The Abdus Salam International Centre for Theoretical Physics

United Nations Educational, Scientific and Cultural Organization

IAEA International Atomic Energy Agency

2494-5

Workshop on High Performance Computing (HPC) Architecture and Applications in the ICTP

14 - 25 October 2013

Managing HPC Clusters

Ershaad Basheer

Temple University
Philadelphia
USA

Managing HPC Clusters

Remote management • Monitoring • Job scheduling • Power/Cooling

Ershaad Basheer
Research Assistant Professor
ebasheer@temple.edu

URL

https://sites.google.com/site/ictphpcworkshop2013/

Remote Management

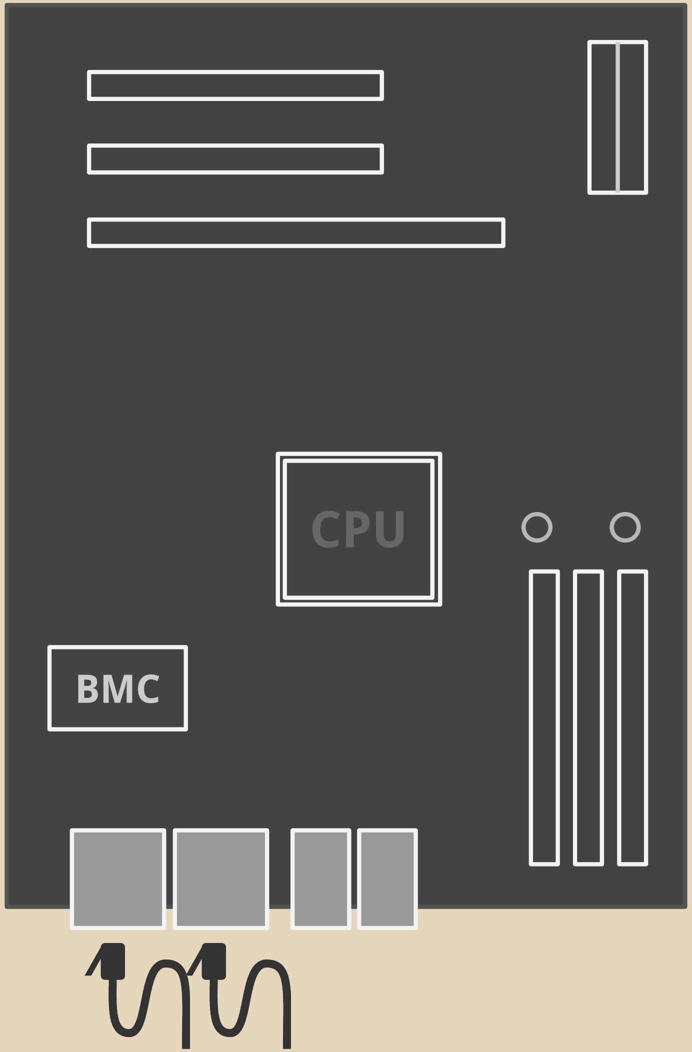
IPMI

Remote Management

- Remote management and HPC
  - Remote management can ease administration of large clusters
  - Can avoid the need for the administrator to be physically present in the datacenter
  - Two types of remote management
    - In-band
      - Remotely access features of a running operating system (OS)
    - Out-of-band
      - Management features provided by the platform hardware

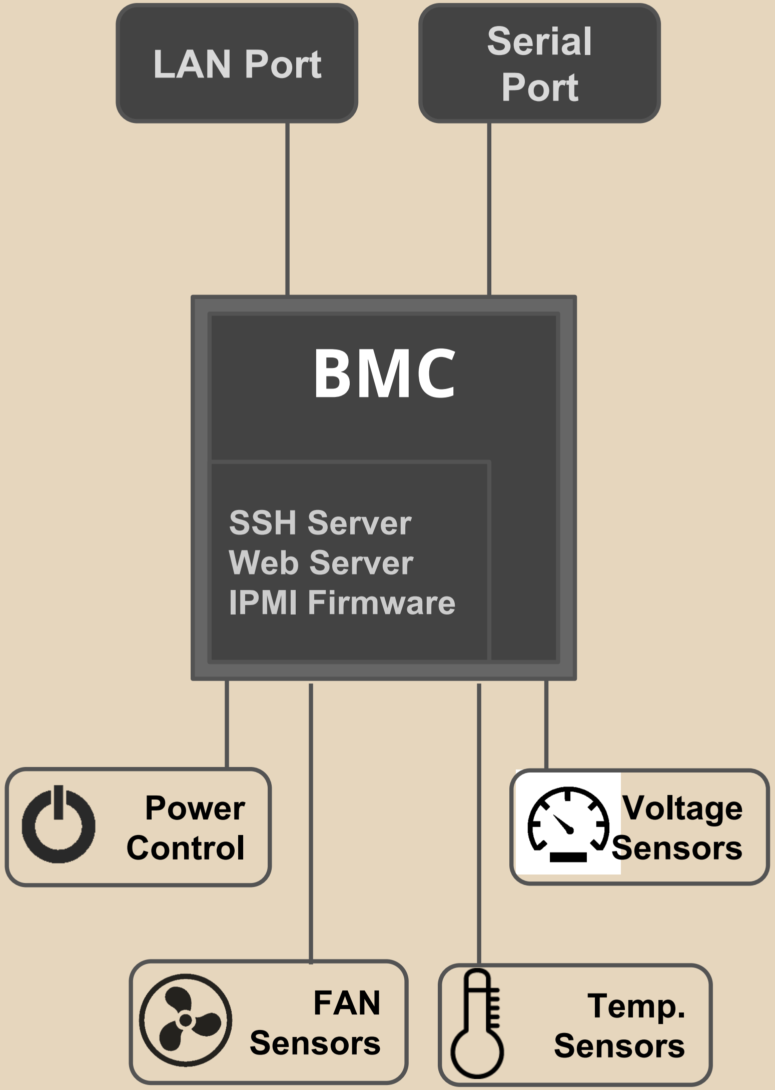
Out-of-band Management

- Monitor and control machines remotely
- Interface provided by dedicated hardware called **B**aseboard **M**anagement **C**ontroller
- Embedded microcontroller with flash storage



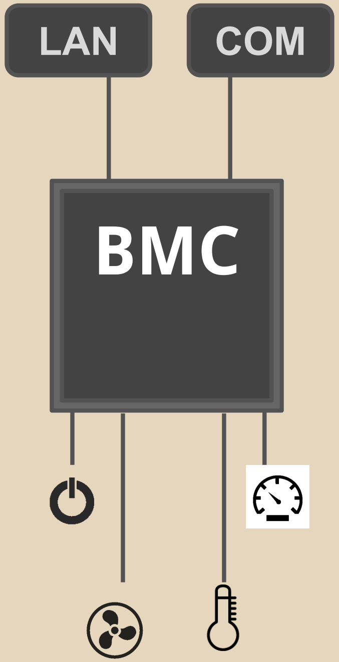
Out-of-band Management

- Powered on as long as the machine is connected to a power supply
- Usually runs an embedded operating system (like embedded Linux)



