</td>
<td>129.00</td>
</tr>
<tr>
<td>5</td>
<td>2,102.12</td>
<td>Forschungszentrum Jülich (FZJ)</td>
<td>JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect</td>
<td>1,970.00</td>
</tr>
<tr>
<td>6</td>
<td>2,101.39</td>
<td>Southern Ontario Supercomputing Innovation Consortium</td>
<td>BGQdev - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect</td>
<td>41.09</td>
</tr>
<tr>
<td>7</td>
<td>2,101.39</td>
<td>DOE/NNSA Alliance (FZJ)</td>
<td>JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect</td>
<td>41.09</td>
</tr>
<tr>
<td>8</td>
<td>2,101.39</td>
<td>IBM Thomas J Watson</td>
<td>IBM Power Systems, Custom Interconnect</td>
<td>41.09</td>
</tr>
<tr>
<td>9</td>
<td>2,101.12</td>
<td>IBM Thomas J Watson</td>
<td>IBM Power Systems, Custom Interconnect</td>
<td>41.09</td>
</tr>
<tr>
<td>10</td>
<td>2,101.12</td>
<td>Ecole Polytechnique</td>
<td>Custom Interconnect</td>
<td>82.19</td>
</tr>
</tbody>
</table>

* Performance data obtained from publicly available sources including TOP500

**TITAN – 1st TOP500**

27 PFLOPS (peak) / 8 MW
At $1M per MW, energy costs are substantial

- 1 petaflop in 2010 uses 3 MW
- 1 exaflop in 2018 at 200 MW with “usual” scaling
- 1 exaflop in 2018 at 20 MW is target
Introduction
Deflection with the Roadmap in Power Consumption

- **Power consumption fraction** of the interconnection network near of 35%
- **Most of the network power consumption** is devoted to the links
- **Depending on the application**, the power consumption can be significantly affected

## Introduction

Interconnection Networks

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2018</th>
<th>Factor Change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System peak</strong></td>
<td>2 Pf/s</td>
<td>1 Ef/s</td>
<td>500</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>6 MW</td>
<td>20 MW</td>
<td>3</td>
</tr>
<tr>
<td><strong>System Memory</strong></td>
<td>0.3 PB</td>
<td>10 PB</td>
<td>33</td>
</tr>
<tr>
<td><strong>Node Performance</strong></td>
<td>0.125 Gf/s</td>
<td>10 Tf/s</td>
<td>80</td>
</tr>
<tr>
<td><strong>Node Memory BW</strong></td>
<td>25 GB/s</td>
<td>400 GB/s</td>
<td>16</td>
</tr>
<tr>
<td><strong>Node Concurrency</strong></td>
<td>12 cpus</td>
<td>1,000 cpus</td>
<td>83</td>
</tr>
<tr>
<td><strong>Interconnect BW</strong></td>
<td>1.5 GB/s</td>
<td>50 GB/s</td>
<td>33</td>
</tr>
<tr>
<td><strong>System Size (nodes)</strong></td>
<td>20 K nodes</td>
<td>1 M nodes</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total Concurrency</strong></td>
<td>225 K</td>
<td>1 B</td>
<td>4,444</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>15 PB</td>
<td>300 PB</td>
<td>20</td>
</tr>
<tr>
<td><strong>Input/Output bandwidth</strong></td>
<td>0.2 TB/s</td>
<td>20 TB/s</td>
<td>100</td>
</tr>
</tbody>
</table>

Introduction
Interconnection Networks

- The Interconnection network is an essential element in current High-Performance Computing (HPC) systems

Thousands of processors (increasing their speed)

Applications demanding more Computing Power and Big-Data functionality

High-Performance is a persevering challenge for the interconnection network

Titan – Cray XK7
560,640 cores (Opteron 6274 16C 2.2GHz)
Cray Gemini Interconnect
1st TOP500 – November 2012
(17.5 PFLOPS)

Beacon Appro GreenBlade GB824M
9,216 cores InfiniBand Interconnect
Custom interconnection network
1st Green500 – November 2012
(2,499.44 MFLOPS/W)
Introduction

