Technical
OVERVIEW

AMD | SeaMicro

AMD Reference Architecture for SeaMicro SM15000 Server and Hortonworks Hadoop 2.0 (YARN) Solution

February 2015
www.seamicro.com

Table of Contents

Hortonworks

Overview

This document describes the best practices for the deployment of the new Hadoop 2.0 framework (YARN) on AMD's SeaMicro SM15000 server. It contains specific recommendations for hardware and software configurations to optimize the system for the input/output (I/O) intensive MapReduce programs, e.g., the Terasort benchmark. These best practices have been thoroughly tested and validated using Hortonworks 2.15 Hadoop distribution (certification achieved in December, 2014).

The information contained in this document is intended for use by systems engineers and administrators who are deploying the Hadoop 2.0 framework on the SeaMicro SM15000 server. Intermediate expertise of Hadoop is recommended to effectively deploy the reference architecture.

AMD's SeaMicro SM15000 Solution

The SeaMicro SM15000 uses the patented SeaMicro Freedom™ Supercompute Fabric to interconnect computing, networking and storage together in a single 10-rack unit (RU) system (see Figures 1 and 2). It can be configured and managed through a single management interface with the following hardware components:

- 64 compute cards with AMD Opteron® 4365EE or Intel® Xeon® E3-1265Lv3 processor
- 512 cores and up to 4 TB of DRAM (8 GB per core)
- 1.28 Tbps Freedom Supercompute Fabric bandwidth
- 64 internal SSD or HDD drives
- Up to 5.4 PB of direct attached storage (16 x mini SAS connectors 4 x 6 Gbps, 16 fabric storage devices, 84 SAS/SATA drives per fabric storage device, 4 TB per drive)
- 160 Gbps of network uplink bandwidth (16 x 10 Gbps full duplex)

Figure 1: SeaMicro SM15000 at front and right side view showing 64 internal disks and 32 compute cards.

Figure 2: Back view showing 6 x 10 Gbps, 16 x 1 Gbps uplinks, and 16 x mini SAS connectors 4 x 6 Gbps to external fabric storage devices.

Computing infrastructure

64 compute cards are enclosed in the 10RU chassis. Power consumption is 65W per server. A table of core technologies within the compute cards is provided below.

Table 1: SeaMicro SM15000 Compute Card Core Technologies

Compute card	CPU model	Number of cores	Base/Boost frequency	CPU Power	Memory support
AMD	**Opteron 4365EE**	**8**	**2.0GHz/2.7GHz**	**40W**	**64GB DDR3 1333 MHz**
INTEL	**Haswell E3-1265Lv3**	**4, 8 with hyperthreading**	**2.5GHz/3.1GHz**	**45W**	**32GB DDR3 1600 MHz**

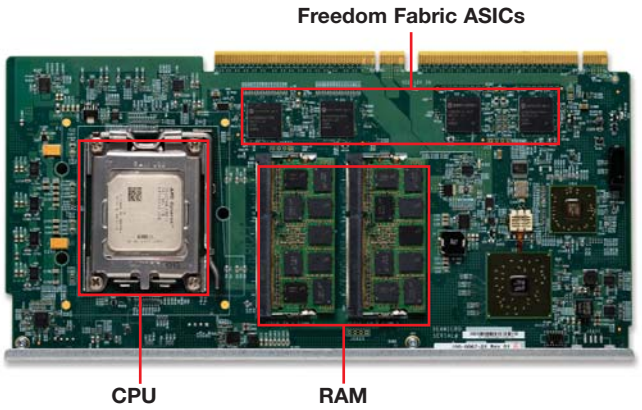

Figure 3: Each compute card has eight fabric nodes providing an aggregated Ethernet bandwidth of 8 x 1 Gbps full duplex.

Networking infrastructure

Each compute card has 8 fabric nodes that provide an aggregated Ethernet bandwidth of 8 x 1 Gbps full duplex. Compute cards connect to the SeaMicro supercomputing fabric, a 3D torus fabric depicted below in Figure 4.

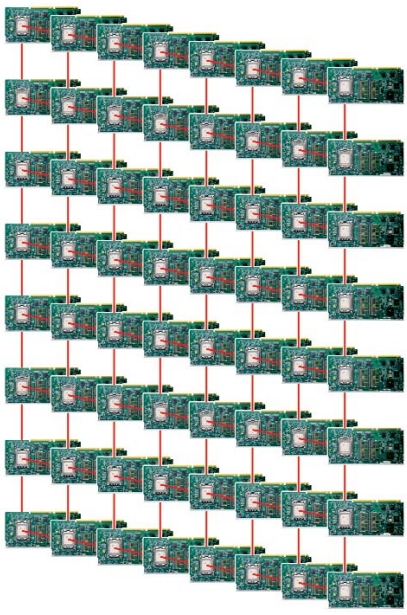

Figure 4: SeaMicro 3D torus supercomputing fabric.