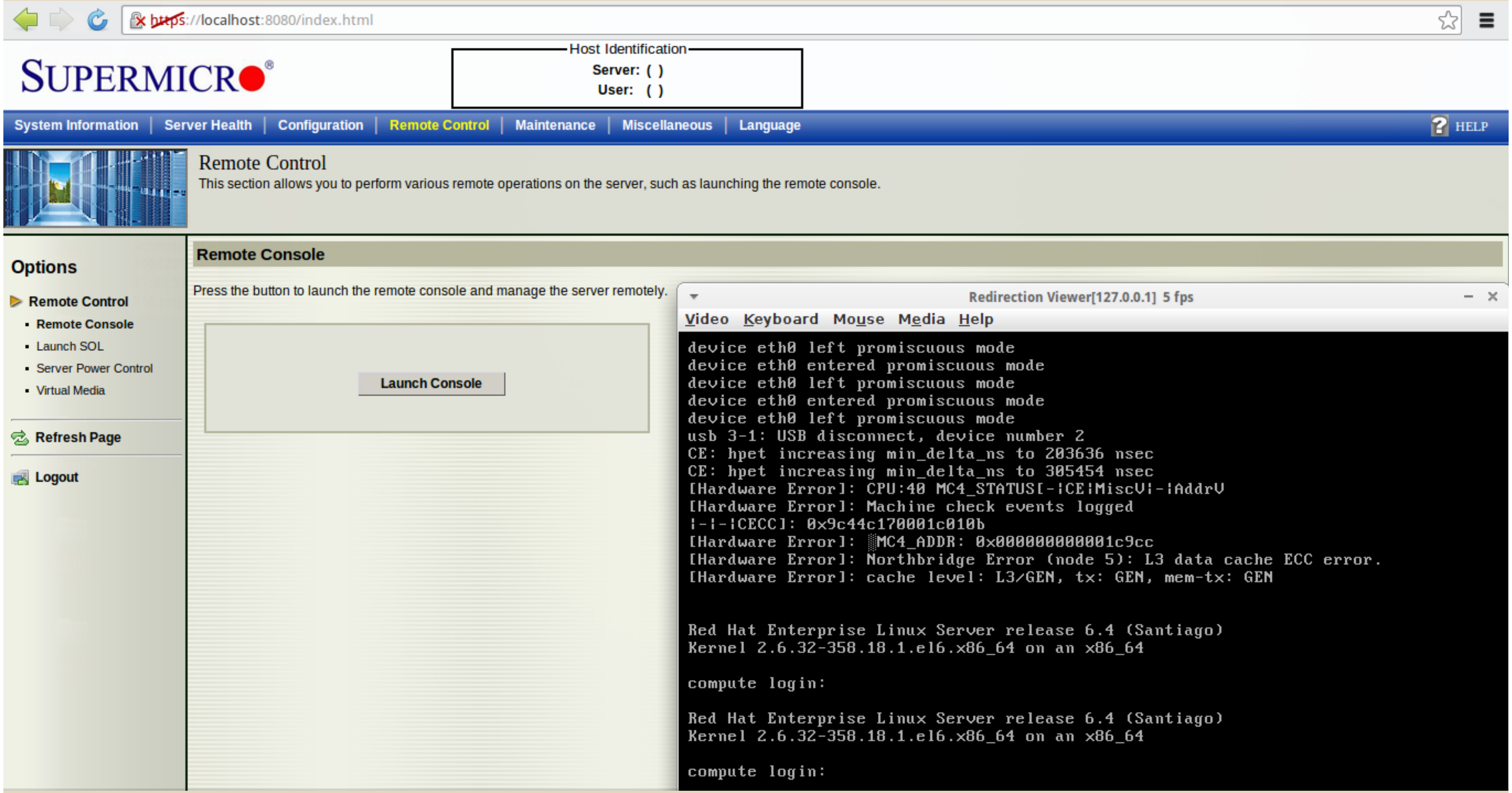
BMC

- BMC hosts
  - Web or shell interface that runs on the embedded OS (Embedded Web/SSH server)
  - Some interfaces need to be accessed through vendor specific client software
  - IPMI firmware
- BMCs can be accessed via a shared or dedicated management ethernet port or a serial port



BMC

- Web interface may also include Java based remote console
- Allows to view display, control mouse and keyboard through a browser with Java runtime
- Implemented using VNC protocol



Remote Control

This section allows you to perform various remote operations on the server, such as launching the remote console.

Remote Console

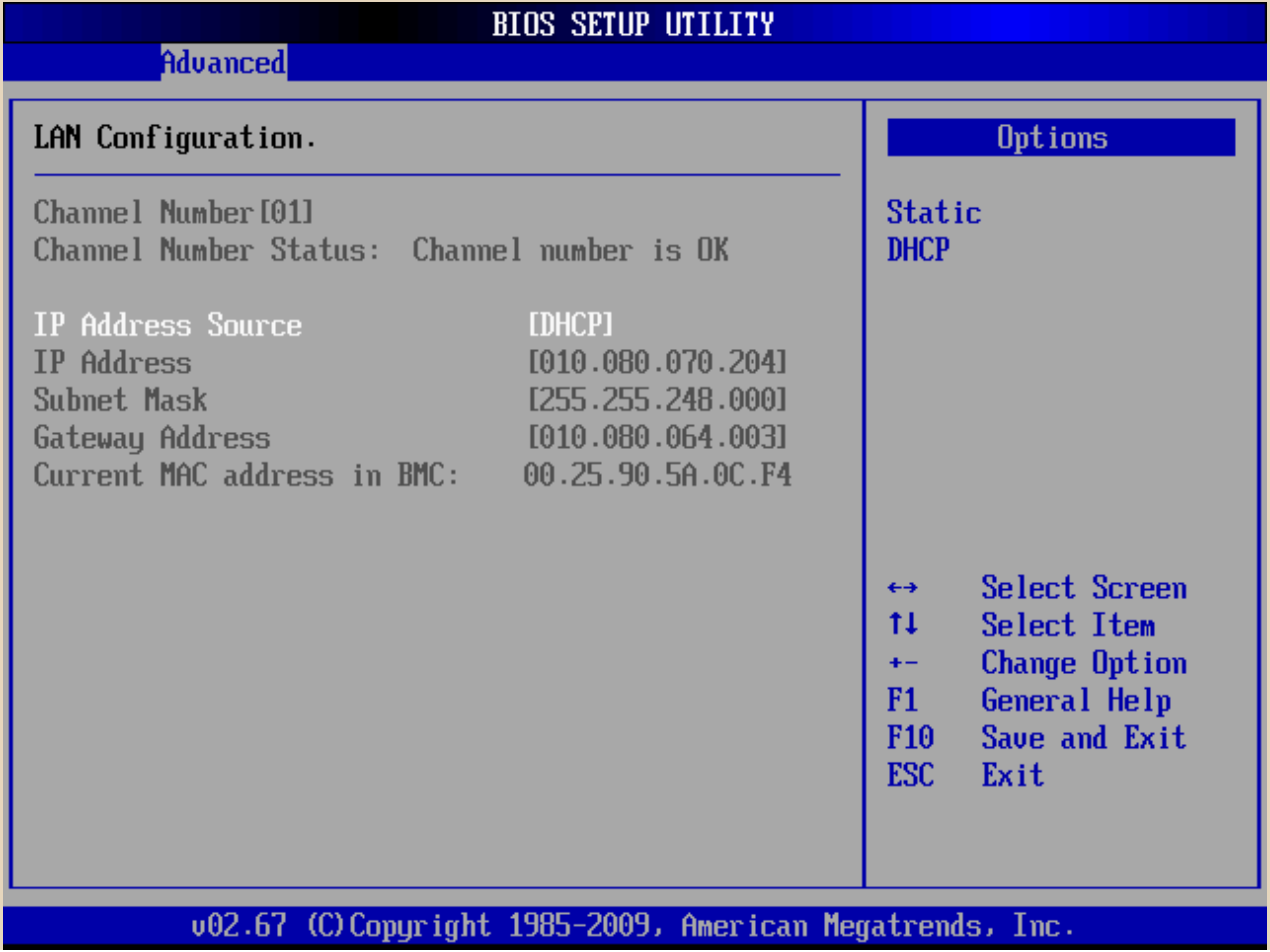
Press the button to launch the remote console and manage the server remotely.

```
device eth0 left promiscuous mode
device eth0 entered promiscuous mode
device eth0 left promiscuous mode
device eth0 entered promiscuous mode
device eth0 left promiscuous mode
usb 3-1: USB disconnect, device number 2
CE: hpet increasing min_delta_ns to 203636 nsec
CE: hpet increasing min_delta_ns to 305454 nsec
[Hardware Error]: CPU:40 MC4_STATUS[-|CE|MiscV|-|AddrV
[Hardware Error]: Machine check events logged
|-|-|CECC]: 0x9c44c170001c010b
[Hardware Error]: ░MC4_ADDR: 0x000000000001c9cc
[Hardware Error]: Northbridge Error (node 5): L3 data cache ECC error.
[Hardware Error]: cache level: L3/GEN, tx: GEN, mem-tx: GEN


Red Hat Enterprise Linux Server release 6.4 (Santiago)
Kernel 2.6.32-358.18.1.el6.x86_64 on an x86_64

compute login:

Red Hat Enterprise Linux Server release 6.4 (Santiago)
Kernel 2.6.32-358.18.1.el6.x86_64 on an x86_64

compute login:
```

Out-of-band Management

- Out-of-band management provides several practical advantages
- Allows administrator to perform operations such as
  - Remotely control power (on, off, reset, cycle)
  - Monitor health (voltages, temperatures, MCE, etc.)
  - Allow the operator to access a serial console over the LAN
  - Modify boot device order
  - Modify LAN port parameters
  - Blink indicator lights on chassis
  - etc..