Interconnection Networks

Interconnect Family / Systems Share
82.6% of TOP500 supercomputers list is dominated by InfiniBand and Gigabit Ethernet systems.
## Introduction

### Interconnection Networks

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2018</th>
<th>Factor Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Pf/s</td>
<td>1 Ef/s</td>
<td>500</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>20 MW</td>
<td>3</td>
</tr>
<tr>
<td>System Memory</td>
<td>0.3 PB</td>
<td>10 PB</td>
<td>33</td>
</tr>
<tr>
<td>Node Performance</td>
<td>0.125 Gf/s</td>
<td>10 Tf/s</td>
<td>80</td>
</tr>
<tr>
<td>Node Memory BW</td>
<td>25 GB/s</td>
<td>400 GB/s</td>
<td>16</td>
</tr>
<tr>
<td>Node Concurrency</td>
<td>12 cpus</td>
<td>1,000 cpus</td>
<td>83</td>
</tr>
<tr>
<td>Interconnect BW</td>
<td>1.5 GB/s</td>
<td>50 GB/s</td>
<td>33</td>
</tr>
<tr>
<td>System Size (nodes)</td>
<td>20 K nodes</td>
<td>1 M nodes</td>
<td>50</td>
</tr>
<tr>
<td>Total Concurrency</td>
<td>225 K</td>
<td>1 B</td>
<td>4,444</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>300 PB</td>
<td>20</td>
</tr>
<tr>
<td>Input/Output bandwidth</td>
<td>0.2 TB/s</td>
<td>20 TB/s</td>
<td>100</td>
</tr>
</tbody>
</table>

Introduction
Challenges in Exascale Interconnection Networks

- Performance Requirements
- Scalability
- Simplicity
- Reliability
- Fault Tolerance
- Cost and Power Consumption
- Congestion Management

They must not be considered separately, since they are strongly linked.
Outline

• Introduction

• **Topologies: Scalability, Routing and Fault-Tolerance**

• Power Efficiency

• Congestion Awareness

• Conclusions
Topologies
Scaling to 1M of endnodes

• Main objectives:
  – Low latency and high-throughput
  – Reduce the network cost and power consumption

• Design trends in network topologies:
  – Reduce network diameter
  – Reduce the number of components
  – Cost-efficient routing algorithms
  – Increase the path diversity
Topologies
Direct Networks

• **Good Connectivity** as every node has at least one link in each dimension
• **Network Latency** is related to the network diameter
• Efficient **routing algorithms**: DOR, Oblivious, Adaptive, etc.
• **High number of dimensions** increase the switch/routing complexity, the diameter and the probability of contention

Mesh
Iorus
Hypercube
Topologies

Indirect Networks

- **Fat-Trees** are the common alternative for InfiniBand
- High effective bandwidth
- **Cost-efficient routing algorithms** (e.g. DESTRO / D-mod-K)
- **Tradeoff**: high-radix switches (fewer switches but more complex) versus low-radix switches (more switches, simplicity, high cost)
- **Network diameter** depends on the number of stages
Topologies
Indirect Networks

Efficient Deterministic Routing Algorithms

- Tailored to specific network topologies
- Balance the destinations among the different paths
- Offer the same performance as do adaptive routing. They require fewer resources to be implemented
- They solve packet out-of-order delivery problems
- Can be recalculated if some faults appear in the network


Topologies
Indirect Networks

DESTRO example

Balances the use of links by different paths
Topologies
Hierarchical Networks

- **Cost-efficient topologies** for building large topologies
- **Hierarchical network** (3-levels): switch, group and system
- Global links are significantly long
- Network diameter reduction
- High number of links makes them **expensive**
- **Non-minimal routing**

*John Kim, William J. Dally, Steve Scott, Dennis Abts: Technology-Driven, Highly-Scalable Dragonfly Topology. ISCA 2008: 77-88*
Topologies
Hybrid Networks (KNS)

- Designed for large networks
- Based on direct and indirect topologies
- Reduces the diameter, number of switches and links
- High path diversity, which allows a high level of fault-tolerance
- Low latency, high-throughput and lower cost than indirect networks
- Hybrid-DOR routing algorithm
- Implementation in IB is an issue

Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: *A New Family of Hybrid Topologies for Large-Scale Interconnection Networks*. NCA 2012: 220-227
Topologies
Hybrid Networks (KNS)

- Hybrid-DOR example:

Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: A New Family of Hybrid Topologies for Large-Scale Interconnection Networks. NCA 2012: 220-227
Topologies
Hybrid Networks (KNS)

- Hybrid-DOR example:
  - 1st hop (0, 2)

Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: A New Family of Hybrid Topologies for Large-Scale Interconnection Networks. NCA 2012: 220-227
Topologies
Hybrid Networks (KNS)

- Hybrid-DOR example:
  - 1\textsuperscript{st} hop (0,2)
  - 2\textsuperscript{nd} hop (2,14)

- Reduces the network diameter and latency

Roberto Peñaranda, Crispín Gómez Requena, María Engracia Gómez, Pedro López, José Duato: A New Family of Hybrid Topologies for Large-Scale Interconnection Networks. NCA 2012: 220-227
Topologies
Hybrid Networks (KNS)

• Reducing the network components makes cheaper the HPC infrastructure: **larger networks at less cost**

• **Open issues to be solved** by current infrastructure:
  – Is **current technology** able to implement the routers features? (even for 3D, 4D KNS networks)
  – Could the Hybrid-DOR be included in the OFED OpenSM?
  – Fault tolerance and power efficiency
  – Congestion management [1]

*Pedro Yebenes, Jesús Escudero-Sahuquillo, Crispin Gomez-Requena, Pedro Javier García, Francisco J. Quiles and Jose Duato. **BBQ: A Straightforward Queuing Scheme to Reduce HoL-Blocking in High-Performance Hybrid Networks.** Euro-Par 2013 – To be presented*
Topologies and Scalability

Fault Tolerance

- Hybrid topologies offer a high number of alternative paths
- Current techniques (DFSSSP, LASH) could be applied to hybrid topologies with minimal cost

Outline

• Introduction
• Topologies: Scalability, Routing and Fault-Tolerance
• Power Efficiency
• Congestion Awareness
• Conclusions
Power Efficiency

Motivation

• High **cost of the power consumption bill** for large HPC systems: power and cooling

• The **interconnection network power consumption fraction** is about 20% of the total idle power, increasing an additional 20% when simple benchmarks are used [1]

• Some **advances in power consumption for CPUs** and/or memories, but there is a gap to cover in interconnects

• Power Efficiency in HPC interconnect is still a **challenge**:
  – **Idle networks** have a high power consumption
  – **Hw/Sw infrastructure** must offer power efficiency

Power Efficiency

Energy consumption

- Most of the interconnects energy spent by the links
- **Number and length** of the links is important
- **Contention** increases the power consumption
- Current solutions:
  - Hardware
  - Software
Power Efficiency
Software solutions

• **Topology Awareness:**
  – Schedule the traffic so that hot-spots are minimized
  – Maintain the network with low utilization

• **Problems:**
  – Medium term topologies increase the link speed
  – Exascale topologies make the traffic scheduling very difficult
  – Even at low network utilization, the idle power consumed by the links is significant
Power Efficiency

Hardware solutions

- Dynamic Voltage Scaling (DVS)
  - Adds complexity
  - Introduces delay overhead

- Turn off the links completely:
  - Requires a fault tolerant routing algorithm
  - Path diversity is also required
  - Adds complexity
  - Slow reaction to traffic bursts
Power Efficiency

Hardware solutions

• At low traffic loads it makes sense to disable some parts of the network to save power

• **Links aggregation**: Turning on and off dynamically individual links of the same port (w/o disabling it completely):
  
  – Connectivity is not affected
  
  – The routing algorithm is preserved

• When only one link is in operation, its bandwidth is dynamically adjusted as a function of network load

Power Efficiency

Hardware solutions

Power Efficiency
What is the situation in Exascale Interconnects?

