RUN-TO-RUN VARIABILITY ON THETA

SUDHEER CHUNDURI, KEVIN HARMS, SCOTT PARKER, VITALI MOROZOV, KALYAN KUMARAN

Performance Engineering Group
Argonne Leadership Computing Facility

July 11, 2017
VARIABILITY

- Desirable requirement: Variability in production mode < 7%
- Noticed much higher variability than the desired.

Challenges
- Less reliable performance measures (multiple repetitions with statistical significance analysis is required)
- Performance tuning – quantifying the impact of a code change is difficult
- Difficult to predict job duration
  - Less user productivity
  - Inefficient system utilization
  - Complicates job scheduling
OUTLINE

- Theta System Description
- Variability
  - Identify sources of variability
  - Characterize network level variability
    - Micro-benchmarks
    - Applications
- Metric used to quantify – Max to min range
- Network level variability is still unsolved
  - We hope CODES can help us explore solutions
  - We just started exploring CODES
THETA

- **System:**
  - Cray XC40 system
  - 3,624 compute nodes / 231,936 cores
  - 9.65 PetaFlops peak performance

- **Processor:**
  - Intel Xeon Phi, 2nd Generation (Knights Landing) 7230
  - 64 Cores
  - 1.3 GHz base / 1.1 GHz AVX / 1.4-1.5 GHz Turbo

- **Memory:**
  - 16 GB MCDRAM per node (Cache mode/Flat mode)
  - 192 GB DDR4-2400 per node

- **Network:**
  - Cray Aries interconnect
  - Dragonfly network topology
VARIABILITY ON THETA

- Variability between runs is frequently 15% or greater, can be up to 100%
- Identified 4 causes of potential variability
  - Core level variability due to OS noise
    - Impact on applications: minimal
    - Available mitigations: Use core spec (mechanism to reduce OS noise), exclude tile 0 & 32
  - Tile level variability due to shared resource contention on tile (L2)
    - Impact on applications: yes
    - Available mitigations: run using only 1 core per tile
  - Memory mode variability due to cache mode page conflicts
    - Impact on applications: yes
    - Available mitigations: run in flat mode
    - Potential mitigations: improved zone sort
  - Network variability due to shared network resources
    - Impact on applications: yes
    - Available mitigations: run without other jobs present on system
    - Potential mitigations: compact job placement (?)
ARIES DRAGONFLY NETWORK

**Aries Router:**
- 4 Nodes connect to an Aries
- 4 NIC’s connected via PCIe
- 40 Network tiles/links
- 4.7-5.25 GB/s/dir per link

**Connections within a group:**
- 2 Local all-to-all dimensions
  - 16 all-to-all horizontal
  - 6 all-to-all vertical
- 384 nodes in local group

**Connectivity between groups:**
- Each group connected to every other group
- Restricted bandwidth between groups

Figure credit: Cray
**NETWORK LEVEL VARIABILITY**

- Cray XC Dragonfly topology
  - Chances for link sharing are high

- Sources of variability
  - Inter-job contention
    - Size of the job
    - Node placement
    - Workload characteristics
    - Co-located job mix

- Isolated system run
  - Allocated the whole system ourselves
    - Use only 256 nodes
  - less than 1% variability from
    - Run-to-run
    - Job-to-job
    - Refer this as 'Ideal'

256 nodes Allreduce 8MB
MPI BENCHMARKS

- MPI Collective Benchmarks
- Repeated 100 times with in a job
- Measured across several days
  - Node placement change
  - Job mix change
- Ignoring the few outliers, the variability range is within 15%.
  - one job around 70% higher
- Ideal - performance in isolated system.
- Each box show the median, IQR and the outliers

128 nodes Allreduce 8MB 1 PPN
MPI BENCHMARKS

128–Allreduce–64–1048576

variable
- Ideal
- 02–16–17
- 02–17–04
- 02–17–13
- 02–17–17
- 02–18–15
- 02–20–03
- 02–21–02
- 02–21–17
- 02–22–15
- 02–23–17
- 02–24–21
- 02–25–17
- 02–26–17
- 02–27–17
- 03–02–04
- 02–15–20

name
- −10%
- −5%
- +5%
- +10%
- MoM

Latency(us)

FOUR 256 node jobs on Edison Allreduce 1K 1 ppn

- static routing mode does not seem to help reduce the variability

Around 20% job-to-job

128 nodes Allreduce 8MB 64 PPN
APPLICATIONS

- How much is the Variability range with real application runs?
- The runs were performed by applying the mitigations described until now
  - Core spec (use –r 1 with aprun to assign one hw thread to OS)
  - Leave out one core for OS
  - Only Flat memory mode is used

- Nekbone
  - Nekbone mini-app derived from Nek5000
  - Streaming kernels – BW limited
  - Matrix multiply – Compute limited
  - Communication – Communication limited

- Variability in Total time correlates with variability in the COMM time
- Job-to-job variability: 32% on 128 node job on Theta
  21% on 64 node job on CORI

128 nodes on Theta
64 nodes on CORI
NEKBONE VARIABILITY DUE TO NETWORK

5 repeats with in a job
- runs with the same node allocation in a job

256 nodes on Theta

Nothing peculiar with the node placement for the job 7

256 nodes on Theta
w/o the Outlier
MILC

128 nodes on Theta

256 nodes on Theta

All the Theta results were performed on a 512 contiguous node partition that house Flat memory mode nodes

Scheduling policy changed recently when Theta went onto production last week Possibly, more variability expected.

128 nodes  - 74%
256 nodes  - 45%

Higher the time has a corresponding higher time in the communication (MPI) part – Cray PAT MPI profiling

(number of repetitions selected here is just random – they are not statistical significant)
We don’t yet have a concrete approach to contain the network level variability other than running in an isolated mode
- Node placement does not have strong correlation
- Routing modes – while static routing modes seem to contain the extent of variability – it appears performance would be impacted
- Haven’t explored the job mix study

We intend to explore CODES to potentially find ways to contain, if not totally remove, variability:
- Simulate the variable network performance
- Determine the source(s) of network level variability
- Experiment with methods to contain/control it and then depending on the results, try them on Theta
QUESTIONS?
VARIABILITY DUE TO MEMORY MODES

- The MCDRAM cache is a **direct-mapped cache**.
- Possible conflicts for apps with > 16GB working sets.
- Conflicts possible with < 16 GB working set
- Memory bandwidth variability from node to node
  - Repeatedly run STREAM Triad on 50 different nodes
- Zone sort: Cray's kernel module to sort the free page list.
  - Does Zone sort help to reduce variability?

---

**DRAM** -
- 190 GB capacity
- ~ 90 GB/s bandwidth (measured)
- ~ 180 cycles latency (measured)

**MCDRAM** -
- 16 GB capacity
- ~ 480 GB/s bandwidth (measured)
- ~ 220 cycle latency (measured)

Note: the bandwidth and latency numbers are for 7230 KNL.
MPI BENCHMARKS ON EDISON

Edison
• 12 core Ivy Bridge
• Cray Aries with Dragonfly (5.5K nodes)

Benchmark
• 256 nodes using 1 process per node
• Allreduce across 10 jobs for different message sizes

Variability
• Highest job-to-job variability of 7x for 512 B
• Not a clear correlation of message size vs. variability

Message size (B)
256 nodes Allreduce across 10 jobs
NODE PLACEMENT (256 NODES ON THETA)

alltoall  Allreduce  Broadcast