Configure BMC LAN Interface (IPMI over LAN Interface)



```
                         BIOS SETUP UTILITY
      Advanced

 LAN Configuration.                                       Options

 Channel Number[01]                                      Static
 Channel Number Status:  Channel number is OK            DHCP

 IP Address Source                [DHCP]
 IP Address                       [010.080.070.204]
 Subnet Mask                      [255.255.248.000]
 Gateway Address                  [010.080.064.003]
 Current MAC address in BMC:     00.25.90.5A.0C.F4

                                                         ↔    Select Screen
                                                         ↑↓   Select Item
                                                         +-   Change Option
                                                         F1   General Help
                                                         F10  Save and Exit
                                                         ESC  Exit

         v02.67 (C)Copyright 1985-2009, American Megatrends, Inc.
```

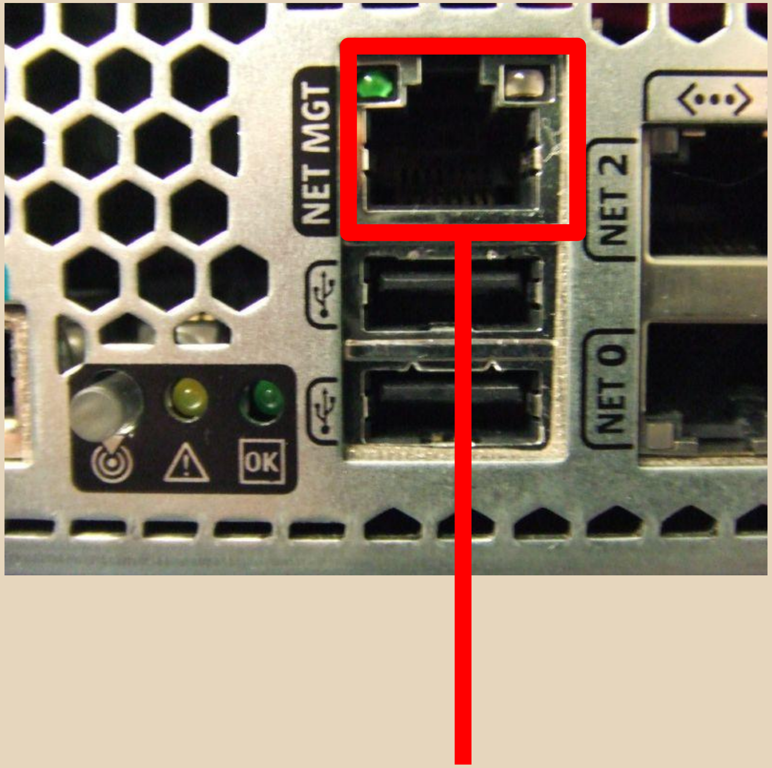
IPMI

- Intelligent **P**latform **M**anagement **I**nterface
  - Is a standardized vendor neutral protocol for out-of-band management
  - IPMI client software is required to talk to a server that supports IPMI
  - Can communicate with the server
    - In-band (from within OS) or
      - Allows you to change the IPMI password if you forgot it for instance
    - out-of-band: (over the the network: IPMI over LAN)
  - Several client implementations exist
    - FreeIPMI, OpenIPMI, Ipmitool, Ipmiutil...

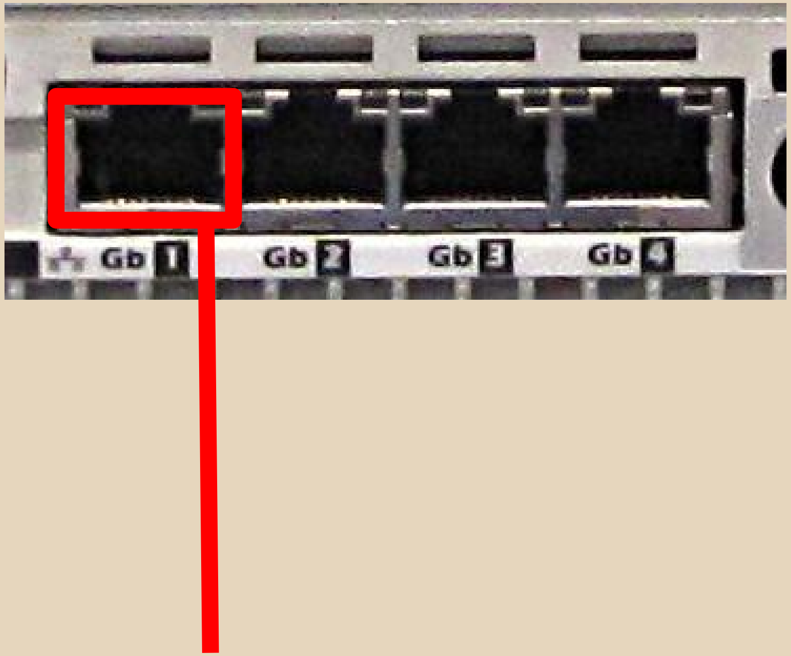
Configuring IPMI

- Enable IPMI on the node
  - Done from within the BIOS
  - Usually differs slightly between manufacturers
  - IPMI interface accessed through serial or LAN port
    - Some vendors require enabling IPMI over LAN
    - LAN port could be dedicated for management or shared with a Network Interface Card (NIC)
    - When ports are shared, management interface and NIC interface have different MAC addresses

Configuring IPMI



Dedicated Management Port



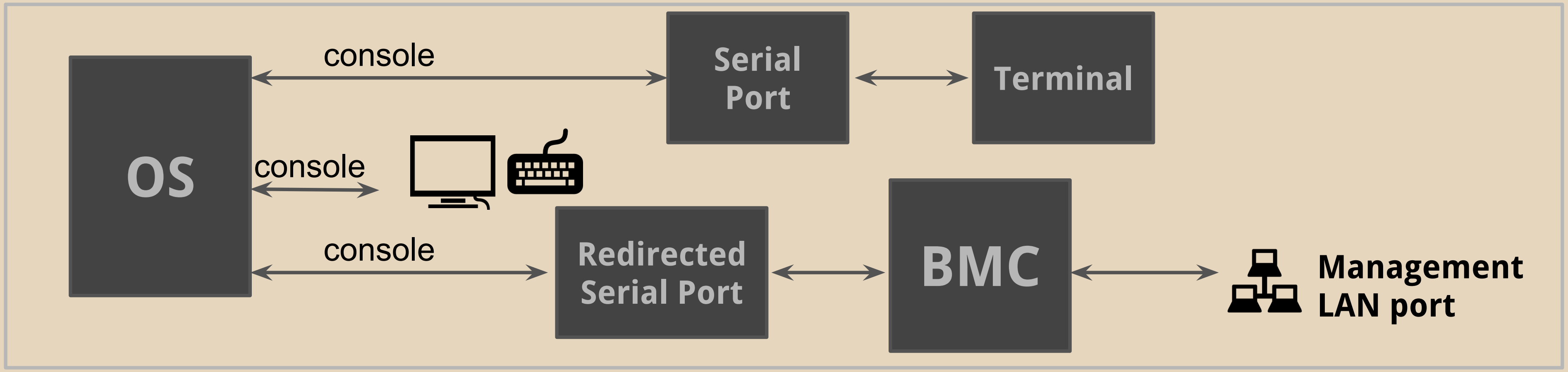
Shared Management Port (Shared with NIC1)