SeaMicro SM15000 Supercomputing Fabric Main Features

- 3D torus network fabric
- 8 x 8 x 8 fabric nodes
- Diameter (max hop) 4 + 4 + 4 = 12
- Theoretical cross section bandwidth = 2 (periodic) x eight x eight (section) x 2(bidir) x 2.5 Gbps/link = 640 Gbps
- Compute, storage, management cards are plugged into the network fabric
- Support for hot plugged compute cards

Storage infrastructure

The storage infrastructure is attached to the computing infrastructure via the SeaMicro Freedom Fabric. A set of eight storage cards serve disk data to the 64 compute cards. Disks (HDD and/or SSD) can be either within the storage card, which consists of eight disk bays, or externally attached through 4 x 6Gbps mini SAS ports to fabric storage devices of up to eight HDD disks at 7200 rpms. The system has up to 16 mini SAS ports to attach fabric storage devices. Figure 5 and 6 illustrate the computing, fabric, and storage infrastructure for Hadoop.

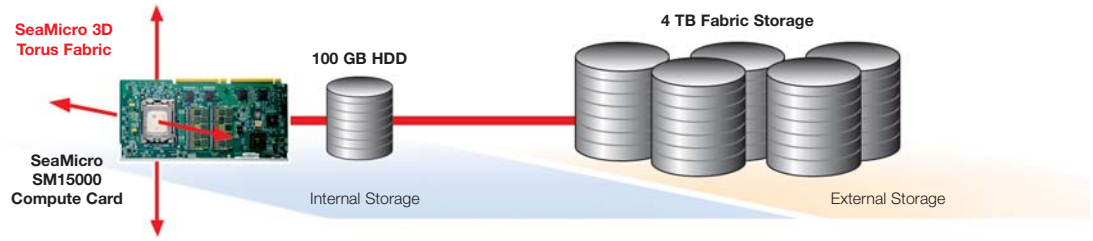

Figure 5: For this reference architecture each SeaMicro SM15000 compute card has internal 100 GB HDD for the OS and 2-5 HDD (4 TB each) for external storage.

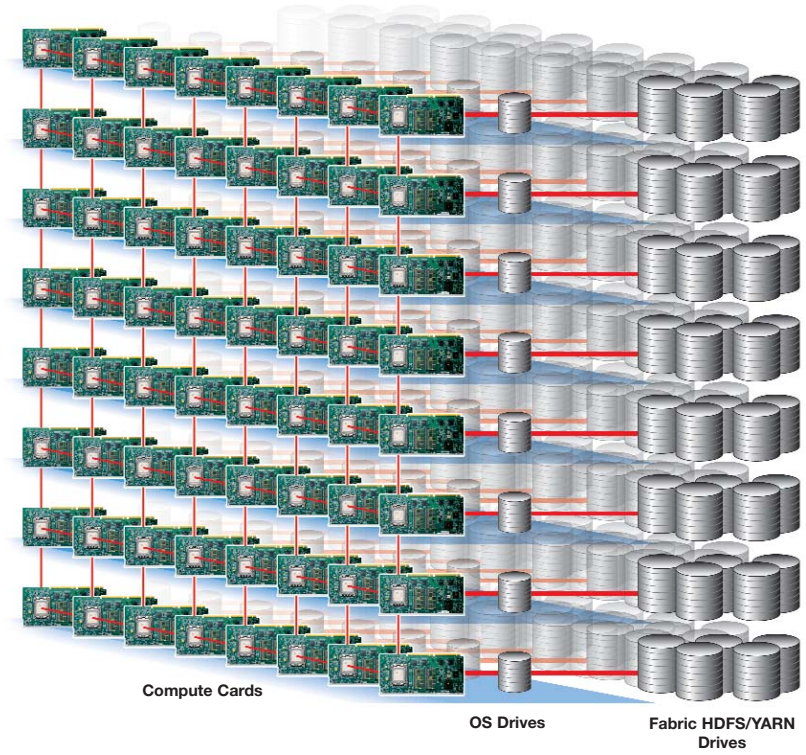

Figure 6: Computing, fabric and storage infrastructure for Hadoop.

All 64 compute cards (0–63) within a chassis are grouped in sets of eight compute cards. Each set of compute cards corresponds to a different storage card numbered from 0 to 7. Each storage card has eight internal drives that serve the OS for each compute card. Each storage card also provides the Hadoop distributed file system (HDFS) to those same compute cards through one fabric storage device directly attached through a mini SAS cable of 4 x 6 Gbps. Provided that each compute card will use only 2 to 5 disks from the fabric storage device for HDFS, the eight compute cards will need a total of 16 to 40 disks, respectively, all being served from the same storage card connected to the same fabric storage device. Since a fabric storage device has many more drives than the required 32–80 drives, e.g., 84 drives, the fabric storage device can be partitioned into two zones. If you would like to receive the paper "Technology Brief on SAS Zoning Configuration" for details on how to partition and assign disks from fabric storage devices through specific storage cards and to specific compute cards, please contact your AMD sales representative or email seamicro@amd.com.