• Exascale topologies:
  – Switches with **high number of ports**
  – Ports as **aggregation of links**: high bandwidth, path diversity and reduced latency

• Problems:
  – **Slow reaction** when traffic bursts appear
  – Traffic bursts may lead the system to congestion
Outline

• Introduction
• Topologies: Scalability, Routing and Fault-Tolerance
• Power Efficiency
• Congestion Awareness
• Conclusions
Congestion Awareness

What is exactly the congestion?

- Persistent contention
- Buffers containing blocked packets **fill up**. In this moment congestion appears
Congestion Awareness

What is exactly the congestion?

- **Different congestion trees dynamics** makes more complex the congestion management.

---

Congestion Awareness

What is exactly the congestion?

- **Congestion trees dynamics** may cause Head-of-Line (HoL) blocking, the main negative effect of the congestion.
Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

HPC Advisory Council
Jesús Escudero Sahuquillo and José Duato
June, 16. 2013, Leipzig GERMANY

Congestion Awareness
Low-Order Head-of-Line (HoL) Blocking

Flow Control backpressure

Hot packets (Dst 1)
Cold packets (Dst 2)
Cold packets (Dst 3)

Low-order HoL-blocking
33 % Sending
33 % Stopped
33 % Sending
Congestion Awareness
High-Order Head-of-Line (HoL) Blocking
Congestion Awareness
High-Order Head-of-Line (HoL) Blocking

Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions
HPC Advisory Council
Jesús Escudero Sahuquillo and José Duato
June, 16. 2013, Leipzig GERMANY
Congestion Awareness
Buffer Hogging / Intra-VL hogging

Flow(VirtualLane,OutputPort)

Virtual Lane 2
Free Buffer Slots
Flow (2,1)
Flow (2,2)
Flow (2,3)
Flow (2,4)

Virtual Lane 1
Flow (1,1)
Flow (1,2)
Flow (1,3)
Flow (1,4)

Kenji Yoshigoe: Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues. ICC 2007: 6325-6329
Kenji Yoshigoe: Threshold-based Exhaustive Round-Robin for the CICQ Switch with Virtual Crosspoint Queues. ICC 2007: 6325-6329
Congestion Awareness
Why is congestion management necessary?

- Exascale networks: around one million of endnodes
- **Scalability** → Less available resources in the network
- **Power efficiency** policies react slowly to traffic bursts
- Traffic loads to reach the limits of saturation
- **Congestion effects** are noticed very quickly
Congestion Awareness

Why is congestion management necessary?

At saturation, network performance drops dramatically due to the effects of congestion situations.
Congestion Awareness

The Big Picture

- Growing concurrency at endnodes
- Processor prices drop (demand)
- Congestion probability grows
- Growing link speed
- Relative interconnect cost increases
- Network power consumption fraction increases
- Power management
- Smaller networks
- Saturation point reached with lower traffic load
- Overall effective bandwidth decreases

Congestion Management Strategies
Congestion Awareness

Proposed Solutions

- **Overdimensioning & Packet Dropping**: Not applicable
- **Proactive Strategies**: network status knowledge is not always available
- **Reactive Strategies**: Delay between congestion detection and notification and low scalability in large networks

All them have problems that may lead to a performance degradation (Head-of-Line blocking effect)
Congestion Awareness
Our philosophy

The real problem is not the congestion itself, but its negative effects (HoL-blocking and Buffer Hogging).

By preventing HoL-blocking and Buffer Hogging, congestion becomes harmless.

And it does not matter the time spent to throttle or remove it.
<table>
<thead>
<tr>
<th>Techniques</th>
<th>Low-order prevention</th>
<th>High-order Prevention</th>
<th>Scalable</th>
<th>Additional Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOQnet</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>VOQsw/OBQ A/BBQ</td>
<td>Yes</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DAMQs</td>
<td>Yes</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DBBM</td>
<td>Partial</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DSBM</td>
<td>Partial</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Virtual Channels</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>(VOQsw)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECN-like</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

HPC Advisory Council                           Jesús Escudero Sahuquillo and José Duato                       June, 16. 2013, Leipzig GERMANY