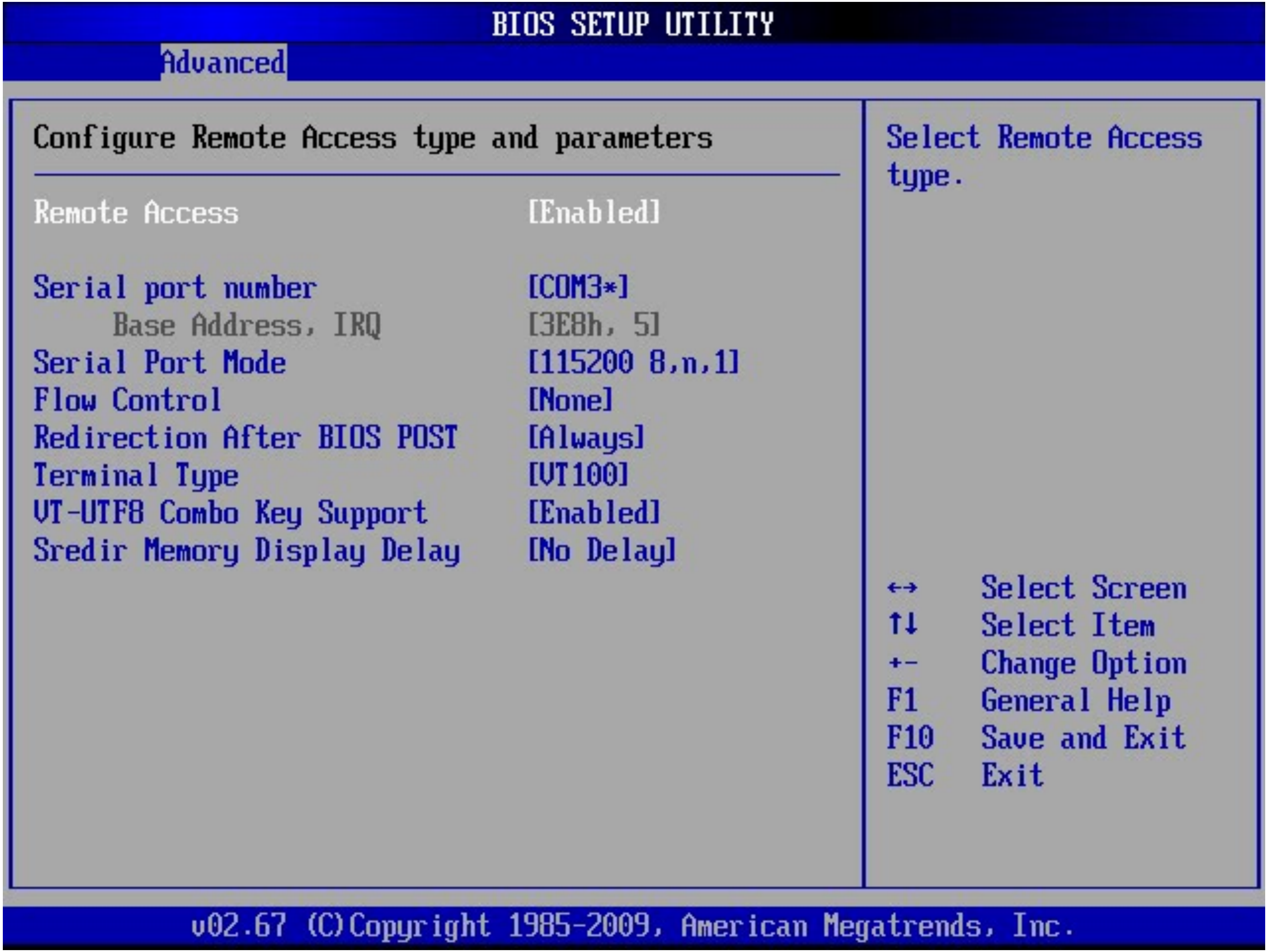
image on left from http://www.flickr.com/photos/clonedmilkmen/

Configuring IPMI: Serial over LAN

- Ways to interact with the Linux console
  - Terminal: Display and Keyboard
  - Remote shell: Telnet, SSH, etc.
  - Serial device: Terminal emulator connected to serial port

- IPMI 2.0: Redirection of serial port to LAN (SOL)





```
Configure Remote Access type and parameters

Remote Access                         [Enabled]

Serial port number                    [COM3*]
     Base Address, IRQ                [3E8h, 5]
Serial Port Mode                      [115200 8,n,1]
Flow Control                          [None]
Redirection After BIOS POST           [Always]
Terminal Type                         [VT100]
VT-UTF8 Combo Key Support             [Enabled]
Sredir Memory Display Delay           [No Delay]
```

Select Remote Access type.

Configuring IPMI: Serial over LAN

- Note the baud rate and serial port (COM3)
- COM3 is special: This serial port is redirected to IPMI 2.0 Serial over LAN
  - Hardware dependent: different vendors redirect different serial ports to the networked interface
  - Check manual
- LInux and X terminals support VTxxx

Configuring IPMI: Client

- We need to use a client to communicate with the IPMI enabled server
- The client must be running on a machine that has connectivity to the target machine’s management port
- Several implementations exist. Following examples use `ipmitool`

Configuring IPMI: Client

`ipmitool` *`options <command>`*

- Essential options

  `-I`

  - IPMI interface to use
  - Usually `lan` for IPMI 1.5 and `lanplus` for IPMI 2.0

  `-H`

  - Hostname or IP address of management port

Configuring IPMI: Client

- `ipmitool` *`options <command>`*
- Essential options

  `-U`

  - Username (default depends on vendor)

  `-P`

  - Password (default depends on vendor)
  - Not recommended to specify password as command line option

  `-a`

  - Prompt for password

Configuring IPMI: Client

- Examples

Check power status of a machine

```
$ ipmitool -I lanplus -H sp-h074 -U root -a power status
Password:
Chassis Power is on
```

Configuring IPMI: Client

- Command categories

```
$ ipmitool help
Commands:
    raw           Send a RAW IPMI request and print response
    i2c           Send an I2C Master Write-Read command and print response
    spd           Print SPD info from remote I2C device
    lan           Configure LAN Channels
    chassis       Get chassis status and set power state
    power         Shortcut to chassis power commands
    event         Send pre-defined events to MC
    mc            Management Controller status and global enables
    sdr           Print Sensor Data Repository entries and readings
    sensor        Print detailed sensor information
    fru           Print built-in FRU and scan SDR for FRU locators
    gendev        Read/Write Device associated with Generic Device
                                        ...
                                        ...
```

Configuring IPMI: Client

- Command categories and subcategories

```
$ ipmitool -I lanplus -H sp-h074 -U root -a chassis help
Password:
Chassis Commands:  status, power, identify, policy, restart_cause,
poh, bootdev, bootparam, selftest


$ ipmitool -I lanplus -H sp-h074 -U root -a chassis power help
Password:
chassis power Commands: status, on, off, cycle, reset, diag, soft
```

Configuring IPMI: Serial Over LAN

- Soon after boot, console is redirected to serial port if enabled in BIOS
- By default, the bootloader and Linux display their console only on the terminal screen
- Bootloader and Linux can be configured to display the console on a serial port
- For SOL, the selected serial port must match the port that is redirected

Configuring GRUB for Serial

- /etc/default/grub must be edited to enable interaction via serial in both Grub and the Linux kernel
- Variables that need to be modified are

GRUB_TERMINAL

GRUB_SERIAL_COMMAND

GRUB_CMDLINE_LINUX

Configuring IPMI: Client

- Serial over LAN (SOL)
- To activate serial over LAN session

```
$ ipmitool -I lanplus -H sp-h074 -U root -a sol activate
Password:
[SOL Session operational.  Use ~? for help]

Red Hat Enterprise Linux Server release 6.2 (Santiago)
Kernel 2.6.32-71.18.2.el6.x86_64 on an x86_64

h074 login:
```

