Building your own GPU Research Cluster using Open Source Software stack
About the speaker and you

- **[Pradeep]** is Developer Technology engineer with NVIDIA.
  - I help Customers in parallelizing and optimizing their applications on GPUs.
  - Responsible for GPU evangelism at India and South-East Asia.

- **[Audience]**
  - Looking for building a research prototype GPU cluster
  - All open-source SW stack for GPU based clusters.
Outline

- Motivation
- Cluster - Hardware Details
- Cluster Setup - Head Node, Compute Nodes
- Management and Monitoring Snapshots
Why to build a small GPU based Cluster

- Get feel of production system and performance estimates
- Port your applications
- GPU and CPU load balancing
- Small investment
- Use it as development platform
- Early experience -> Better readiness
Today’s Focus

- Trying to build 4-16 nodes cluster
- Your first GPU based cluster
- You can build in 3 weeks, 2 weeks or less ...
- GPUs - Just add and start using them

Building GPU based Clusters - Very easy, start right now...
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Compute Node Installation
5. Head Node Installation
6. Management & Monitoring
7. Run Benchmarks & Applications
Node HW Details

- CPU Processor
- 2 PCIe x16 wide Gen2/3 connections for Tesla GPUs
- 1 PCIe x8 wide for HCI card for Infiniband
- 2 Network ports
- Min of 16/24 GB DDR3 RAM
- SMPS with required power supply
- 2 x 1 TB HDD
Tesla™ for Supercomputing

- **Choose the right Form Factor** - Kepler GPUs are available in
  - Workstation products - C Series
  - Server products - M Series

- **Different options for adding GPUs**
  - Add C series GPUs to existing Workstations
  - Buy a workstation & have C series GPUs
  - Buy servers with M series
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Head Node Installation
5. Compute Node Installation
6. Management & Monitoring
7. Run Benchmarks & Applications

- Choose Correct HW
- Assemble & Physical Deployment
- Ensure Space/Power & cooling
- Head Node Installation
- Compute Node Installation
- Management & Monitoring
- Run Benchmarks & Applications
Other Hardware & real-state

- Space
- Power & Cooling
- Network - Infiniband
- Storage
- Maintenance/repair
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
Setup

- Physical Deployment of cluster
- Head Node & Compute Nodes connections
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
Head Node Installation

- Linux distribution, Open Source
- Customizable, Quick & Easy
- Contains integral components for Clusters - MPI
- Flexibility with compute nodes
- Batch Processing

Rocks Installation

CUDA Software

High Speed Interconnect

- Requirements - Driver, CUDA Toolkit, CUDA Samples
- Downloadable from
- CUDA 5 provides unified Package

- Open standard -
  http://www.infinibandta.org/home
- Install drivers on Head node (Download from vendor site)

- Downloadable from
- CUDA 5 provides unified Package

- Open standard -
  http://www.infinibandta.org/home
- Install drivers on Head node (Download from vendor site)
Head Node Installation... contd

Head Node Installation

SW Package Installation

Nagios Installation -
- Monitoring of network services
- Monitoring of host resources
- Web interface and many more ...

NRPE Installation -
- Execute Nagios plugins on remote machines
- Enables monitoring of local resources
Steps in building GPU based Clusters

1. Choose Correct Hardware
2. Ensure Space/Power & Cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
Compute Node Installation

- On head node: Execute "insert-ethers"
- Choose "compute Nodes" as the new node to be added
- Boot a compute node in installation mode - Network boot or ROCKS CD

- Requirements - Install NVIDIA Driver
- No need to install - CUDA Toolkit, CUDA Samples

- Install NRPE package
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
System Management for GPU

- **nvidia-smi utility** -
  - Thermal Monitoring Metrics - GPU temperatures, chassis inlet/outlet temperatures
  - System Information - firmware revision, configuration info
  - System State - Fan states, GPU faults, Power system fault etc.

- **nvidia-smi allows you**
  - Different Compute modes - Default/Exclusive/prohibited
  - ECC on/off
GPU Monitoring

- **NVIDIA Provides “TESLA Deployment Kit”**
  - Set of tools for better managing Tesla GPUs
  - 2 main components - NVML and nvidia-healthmon

- **NVML can be used from Python or Perl**
  - NVML - Set of APIs provide state information for GPU monitoring.

- **NMVL has been integrated into Ganglia gmond.**
nvidia-healthmon

- Quick health check, Not a full diagnostic
- Suggest remedies to SW and system configuration problems
- Feature Set
  - Basic CUDA and NVML sanity check
  - Diagnosis of GPU failures
  - Check for conflicting drivers
  - Poorly seated GPU detection
  - Check for disconnected power cables
  - ECC error detection and reporting
  - Bandwidth test
Nagios

Monitoring of Nodes

Monitoring of Services
Nagios - GPU Memory Usage

Monitoring of GPU Memory

Service State Information
- Current Status: OK (not exceed 1.8% of 475)
- Status Information: GPU Memory Free: 100%
- Performance Data: OK
- Current Attempt: 1/4 (HARD state)
- Last Check Time: 03-09-2013 14:24:05
- Check Type: ACTIVE
- Check Latency / Duration: 0.0170/0.225 seconds
- Next Scheduled Check: 03-09-2013 14:28:05
- Last State Change: 02-12-2013 12:09:26
- Last Notification: N/A (notification 0)
- Is This Service Failing?: NO (0.00% state change)
- In Scheduled Downtime?: NO
- Last Update: 03-09-2013 14:26:05 (0d 0h 0m 8s ago)

Active Checks: ENABLED
Passive Checks: ENABLED
Obssening: ENABLED
Notifications: ENABLED
Event Handler: ENABLED
Flap Detection: ENABLED
Steps in building GPU based Clusters

1. Choose Correct HW
2. Ensure Space/Power & cooling
3. Assemble & Physical Deployment
4. Management & Monitoring
5. Compute Node Installation
6. Head Node Installation
7. Run Benchmarks & Applications
Benchmarks

- GPUs
  - devicequery
  - Bandwidth Test

- Infiniband
  - Bandwidth and latency test
  - `<MPI Install PATH>/tests/osu_benchmarks-3.1.1`
  - Use Open Source CUDA-aware MPI implementation like MVAPICH2

- Application
  - LINPACK
Questions?
See you at GTC 2014