Congestion Awareness
Proposed Solutions

Static Mapping of Hot-Flows to queues (or VLs)

Dynamic Mapping of Hot-Flows to queues (or VLs)

Injection Throttling

Combining Dynamic Hot-Flow Isolation and Injection Throttling

Simplicity
(Optimal efficiency with minimum complexity)

Use of Additional Resources

Effectiveness
(Complete HoL-blocking prevention)
Congestion Awareness

Static mapping of hot flows to queues (or VLs)

- Significantly avoids HoL-blocking and buffer-hogging
- Tailored to specific topologies and routing algorithms
- The queue assignment criterion exploits the properties of both network topology and routing scheme (topology-aware)
- They achieve the same performance as other proposals but requiring the half (or even a quarter) their resources
- Prevent HoL-blocking and buffer-hogging as much as possible with the available network resources
- We need to pay an “extra-price” for complete effectiveness

Non-aggressive Congestion Control
Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Hot packets (Dst 1)
Cold packets (Dst 2)
Cold packets (Dst 3)

Src. 0 → Sw. 1 → Sw. 4 → Sw. 7 → Dst. 1
Src. 1 → Sw. 2 → Sw. 4 → Sw. 7 → Dst. 2
Src. 2 → Sw. 2 → Sw. 5 → Sw. 8 → Dst. 2
Src. 3 → Sw. 3 → Sw. 6 → Sw. 8 → Dst. 3
Src. 4

High- and Low-order HoL-blocking avoidance

Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Network Latency vs Normalized Generated Traffic

4-ary 5-tree
8x8 switches

8-ary 3-tree
16x16 switches
## Congestion Awareness

Static mapping of hot flows to queues (or VLs)

<table>
<thead>
<tr>
<th>#</th>
<th>Queue Scheme</th>
<th># Queues per port</th>
<th>Input Port RAM Size</th>
<th>RAM area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VOQnet (1024 BMIN)</td>
<td>1024</td>
<td>4,194,304 bytes</td>
<td>45.78821</td>
</tr>
<tr>
<td>2</td>
<td>VOQnet (256 BMIN)</td>
<td>256</td>
<td>1,048,576 bytes</td>
<td>11.28886</td>
</tr>
<tr>
<td>3</td>
<td>VOQnet (64 BMIN)</td>
<td>64</td>
<td>262,144 bytes</td>
<td>2.66470</td>
</tr>
<tr>
<td>4</td>
<td>VOQsw (32-port switch)</td>
<td>32</td>
<td>131,072 bytes</td>
<td>1.15024</td>
</tr>
<tr>
<td>5</td>
<td>VOQsw (16-port switch)</td>
<td>16</td>
<td>65,536 bytes</td>
<td>0.62292</td>
</tr>
<tr>
<td>6</td>
<td>VOQsw / IB-ITH (8-port switch)</td>
<td>8</td>
<td>32,768 bytes</td>
<td>0.36549</td>
</tr>
<tr>
<td>7</td>
<td>DBBM/OBQA/OBDA</td>
<td>8</td>
<td>32,768 bytes</td>
<td>0.36549</td>
</tr>
<tr>
<td>8</td>
<td>DBBM/OBQA/OBDA</td>
<td>4</td>
<td>16,384 bytes</td>
<td>0.15142</td>
</tr>
</tbody>
</table>
Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Commercial solutions

- Easily **adaptable to real systems** by solving some issues
- **Commercial switches have “native” features** to deal with problems derived from congestion:
  - **Several read-ports per input buffer**: they solve low-order HoL-Blocking but not high-order HoL-Blocking
  - **Virtual Lanes**: they limit “inter-VL” buffer-hogging but not “intra-VL” buffer-hogging
Congestion Awareness
Static mapping of hot flows to queues (or VLs)

vFtree – A commercial solution for FTs

Distributes the traffic flows among all the available VLs for a given link

Implemented in OpenSM

Wei Lin Guay, Bartosz Bogdanski, Sven-Arne Reinemo, Olav Lysne, Tor Skeie: vFtree - A Fat-Tree Routing Algorithm Using Virtual Lanes to Alleviate Congestion. IPDPS 2011: 197-208
## Congestion Awareness