- ~? to see a list of escape sequences

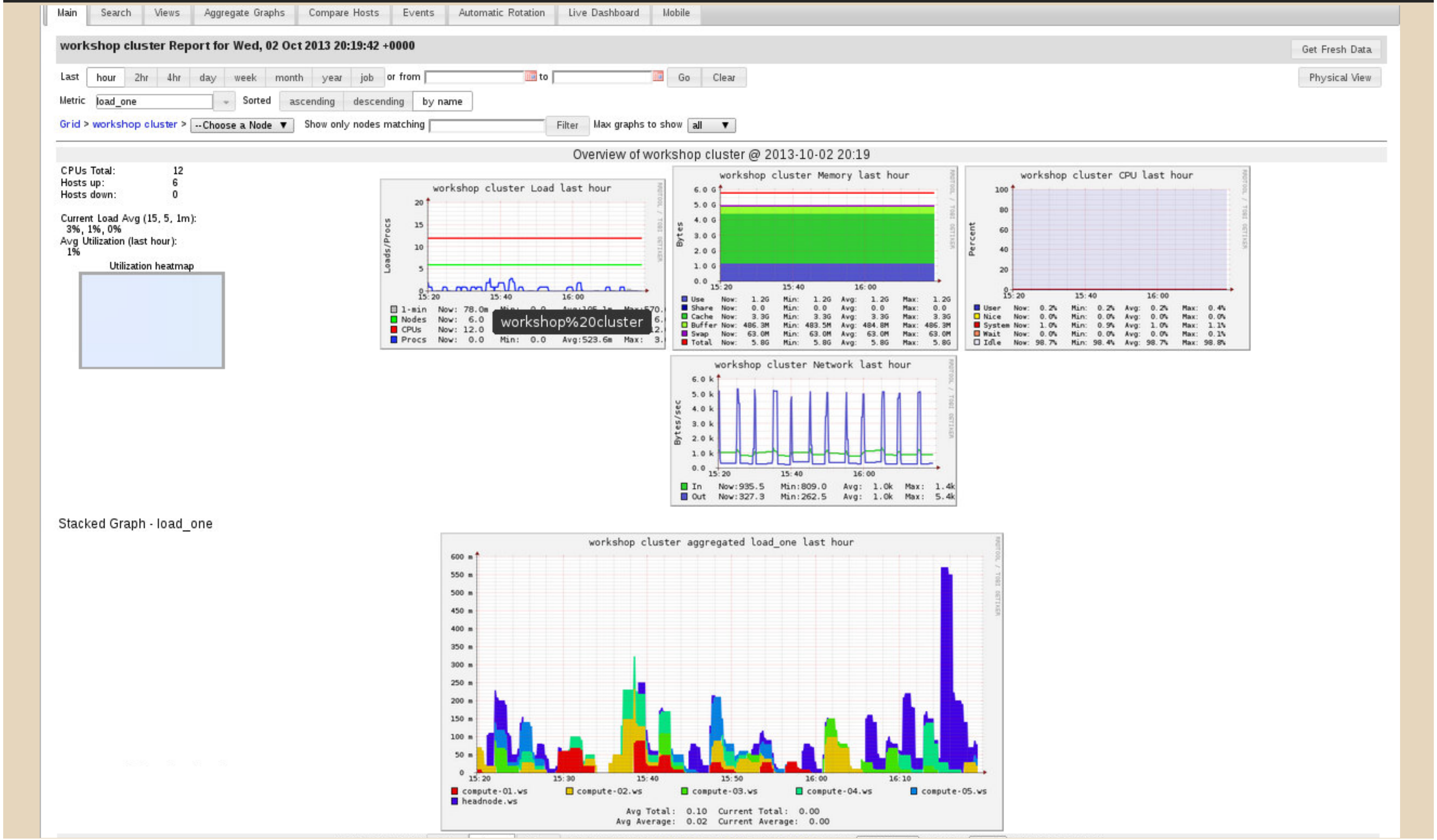
Custer Monitoring

Ganglia

Cluster Monitoring

- A cluster monitoring system can provide an administrator with a high level view of the status of the cluster
- Admin doesn’t need to issue commands or check status of individual machines
- System allows a unified view of..
    - Status of all nodes (up, down, error condition)
    - Cluster load and utilization
    - Utilization of the various networks within the cluster
    - Resource use (memory, swap, disk etc..)
    - Historical data for all monitored variables

Cluster Monitoring



workshop cluster Report for Wed, 02 Oct 2013 20:19:42 +0000

Last: [✓ hour] 2hr 4hr day week month year job or from ____ to ____

Metric: load_one   Sorted: ascending descending [✓ by name]

Grid > workshop cluster > --Choose a Node ▼   Show only nodes matching ____   Max graphs to show: all ▼

Overview of workshop cluster @ 2013-10-02 20:19

CPUs Total:	12
Hosts up:	6
Hosts down:	0

Current Load Avg (15, 5, 1m):
3%, 1%, 0%

Avg Utilization (last hour):
1%

Utilization heatmap

workshop cluster Load last hour

```
□ 1-min   Now: 78.0m  Min:  0.0  Avg:105.1m  Max:570.
■ Nodes   Now:  6.0                                6.
■ CPUs    Now: 12.0                               12.
■ Procs   Now:  0.0   Min:  0.0  Avg:523.6m  Max:  3.
```

workshop%20cluster

workshop cluster Memory last hour

```
■ Use     Now:  1.2G   Min:  1.2G   Avg:  1.2G   Max:  1.2G
■ Share   Now:  0.0    Min:  0.0    Avg:  0.0    Max:  0.0
■ Cache   Now:  3.3G   Min:  3.3G   Avg:  3.3G   Max:  3.3G
■ Buffer  Now: 486.3M  Min: 483.5M  Avg: 484.8M  Max: 486.3M
■ Swap    Now:  63.0M  Min:  63.0M  Avg:  63.0M  Max:  63.0M
■ Total   Now:  5.8G   Min:  5.8G   Avg:  5.8G   Max:  5.8G
```

workshop cluster CPU last hour

```
■ User    Now:  0.2%  Min:  0.2%  Avg:  0.2%  Max:  0.4%
■ Nice    Now:  0.0%  Min:  0.0%  Avg:  0.0%  Max:  0.0%
■ System  Now:  1.0%  Min:  0.9%  Avg:  1.0%  Max:  1.1%
■ Wait    Now:  0.0%  Min:  0.0%  Avg:  0.0%  Max:  0.1%
□ Idle    Now: 98.7%  Min: 98.4%  Avg: 98.7%  Max: 98.8%
```

workshop cluster Network last hour

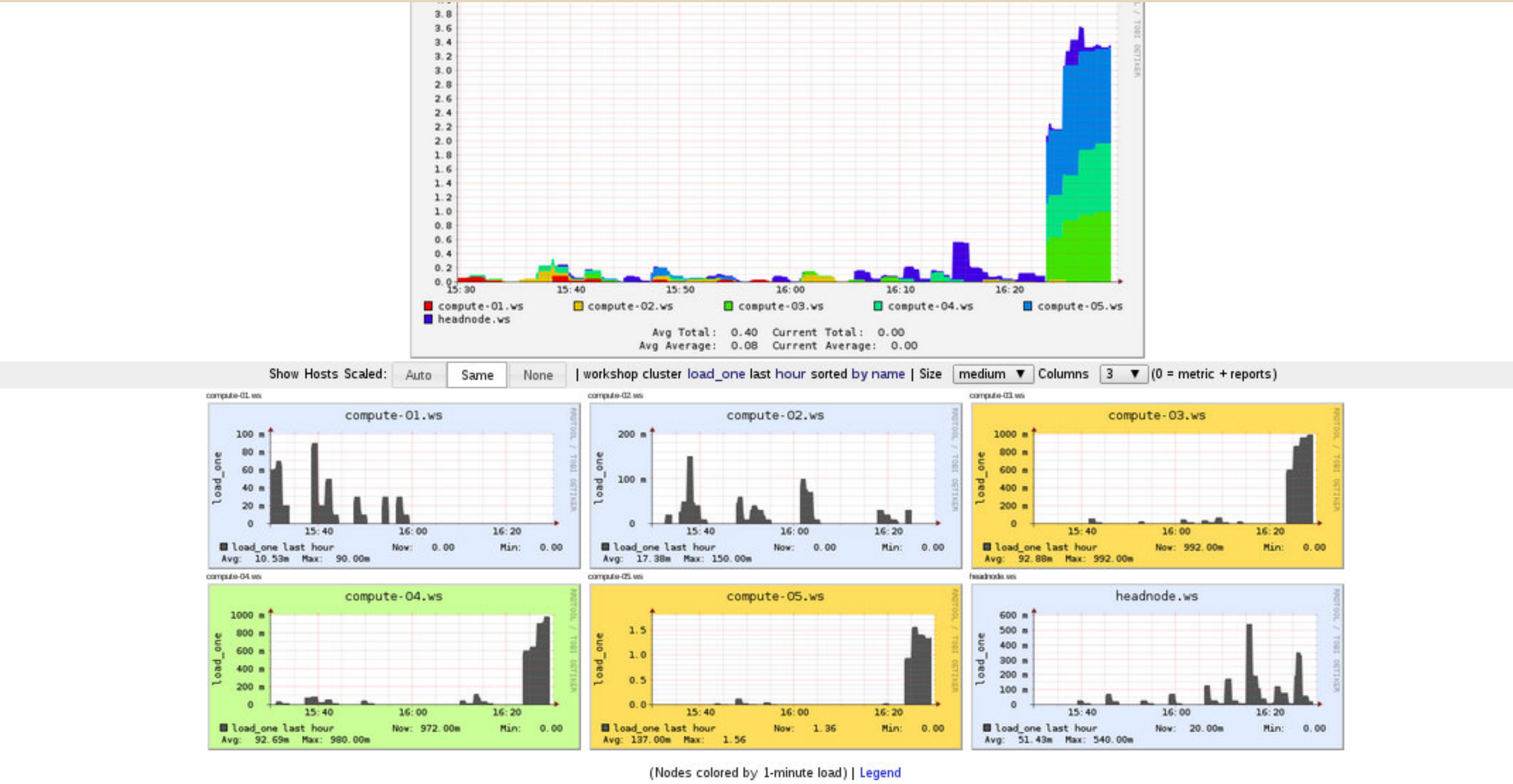
```
■ In   Now:935.5  Min:809.0  Avg:  1.0k  Max:  1.4k
■ Out  Now:327.3  Min:262.5  Avg:  1.0k  Max:  5.4k
```

