Performance Analysis and Prediction of Multi-GPU MPI Applications

Michael Frumkin, Brian Pharris, Ramon Matas

April 7, 2016
Outline

- Distributed Application Performance Model (DAPM)
  - Tracing MPI CUDA applications
  - Integration of MPI Communication graph and CUDA dependency graph into application graph
  - Generating timeline from the application graph
  - Validation of DAPM

- Experimental results
  - Scaling of some MPI applications

- Conclusions
Need for Multi-GPU System Performance Model

- Multi-GPU is an emerging new architecture
- Need to understand how existing applications will perform
- Anticipate future usage of the systems
- System level trade-offs
  - Takes weeks of research to come up with a qualitative performance model
  - Major effort by a group of experts
- Shows direction toward a well balanced architecture
AppGraph Example: simple_mpi

Path_1: MPI_Init->kernel_a_1->MPI_Send_a_1_b->MPI_Recv_a_1_b->kernel_a_2->MPI_Send_a_2_b->MPI_Recv_a_2_b->kernel_a_3->MPI_Finalize;

Path_2: process 2

Mapping of the graph onto the system
Simple_mpi Timelines

Estimator’s generated timeline
Scaling of Simple_mpi

- **Simple_mpi run-time**
  - **Message size (B)**
  - **us**
  - **Measured**
  - **Estimated**

- **Simple_mpi run-time**
  - **Kernel latency (ms)**
  - **us**
  - **Measured**
  - **Estimated**
CUDA Aware simple_mpi

Kernel
CPU Work
cudaFree
MPI_Call
cudaMemcpy

GPU_a
Kernel_a_1
Kernel_a_2
Kernel_a_3

MPI Send/Recv

GPU_b
Kernel_b_1
Kernel_b_2
Kernel_b_3

CPU
Scaling of CUDA Aware simple_mpi

![Graph 1: Simple_mpi_cuda_aware run-time vs Message size (B)]

![Graph 2: Simple_mpi_cuda_aware run-time vs Kernel latency (ms)]
Reproducibility of the Runtime

Estimated/Measured

Ratio

simple_mpi_2
nekbone_2
nekbone_4
nekbone_6
minfft_4
mb_4
mb_6
mb_8
jacrbi_4
jacrbi_cuda_aware_4
umt_4
umt_4_mps
amg_laplace_8
amg_poisid_8
lsms_4
MCB 6 GPU - NVVP vs Estimator

Estimator generated timeline
Sensitivity of MCB 4 Ranks

![Graph showing the sensitivity of MCB 4 ranks over different acceleration factors. The graph compares various components such as 'Comm sensitivity', 'CPU sensitivity', 'GPU sensitivity', and 'Residual sensitivity'.]
Sensitivity of MCB 4 Ranks

Sensitivity to Comm time

Sensitivity to GPU time

Sensitivity to CPU time

Sensitivity to GPU time
Sensitivity of MCB 6 and 8 Ranks

**MCB_6**

**MCB_8**
Timelines miniFE

Estimator generated timeline MiniFE_4_100

Estimator generated timeline MiniFE_4_200

Kernel
CPUWork
cudaFree
MPI_Call
cudaMemcpy
miniFE 4 Ranks Sensitivity

**miniFE 4 ranks grid 100^3**

**miniFE 4 ranks grid 200^3**
miniFE 4 grid size $200^3$ Sensitivity

![Graph showing sensitivity to communication time and GPU time.](image)

- Sensitivity to Comm time
- Sensitivity to GPU time

'\text{minife.dat}'
Timelines NEKBONE

Estimator generated timeline Nekbone_4

Estimator generated timeline Nekbone_6
NEKBONE 4 and 6 Ranks Sensitivity

Nekbone 4

Nekbone 6

Speedup

Acceleration

- Comm sensitivity
- CPU sensitivity
- GPU sensitivity
- Residual sensitivity

Speedup

Acceleration

- Comm sensitivity
- CPU sensitivity
- GPU sensitivity
- Residual sensitivity
NEKBONE 4 Ranks Sensitivity

Sensitivity to GPU time

Sensitivity to Comm time
Timeline UMT 4 GPUs

Estimator generated timeline umt_4

Estimator generated timeline umt_4_mps
UMT-MPS 4 Ranks Sensitivity

UMT-MPS 4 ranks sensitivity

- Comm Sensitivity
- CPU Sensitivity
- GPU Sensitivity
- Residual Sensitivity

Speedup vs. Acceleration
Conclusions

- Multi-GPU systems are here
- Need fine balance of GPU and CPU performance and network bandwidth
- DAPM can be used as a guide to a balanced architecture
- It is tested on many MPI applications
Questions ?