Static mapping of hot flows to queues (or VLs)

vFtree – A commercial solution for FTs

<table>
<thead>
<tr>
<th>HPCC test</th>
<th>Ftree (1 VL)</th>
<th>vFtree (3 VLs)</th>
<th>Improvement in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. ping pong latency (ms)</td>
<td>0.002116</td>
<td>0.002116</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg. ping pong latency (ms)</td>
<td>0.022898</td>
<td>0.013477</td>
<td>41.14</td>
</tr>
<tr>
<td>Min. ping pong latency (ms)</td>
<td>0.050500</td>
<td>0.043005</td>
<td>14.84</td>
</tr>
<tr>
<td>Naturally ordered ring latency (ms)</td>
<td>0.021791</td>
<td>0.014591</td>
<td>33.04</td>
</tr>
<tr>
<td>Randomly ordered ring latency (ms)</td>
<td>0.024262</td>
<td>0.015826</td>
<td>34.77</td>
</tr>
<tr>
<td>Max. ping pong bandwidth (MB/s)</td>
<td>1593.127</td>
<td>1594.338</td>
<td>0.07</td>
</tr>
<tr>
<td>Avg. ping pong bandwidth (MB/s)</td>
<td>573.993</td>
<td>830.909</td>
<td>44.75</td>
</tr>
<tr>
<td>Min. ping pong bandwidth (MB/s)</td>
<td>94.868</td>
<td>345.993</td>
<td>264.71</td>
</tr>
<tr>
<td>Naturally ordered ring bandwidth (MB/s)</td>
<td>388.969246</td>
<td>454.236253</td>
<td>16.78</td>
</tr>
<tr>
<td>Randomly ordered ring bandwidth (MB/s)</td>
<td>331.847978</td>
<td>438.604531</td>
<td>32.17</td>
</tr>
</tbody>
</table>

*Wei Lin Guay, Bartosz Bogdanski, Sven-Arne Reinemo, Olav Lysne, Tor Skeie: vFtree - A Fat-Tree Routing Algorithm Using Virtual Lanes to Alleviate Congestion. IPDPS 2011: 197-208*
Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

HPC Advisory Council
Jesús Escudero Sahuquillo and José Duato
June, 16. 2013, Leipzig GERMANY

Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Adapting vFtree to Exascale FTs

Shuffling of VLs among “consecutive” flows

Intra-VL buffer hogging (3rd stage)
Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Adapting vFtree to Exascale FTs

Intra-VL buffer hogging in different VLs
Congestion Awareness
Static mapping of hot flows to queues (or VLs)

Adapting vFtree to Exascale FTs

- Metrics to achieve a perfect mapping of traffic flows to SLs (i.e. to VLs):
  - **VL Load**: Number of DLIDs (depends on R)
  - **Balancing Degree**: Average number of DLIDs per VLs (depends on the routing algorithm)
  - **Overlapping Degree**: Measures the DLIDs in several VLs (must be zero to reduce the intra-VL hogging probability)
Congestion Awareness

Dynamic mapping of hot flows to queues

- Suitable for **lossless** Interconnection Networks using distributed routing
- Apply the RECN philosophy to distributed routing networks
- Congestion roots are detected in whatever network point
- Hot packets are isolated in special queues at each port, thereby preventing them from causing HoL-blocking
- Quick reaction against congestion
- **Limited resources.** When the number of congestion trees in a port exceeds the number of queues for storing hot-flows
Congestion Awareness
Dynamic mapping of hot flows to queues

RECN – The first successful approach

- Efficiently *prevents* HoL-blocking in *source-routing* networks
- Detects *congestion* in whatever network point
- Keeps track of congestion roots, using *Content-Addressable Memories* (CAMs) which *store congestion information*
- Isolates *hot packets* in special queues
- Propagates *congestion information* in a regional way throughout the affected switches, so that those switches use that information to isolate hot packets as well
Congestion Awareness
Dynamic mapping of hot flows to queues