Stacked Graph - load_one

workshop cluster aggregated load_one last hour

```
■ compute-01.ws   ■ compute-02.ws   ■ compute-03.ws   ■ compute-04.ws   ■ compute-05.ws
■ headnode.ws
          Avg Total:    0.10   Current Total:    0.00
          Avg Average:  0.02   Current Average:  0.00
```

Cluster Monitoring



🟥 compute-01.ws 🟨 compute-02.ws 🟩 compute-03.ws 🟩 compute-04.ws 🟦 compute-05.ws
🟪 headnode.ws

```
Avg Total:    0.40  Current Total:    0.00
Avg Average:  0.08  Current Average:  0.00
```

Show Hosts Scaled: Auto **Same** None | workshop cluster load_one last hour sorted by name | Size medium ▼ Columns 3 ▼ (0 = metric + reports)

compute-01.ws

```
■ load_one last hour      Now:   0.00     Min:   0.00
Avg:  10.53m   Max:  90.00m
```

compute-02.ws

```
■ load_one last hour      Now:   0.00     Min:   0.00
Avg:  17.38m   Max:  150.00m
```

compute-03.ws

```
■ load_one last hour      Now:  992.00m   Min:   0.00
Avg:  92.88m   Max:  992.00m
```

compute-04.ws

```
■ load_one last hour      Now:  972.00m   Min:   0.00
Avg:  92.69m   Max:  980.00m
```

compute-05.ws

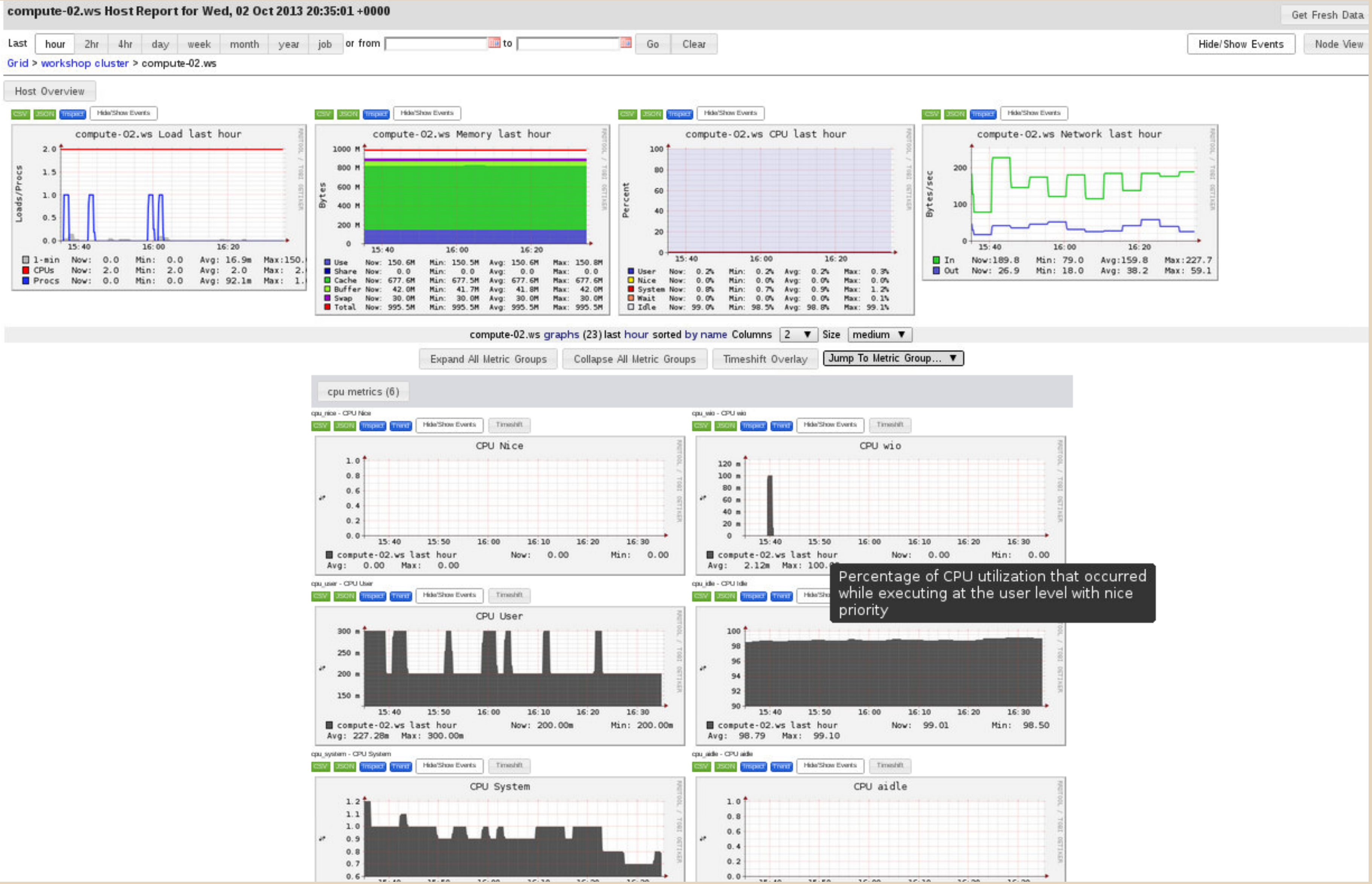
```
■ load_one last hour      Now:   1.36     Min:   0.00
Avg:  137.00m  Max:    1.56
```

headnode.ws

```
■ load_one last hour      Now:  20.00m    Min:   0.00
Avg:  51.43m   Max:  540.00m
```

(Nodes colored by 1-minute load) | Legend

Cluster Monitoring



compute-02.ws Host Report for Wed, 02 Oct 2013 20:35:01 +0000

Grid > workshop cluster > compute-02.ws

compute-02.ws Load last hour

```
■ 1-min   Now:  0.0   Min:  0.0   Avg: 16.9m   Max:150.
■ CPUs    Now:  2.0   Min:  2.0   Avg:  2.0    Max:  2.
■ Procs   Now:  0.0   Min:  0.0   Avg: 92.1m   Max:  1.
```

compute-02.ws Memory last hour

```
■ Use    Now: 150.6M   Min: 150.5M   Avg: 150.6M   Max: 150.8M
■ Share  Now:   0.0    Min:   0.0    Avg:   0.0    Max:   0.0
■ Cache  Now: 677.6M   Min: 677.5M   Avg: 677.6M   Max: 677.6M
■ Buffer Now:  42.0M   Min:  41.7M   Avg:  41.8M   Max:  42.0M
■ Swap   Now:  30.0M   Min:  30.0M   Avg:  30.0M   Max:  30.0M
■ Total  Now: 995.5M   Min: 995.5M   Avg: 995.5M   Max: 995.5M
```

compute-02.ws CPU last hour

```
■ User   Now:  0.2%   Min:  0.2%   Avg:  0.2%   Max:  0.3%
■ Nice   Now:  0.0%   Min:  0.0%   Avg:  0.0%   Max:  0.0%
■ System Now:  0.8%   Min:  0.7%   Avg:  0.9%   Max:  1.2%
■ Wait   Now:  0.0%   Min:  0.0%   Avg:  0.0%   Max:  0.1%
□ Idle   Now: 99.0%   Min: 98.5%   Avg: 98.8%   Max: 99.1%
```

compute-02.ws Network last hour

```
■ In    Now:189.8   Min: 79.0   Avg:159.8   Max:227.7
■ Out   Now: 26.9   Min: 18.0   Avg: 38.2   Max: 59.1
```

compute-02.ws graphs (23) last hour sorted by name Columns 2 ▼ Size medium ▼

cpu_nice - CPU Nice

CPU Nice

```
■ compute-02.ws last hour        Now:    0.00        Min:    0.00
Avg:    0.00    Max:    0.00
```

cpu_wio - CPU wio

CPU wio

```
■ compute-02.ws last hour        Now:    0.00        Min:    0.00
Avg:    2.12m   Max: 100.
```

