SDCI–Net: Collaborative Research: NCAR year–2 review (OCI–1127341)

John Dennis (dennis@ucar.edu)
Zhengyang Liu (zl43f@virginia.edu)
Fabrice Mizero (mizero.fabrice@philander.edu)

Collaborators / Personal
- Malathi Veeraraghavan (University of Virginia)
- Robert Russell (University of New Hampshire)
- John Dennis (NCAR)
- Zhengyang Liu (University of Virginia, NCAR)
- Patrick MacArthur (University of New Hampshire)
- Fabrice Mizero (Philander Smith College, NCAR)
- Jesus Labarta (Polytechnic University of Catalonia, Barcelona Supercomputing Center)
- Judit Gimenez (Barcelona Supercomputing Center)
- Harald Servat (Polytechnic University of Catalonia)
- Srinath Vadlamani (NCAR)
Motivation

- Application performance variability – CESM

Execution Time for ASD on Yellowstone

Large Variability in execution time

CAM Scalasca Analysis
Potential ideas on why applications are running slow on Yellowstone

- Bad links -> reduced bandwidth [Yes, somewhat]
  - Discovered a link in network at FDR10
  - FDR10: 40 Gb/s
  - FDR: 56 Gb/s
  - 28% slower! => 6x larger MPI_Wait time
- Bad links -> routing table recalculation [Yes, likely]
- OS jitter on Nodes
  - Transparent Huge Pages [Yes]
  - Timer interrupt frequency [Yes]
- Congestion in Network [Maybe]

BSC performance analysis tools

- Developed at:
  - Barcelona Supercomputer Center (BSC)
  - Polytechnic University of Catalonia (UPC)
- extrae: trace collection
  - Enables very detailed tracing of application characteristics
  - Creates a performance database
- paraver: visualization client
- Dimemas: trace replay tool
  - Apply ‘what-if’
  - Currently network model is basic: latency + bandwidth
Extrae/Paraver analysis

- Collect traces using Extrae
- Perform visual inspection using Paraver
- Perform quantitative analysis using R (LIU presentation)
- Look at Higher Order Methods Modeling Environment (HOMME) on Yellowstone
  - Atmospheric dynamical core used in CESM
  - 96 cores/6 nodes

HOMME: Useful duration

- Well synchronized
- Zoom in!
- MPI time is black
HOMME: Useful duration (con’t)

HOMME: message passing
HOMME: message passing (con’t)

Fabricé Mizero: Evaluating the Impact of Infiniband Routing Algorithms on Network Performance

- Philander Smith College,
- Computer Science Junior
- SiParCS Intern, 2013
- Mentor:
  - Dr. John Dennis, NCAR
- Collaborators:
  - Prof. Malathi Veeraraghavan, UVA
  - Zhengyang Liu, UVA
  - Dr. Robert D. Russell, UNH
  - Patrick MacArthur, UNH
Subnet Management in Infiniband Networks

- **Subnet Manager**
  - Infiniband compliant subnet manager – OpenSM
  - Tasks:

    1. Initialize Infiniband Hardware
    2. Local Identifiers Assignment
    3. Routing Table Calculations & Distributions

    (reassign lids) → r

    If found

    Regularly Sweeps for changes in the Topology

Routing Recalculation is a huge task in Large Scale Networks

---

**Topo-file Example in Use**

32 nodes, 3 levels, full symmetrical FatTree
Infiniband Routing On a Healthy Subnet
Destination-Based Routing & Credit Based Flow Control

Subnet Manager Adaptation to Link Failures

OpenSM scheduled Sweeps

Link Failure Detected

Find Directly Affected Switches

Update Routing Tables in both Switches

Subnet UP
Experiments

- Tools:
  - Infiniband Management Simulator (IBMgtSim)
  - Subnet Manager (OpenSM)

- Opensm Logs: Calculate subnet recovery times.

Subnet Recovery Times-408 Sim Runs (UpDn Scatter-Ports -A)
Future Work

- **Cost of routing table recalculation**
  - How does this scale with network size?
  - Cost of partial routing table update.
- **Understand network contention issues**
  - Determine self interference
  - Estimate interference from other network traffic
  - Impact of network topology
- **Minimize OS jitter**
  - Eliminate THP
  - Reduce clock interrupt frequency
  - Other non-network sources of de-synchronization
- **Understand MPI stack versus hardware overhead**
- **Interface Dimemas & OMNet++**

---

Useful duration

Extrae trace of HOMME (ne=3) on KNC using 54 MPI tasks:

Timeline trace:
- x-axis is time
- y-axis is first 8 MPI tasks
- Color indicates user computational bursts
- Black indicates MPI time
- Most time spent in computational bursts of duration: 260, 400 usec
Extrace trace of HOMME (ne=3) on KNC using 54 MPI tasks:

Timeline trace:
- x-axis is time
- y-axis is first 8 MPI tasks
- Color indicates time in MPI calls
- Lines indicate message that was passed
- Black indicates useful duration
- Was late sender caused by preemption of MPI task 7?

Histogram of useful duration

Each row corresponds to a different thread
- y-axis is duration of computational bursts
- Blue corresponds to a large number of events with a particular duration
- 260 usec events
- 400 usec events
- Infrequent long latency Events (2–4 per core)