DRBCM–The solution for Distributed Routing

- **Effective HoL-blocking elimination** in networks with **distributed routing** (using the RECN philosophy)
- **Congestion roots identification** by detecting “hot-flows” addressed to them only using the destinations IDs information
- “**Hot packets**” (i.e. packets belonging to hot flows involved in a congestion situation) are **isolated in special queues**
- **Congestion information** is based on destinations instead of turnpools, and it represents any network congested point

*New CAM structure, new detection, propagation and resource management policies*
Congestion Awareness
Dynamic mapping of hot flows to queues

DRBCM—The solution for Distributed Routing

- Cold Packets Queue (CPQ)
- Hot Packets Queues (HPQ)

\[ \text{To Isolate} \text{ hot flows from cold ones} \]
Congestion Awareness

Dynamic mapping of hot flows to queues

DRBCM–The solution for Distributed Routing

- The **mask field** (using bits with values 0, 1 and X) identifies all the destinations crossing a congestion root

- The **remainder fields are similar** to those of the CAMs

*Jesus Escudero-Sahuquillo, Pedro J. Garcia, Francisco J. Quiles, Jose Flich, Jose Duato, An Effective and Feasible Congestion Management Technique for High-Performance MINs with Tag-Based Distributed Routing, IEEE Transactions on Parallel and Distributed Systems, 14 Nov. 2012.*
Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

Jesús Escudero Sahuquillo and José Duato

June, 16. 2013, Leipzig GERMANY

Congestion Awareness
Dynamic mapping of hot flows to queues

DRBCM–The solution for Distributed Routing

DrBCM - The solution for Distributed Routing

- Congestion Awareness
- Dynamic mapping of hot flows to queues

**DrBCM**

- The solution for Distributed Routing

**Sw. 0 - Stage 0**
- 0

**Sw. 16 – Stage 1**
- 16, 32, 48

**Sw. 32 – Stage 2**
- 1, 32

<table>
<thead>
<tr>
<th>#</th>
<th>Root</th>
<th>Mask</th>
<th>Hops</th>
<th>oPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rz</td>
<td>010000</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Ry</td>
<td>xx0000</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Rx</td>
<td>xxxx00</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th>Root</th>
<th>Mask</th>
<th>Hops</th>
<th>oPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rz</td>
<td>010000</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Ry</td>
<td>xx00xx</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Jesus Escudero-Sahuquillo, Pedro J. Garcia, Francisco J. Quiles, Jose Flich, Jose Duato, *An Effective and Feasible Congestion Management Technique for High-Performance MINs with Tag-Based Distributed Routing*, IEEE Transactions on Parallel and Distributed Systems, 14 Nov. 2012.
Congestion Awareness
Dynamic mapping of hot flows to queues

Experiments results

4-ary 5-tree
1024 nodes
(traffic with 4 hot-spots)

4-ary 4-tree
256 nodes
(Real traffic)
## Congestion Awareness

Dynamic mapping of hot flows to queues

### RAM Memory and Area comparison

<table>
<thead>
<tr>
<th>#</th>
<th>Queue Scheme</th>
<th># Queues per port</th>
<th>Input Port RAM Size</th>
<th>RAM area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VOQnet (1024 BMIN)</td>
<td>1024</td>
<td>4,194,304 bytes</td>
<td>45.78821</td>
</tr>
<tr>
<td>2</td>
<td>VOQnet (256 BMIN)</td>
<td>256</td>
<td>1,048,576 bytes</td>
<td>11.28886</td>
</tr>
<tr>
<td>3</td>
<td>VOQnet (64 BMIN)</td>
<td>64</td>
<td>262,144 bytes</td>
<td>2.66470</td>
</tr>
<tr>
<td>4</td>
<td>VOQsw (32-port switch)</td>
<td>32</td>
<td>131,072 bytes</td>
<td>1.15024</td>
</tr>
<tr>
<td>5</td>
<td>VOQsw (16-port switch)</td>
<td>16</td>
<td>65,536 bytes</td>
<td>0.62292</td>
</tr>
<tr>
<td>6</td>
<td>VOQsw / IB-ITh (8-port switch)</td>
<td>8</td>
<td>32,768 bytes</td>
<td>0.36549</td>
</tr>
<tr>
<td>7</td>
<td>DBBM/OBQA/vFtree</td>
<td>4</td>
<td>16,384 bytes</td>
<td>0.15142</td>
</tr>
<tr>
<td>8</td>
<td>DRBCM</td>
<td>5</td>
<td>20,480 bytes</td>
<td>0.26548</td>
</tr>
</tbody>
</table>
Congestion Awareness