Percentage of CPU utilization that occurred while executing at the user level with nice priority

cpu_user - CPU User

CPU User

```
■ compute-02.ws last hour        Now: 200.00m        Min: 200.00m
Avg: 227.28m    Max: 300.00m
```

cpu_idle - CPU Idle

```
■ compute-02.ws last hour        Now:   99.01        Min:   98.50
Avg:   98.79    Max:   99.10
```

cpu_system - CPU System

CPU System

cpu_aidle - CPU aidle

CPU aidle

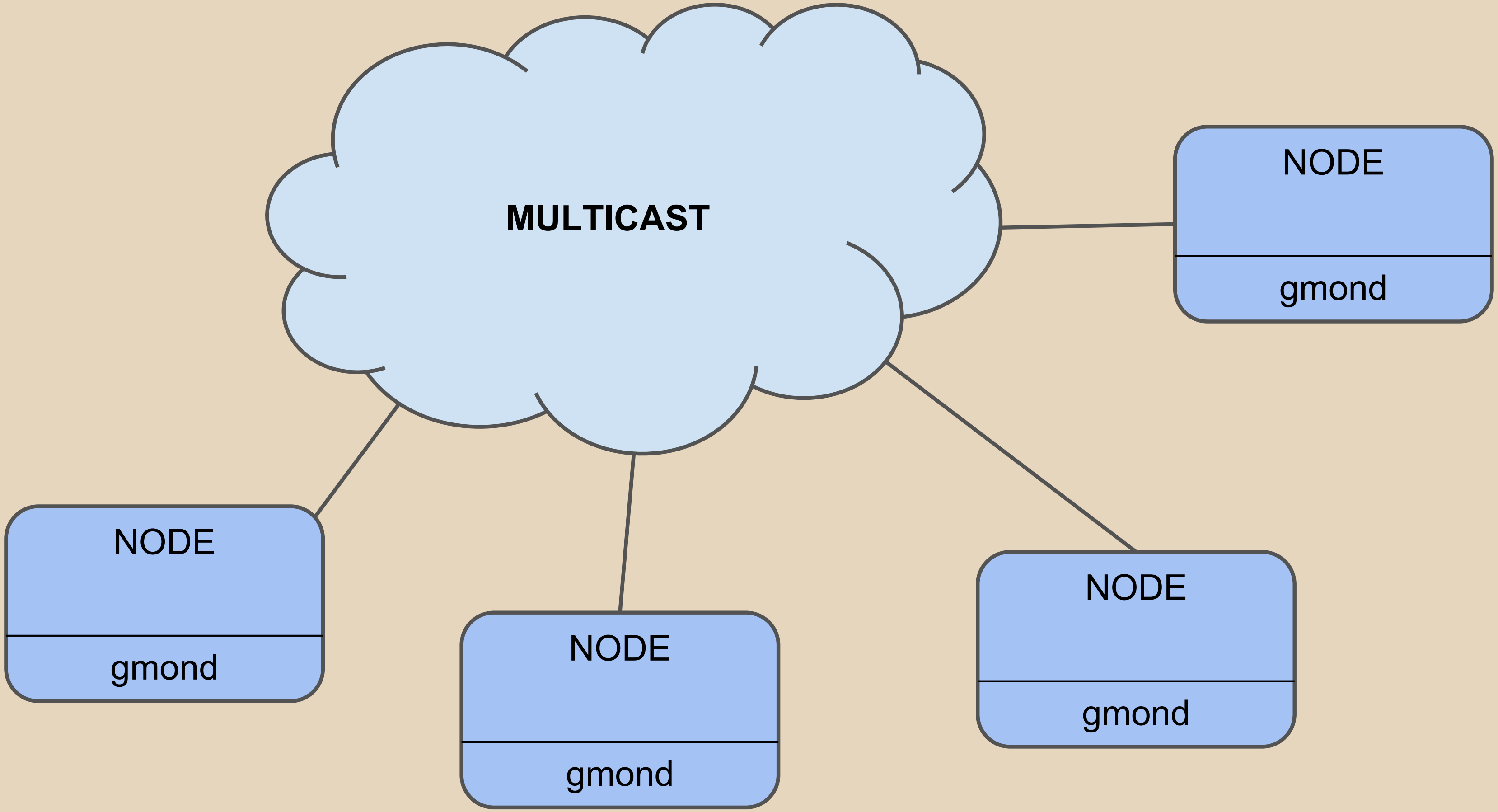
Cluster Monitoring: Ganglia

- Ganglia is open-source and began as a project at University of California, Berkeley
- Designed to scale to a large number of nodes
- Consists of the following main components
  - Data collection daemon (gmond)
  - Data aggregator daemon (gmetad)
  - Web GUI scripts

Ganglia Model

- A `gmond` process runs on every node of a cluster and obtains statistics
- By default all `gmond`’s join the same multicast group
- That is metrics are exchanged among all running `gmond` daemons
- Each `gmond` has complete state information for the entire cluster

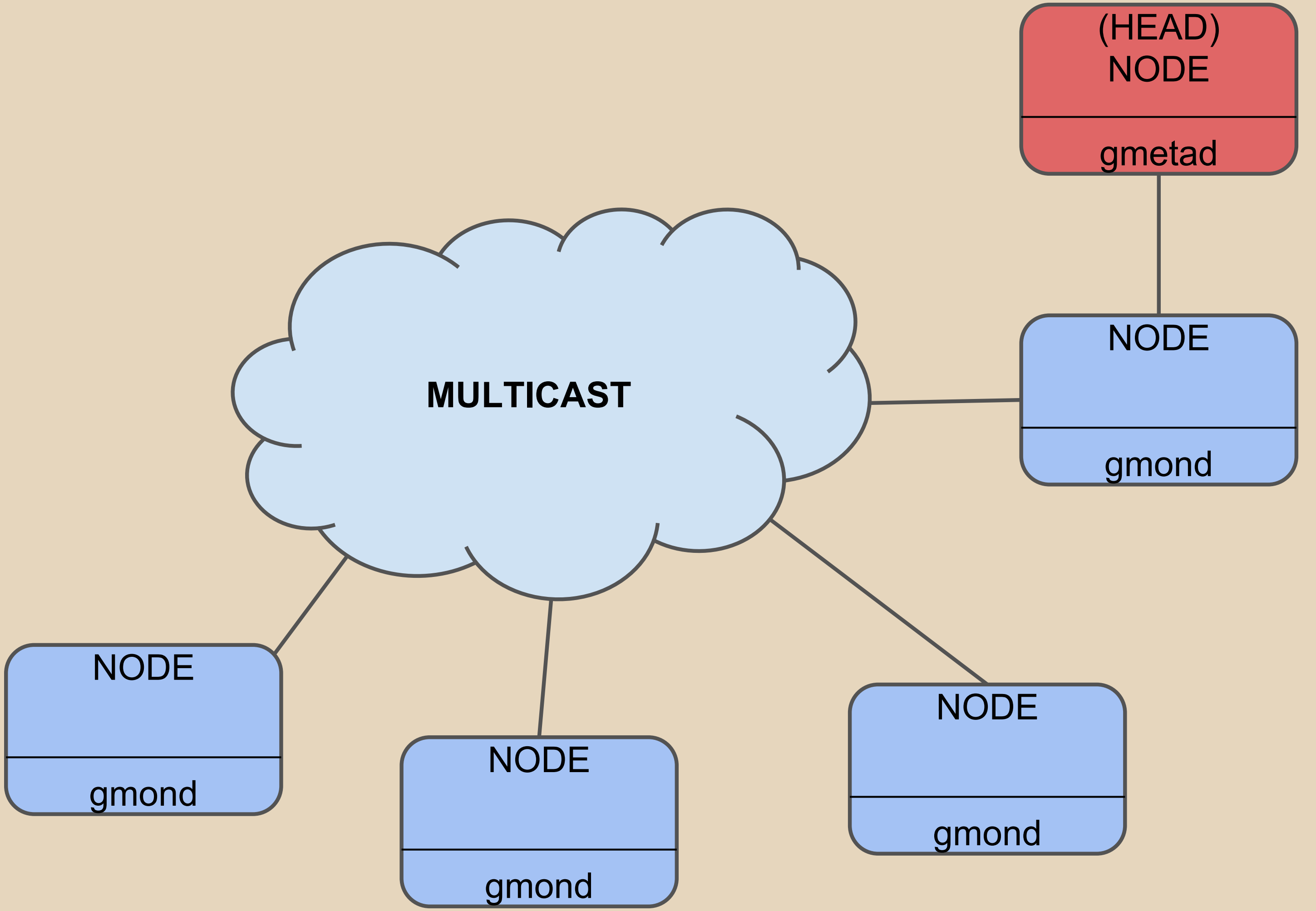
Ganglia Model



Ganglia Model

- For each cluster, a `gmetad` daemon collects and aggregates metrics into a database
- `gmetad` queries one of the `gmond` daemons to collect metrics for the entire cluster
- The `gmetad` daemon can run on a machine external to the cluster. Or on one of the nodes of the cluster (i.e `gmond` and `gmetad` can run on the same machine)