Software Setup

Operating system settings

Although a wide range of operating systems (Linux and Windows) are supported, the setup described in this document refers to the 64-bit operating system CentOS, version 6.5. There is a minimal set of recommended changes related to the operating system.

```
Changes to sysctl.conf
Add to /etc/sysctl.conf file the following new values
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 250000
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304
net.core.rmem_default = 4194304
net.core.wmem_default = 4194304
net.core.optmem = 4194304

net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 1
net.ipv4.tcp_rmem = "4096 87380 4194304"
net.ipv4.tcp_wmem = "4096 87380 4194304"
net.ipv4.tcp_low_latency = 1

vm.swappiness = 0

fs.file-max = 100000
```

Most of them refer to the TCP sockets.

Swappiness and file-max refer to memory and file system.

Changes to limits.conf

Add to /etc/security/limits.conf file the new values for the number of files and processes.

```
* hard nofile 65536
* soft nofile 65536
* hard nproc  65536
* soft nproc  65536
```

Tuning of interrupt handling

Interrupt handling via irqbalance service could be either set on or manually find the IRQs and assign all the cores to each of them for both networking and storage devices. On the datanodes, they are handled by core ids 0, 2, 4 and 6 on an 8 core hyperthreaded Intel CPU.

An excerpt of the output of properly balanced interrupt handling is provided below, where IRQs 16, 17, 18 and 19, which deal with devices eth0-7 are handled fairly equally among core IDs 0, 2, 4 and 6.

```
[root@datanode ~]# cat /proc/interrupts
```

```
           CPU0       CPU1       CPU2       CPU3       CPU4       CPU5       CPU6       CPU7
 0:        136          0          0          0          0          0          0          0   IR-IO-APIC-edge      timer
 4:        153          0          0          0          0          0          0          0   IR-IO-APIC-edge      serial
 8:          1          0          0          0          0          0          0          0   IR-IO-APIC-edge      rtc0
 9:          0          0          0          0          0          0          0          0   IR-IO-APIC-fasteoi   acpi
16:    2062755          0   16758786          0   15284127          0    2624381          0   IR-IO-APIC-fasteoi   ehci_hcd:usb1, ahci, ahci, eth3, eth7
17:        138          0          0          0          0          0   35177929          0   IR-IO-APIC-fasteoi   ahci, ahci, eth0, eth4
18:      06460          0   17052262          0   20955800          0   15967765          0   IR-IO-APIC-fasteoi   ahci, ahci, eth1, eth5
19:   35557388          0          0          0          0          0          0          0   IR-IO-APIC-fasteoi   ahci, ahci, eth2, eth6
```

Synchronizing compute cards

Set Network Time Protocol service on all compute nodes so they are all synchronized.

```
chkconfig ntpd on
service ntpd start
```

Changes to rc.local

Add to /etc/rc.local file to disable large pages and memory fragmentation.

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

Set power management to use the performance governor as follows in /etc/rc.local file.

```
cpufreq-set -r -g performance
```

where `cpufreq-set` is part of cpufrequtils package, which can be installed on CentOS via yum command:

```
yum install cpufrequtils
```

Network settings

Update e1000 driver to 8.0.35-NAPI, since default is 7.3.21-k8-NAPI. A link to the website that hosts the driver is provided: http://sourceforge.net/projects/e1000/files/e1000%20stable/8.0.35/

Update e1000 options to change Interrupt Throttle rate by having a /etc/modules/e1000.conf file with the following contents:

```
options e1000 InterruptThrottleRate=1,1,1,1,1,1,1,1
```

Setup NIC bonding across the 8 x 1 Gbps interfaces on each compute card with hashing layer 3+4.

Use for that our SeaMicro script nicbonding.sh with arguments

```
./nicbonding.sh –b 1 –n 8 –h layer3+4
```

The script is provided from support via the SeaMicro FTP server.

Finally, increase MTU size to 9000 (jumbo frames) for the bonding interface by adding at /etc/rc.local the following command

```
ifconfig bond0 mtu 9000
```

Storage settings

Formatting of file system for HDFS using ext4 file system.