Combining the best of two worlds

- RECN-like approaches:
  - Additional, limited resources limit the number of flows to isolate
  - Unfairness in the hot flows scheduling

- Injection Throttling Strategies (e.g. Infiniband CC):
  - Slow reaction to congestion
  - Traffic oscillations (saw-shape effect)
Congestion Awareness
Combining the best of two worlds

• Combination of two of the most successful approaches for congestion control

• Jointly developed by UCLM-UPV (Spain) and Simula Research Laboratory (Norway)

• Our “combined” technique offers the best of two worlds, eliminating their respective problems

Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks. ICPP 2011: 662-672
Congestion Awareness
Combining the best of two worlds

- Input ports like RECN (CAMs at input/output ports)
- HPQs assigned when the CPQ exceeds a threshold
- Output ports in congestion state, when HPQ reaches a High Threshold
- Packets are marked (FECN) at output ports in congestion state
- Output ports congestion state are deactivated when a HPQ reaches the Low Threshold, and there are no HPQs exceeding that threshold

**Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks. ICPP 2011: 662-672**
Congestion Awareness
Combining the best of two worlds

- An IA receives a BECN and applies an Injection Rate Delay (IRD) value
- \( \text{IRD} = \text{CCT}[\text{CCTI}[\text{destination}]] \)
- \( \text{CCTI}[\text{destination}] \) may be increased if more BECNs are received at the IA
- A Timer[destination] is set on for a new calculated \( \text{CCTI}[\text{destination}] \)
- When Timer[destination] expires, \( \text{CCTI}[\text{destination}] \) is decreased in one unit, then the injection is slowly restored (if no more BECNs arrive).
- HPQs prevents HoL-Blocking at IAs

---

Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks. ICPP 2011: 662-672
Congestion Awareness
Combining the best of two worlds

Experiment results

**4-ary 4-tree (256 nodes)**

*4 congestion trees*

**8-ary 3-tree (512 nodes)**

*4 congestion trees*

Jesús Escudero-Sahuquillo, Ernst Gunnar Gran, Pedro Javier García, Jose Flich, Tor Skeie, Olav Lysne, Francisco J. Quiles, José Duato: *Combining Congested-Flow Isolation and Injection Throttling in HPC Interconnection Networks*. ICPP 2011: 662-672
Outline

- Introduction
- Topologies: Scalability, Routing and Fault-Tolerance
- Power Consumption
- Congestion Awareness
- Conclusions
Conclusions

• HPC applications (MPI, Big-Data, etc.) demanding more computing power

• Processors are reducing their peak performance to reduce power consumption:
  – New materials could improve the level of integration

• Endnodes will interconnect a thousand of processors

• Network interfaces will increase their link speed

• Networks of Exascale HPC Systems will interconnect around 1 million of endnodes
Conclusions

- HPC Interconnects challenges:
  - **Low latency** and **high-throughput** → The typical one
  - **Scalability** → Cost-efficiency and path diversity
  - **Reduce** the network **power consumption** fraction
  - React until **congestion** situations:
    - Non-aggressive congestion control: Optimal efficiency with the available resources
    - Aggressive congestion-control (Extra-price): Complete effectiveness with additional resources
Questions???
Keynote

Efficient and Cost-Effective Interconnection Networks in the Road to Exascale HPC: Challenges and Solutions

**Jesús Escudero-Sahuquillo**
University of Castilla-La Mancha
SPAIN
jesus.escudero@uclm.es
http://www.i3a.info

**José Duato**
Universtitat Politècnica de València
SPAIN
jduato@disca.upv.es
http://www.gap.upv.es