Add to the /etc/mke2fs.conf file the following hadoop format under [fs_types] section

```
[fs_types]
ext3 = {…}
ext4 = {…}
…….
hadoop = {
features = has_journal,extent,huge_file,flex_bg,uninit_bg,dir_nlink,extra_isize
  inode_ratio = 131072
  blocksize = -1
  reserved_ratio = 0
  default_mntopts = acl,user_xaddr
}
```

Then format the 2 x 4TB disk (/dev/sdb and /dev/sdc) as follows

```
mkfs.ext4 -F -T hadoop /dev/sdb
mkfs.ext4 -F -T hadoop /dev/sdc
```

Mount them at directories /4TBb and /4TBc with the following options under /etc/fstab file

```
/dev/sdb  /4TBb ext4  inode_readahead_blks=128,data=writeback,noatime,nodev,nobarrier   0 0
/dev/sdc  /4TBc ext4  inode_readahead_blks=128,data=writeback,noatime,nodev,nobarrier   0 0
```

Hadoop 2.0 (YARN) Setup

This section assumes prior knowledge at the level of administration of Hadoop. For detailed instructions on how to install Hadoop, please refer to the well documented installation guides provided by the vendors of Hadoop distributions.

The below changes provide the minimal modifications necessary from a default Hadoop setup that are recommended for the SeaMicro SM15000 server.

Hadoop 2.0 (YARN) Distributions

It has been validated using Hortonworks 2.15 Hadoop distribution.

Hortonworks is installed with Ambari.

A local repository is recommended in order to speed up the installation process of parcels and rpms for Hortonworks on all 64 compute cards.

Roles

Out of the 64 compute nodes, four are dedicated to master roles and 60 to datanodes. Table 2 lists the roles for each compute node.

Table 2: Compute Node Roles

Role	Name node	Secondary Name node	Resource manager	Job History Monitoring services	Data nodes HDFS/YARN
Number of nodes	**1**	**1**	**1**	**1**	**60**

No rack awareness within the chassis

All computing is set within the same rack. Tests using 32 and 32 compute nodes on different racks did not provide any performance benefits (see Tables 3 and 4).

Table 3: HDFS Settings for Terasort Benchmark

File	Property	value
hdfs-site.xml	**dfs.blocksize**	**805,306,368 (768MB)**

Table 4: YARN Settings

File	Property	value
yarn-default.xml	**yarn.nodemanager.resource.cpu-vcores**	**8**
yarn-default.xml	**yarn.scheduler.maximum-allocation-vcores**	**8**
yarn-default.xml	**yarn.nodemanager.resource.memory-mb**	**24576**

Table 6: MapReduce Settings for Terasort Benchmark

File	Property	value
mapred-site.xml	**mapreduce.map.memory.mb**	**3072**
mapred-site.xml	**mapreduce.map.java.opts**	**2576980377**
mapred-site.xml	**mapreduce.reduce.memory.mb**	**5120**
mapred-site.xml	**mapreduce.reduce.java.opts**	**4294967296**
mapred-site.xml	**mapreduce.job.reduce.slowstart.completedmaps**	**0.0**
mapred-site.xml	**mapreduce.task.io.sort.mb**	**1024**
mapred-site.xml	**mapreduce.task.io.sort.factor**	**100**
mapred-site.xml	**mapreduce.map.sort.spill.percent**	**0.95**
mapred-site.xml	**mapreduce.reduce.shuffle.parallelcopies**	**60**
mapred-site.xml	**mapreduce.tasktracker.http.threads**	**60**
mapred-site.xml	**mapreduce.reduce.shuffle.input.buffer.percent**	**0.95**
mapred-site.xml	**mapreduce.output.fileoutputformat.compress**	**false**
mapred-site.xml	**mapreduce.map.output.compress**	**true**
mapred-site.xml	**mapreduce.map.output.compress.codec**	**Snappy**

YARN (Yet Another Resource Negotiator) settings are meant to use the eight cores on the compute cards that work as datanodes and about 24 GB (24,576 MB) of memory, out of 32 GB available. If the compute card has 64 GB, then increase the value of yarn.nodemanager.resource.memory-mb to 57344 MB.

Each MapReduce application requires different settings that can be passed directly to the job. Rather than passing them to the job, those settings have been hardcoded in files hdfs-site.xml and mapred-site.xml for documentation purposes. ***The specific settings in hdfs-site.xml and mapred-site.xml files are solely meant for*** the optimization of Teragen and Terasort benchmark.

These settings allow MapReduce to run concurrently at the beginning with 8 mappers* 3 GB =24 GB per datanode and bit a bit overlap reducer tasks up to 4 reducers * 5 GB = 20 GB < 24 GB per datanode. A Terasort job uses 60 datanodes *4 reducers = 240 reducers, with a total memory of 5 GB*240 reducers = 1.2 TB of RAM, when sorting a 1 TB file.

AMD | SeaMicro, Inc.
1 AMD Place, Sunnyvale, CA 94085 USA
www.seamicro.com or www.amd.com
408 749-3076