Ganglia Model



RRD Database

- The gmetad daemon logs metrics in RRDtool databases
- RRDtool is specialized system for logging and graphing time series data
- RRD databases (RRD) consist of
  - Data sources (DS)
  - constant storage archives (RRA)
- RRA’s consolidate the data that is collected from each source

RRD Example

```
rrdtool create temperature.rrd --step 300 \
  DS:temp:GAUGE:600:-273:5000 \
  RRA:AVERAGE:0.5:1:1200 \
  RRA:MIN:0.5:12:2400 \
  RRA:MAX:0.5:12:2400 \
  RRA:AVERAGE:0.5:12:2400
```

Data Source:
Data point collected every 300 sec

RRA: 100 hrs of data at 300s resolution

RRA: 300 * 12 * 2400 sec (100 days) of data
at 1 hour resolution. each datapoint is MIN of 1hour data

Same as above. Each datapoint is MAX of 1hr data

Same as above. Each datapoint is AVERAGE of 1hr data

Ganglia Model

- Ganglia provides PHP scripts which are placed in the path of a PHP capable web server (apache2)
- This runs on the same server as gmetad and has access to the RRD database files
- These PHP scripts translate the data into the web interface to the Ganglia monitor

Setting up Ganglia

- Hands-on task
- Install and configure Ganglia from source onto virtual cluster

Batch Processing

Torque Resource Manager

Batch Processor

- HPC systems are expected to provide an environment that
    - Supports multiple (simultaneous) users
    - Supports deferred execution of programs (called a job)
    - Automatically schedules the execution of programs to
        - Ensure that only as many of them execute at a time as for which resources are available (CPUs, memory, etc.)
        - Enforce an administrator defined policy by prioritizing jobs based on certain criteria

Batch Processor

- Batch processing function is provided by a software layer
- Sometimes divided into two interacting components
  - Resource manager
  - Scheduler

Batch Processor

- Resource manager
  - A centralized service that keeps track of the state of resources including
    - Availability: Is it currently busy?
    - Capabilities: For example: how many CPUs are installed, memory available, connection to high speed interconnect
    - Error conditions
  - Also
    - Monitors progress of executing jobs
    - Maintains history of completed jobs
    - Launches jobs on designated resources (remote execution)

Scheduler

- A simple scheduler is the FIFO queue
  - Jobs are launched in the order in which they were submitted
  - Scheduler launches jobs until there are not enough resources. Resumes when resources are available
- Practical schedulers are more sophisticated
  - Allow complex policy based scheduling decisions
  - Fair share scheduling to prevent a user/group from monopolizing resources
  - Run jobs out-of-order to maximize utilization (minimize idle resources)

Some Examples

- Some batch processing systems are
  - Open source:
    - SLURM, PBS/TORQUE (plus Maui), Openlava, Open Grid Scheduler…
  - Commercial
    - Platform LSF, Oracle Grid Engine, Moab...

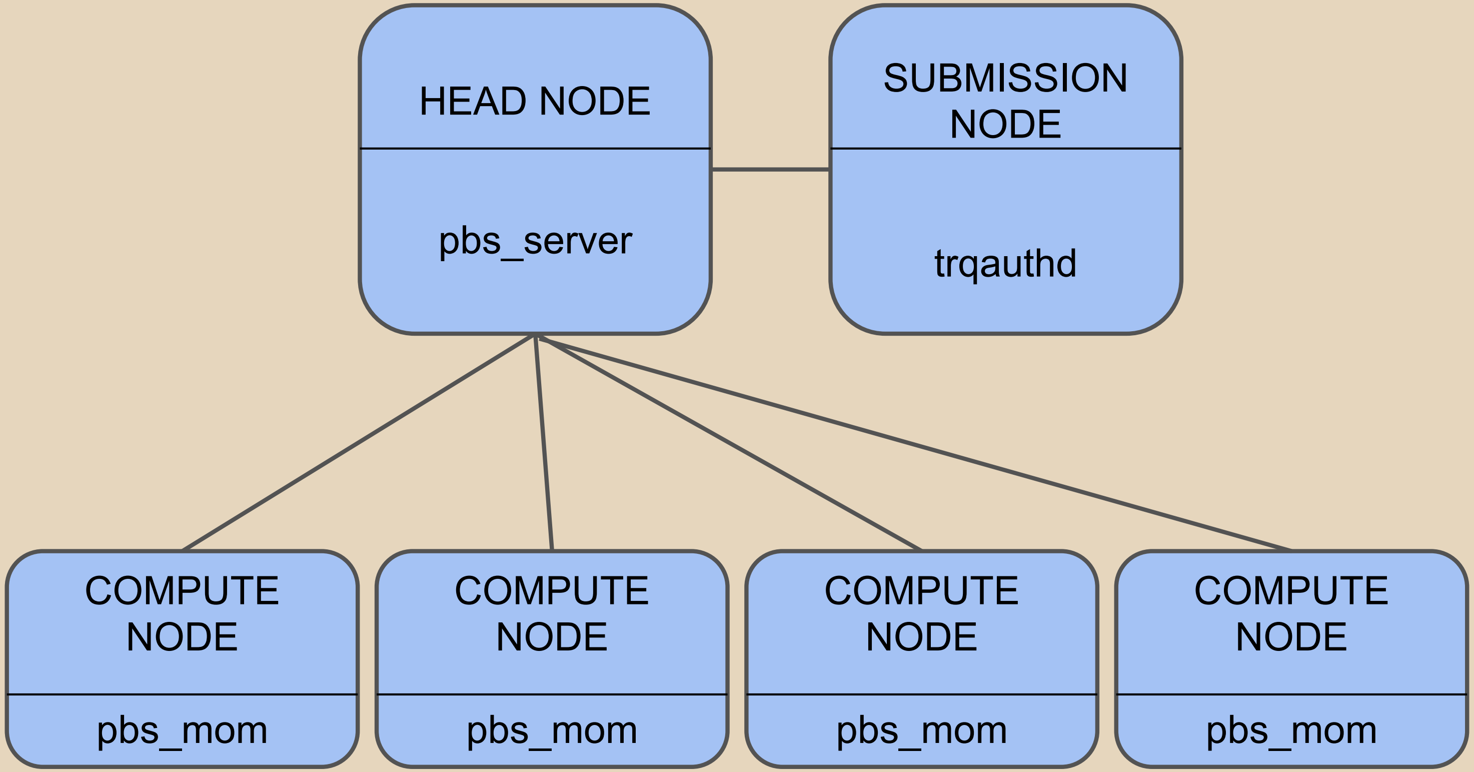
TORQUE resource manager

- TORQUE is and open source project of Adaptive Computing
- It is a fork of the original OpenPBS
- Widely used
- Can be interfaced with schedulers such as Maui

Torque daemons

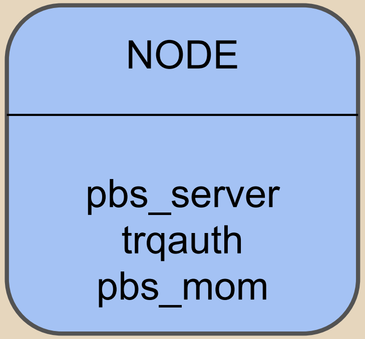
- **`pbs_server`**
  - Main controlling daemon
  - Maintain state information in a database
- **`pbs_mom`**
  - Runs on all compute nodes
  - Communicates nodes status to pbs_server
  - Agent that launches executable on node on behalf of pbs_server and monitors execution
- **`trqauthd`**
  - User authentication daemon
  - Runs on nodes that execute client commands

Torque architecture



Torque architecture

- Daemons need not be on different hosts



Installing Torque

- Configure and build
- Install startup scripts
- Initialize Torque database
- Initialize directories on compute nodes
- Start services

Configuring Torque

- Configure Torque using qmgr
- Add nodes
- Set node properties
- Check nodes-server communication

Installing Torque

- Queues are required to accept jobs
- Multiple queues may be created, each with differing policies
  - Example
  - Queues with different priorities with access granted to specific accounts or groups
  - Queues mapping to different sets of hardware
- Create queues
- Set queue attributes
  - Example: Set maximum allowable job walltime. Shorter time limits improve job turnaround time
  - Therefore better balance between users

Installing Torque

- Torque includes a basic job scheduler.
  - started by executing the `pbs_sched` command which starts the scheduling daemon
- `pbs_sched` has basic support for backfill, fairshare and other scheduling features
- Jobs will not start without scheduling being enabled in the Torque server

```
# qmgr -c 'set server scheduling = True'
```

Torque features

- prologue and epilogue scripts run before and after a job executes respectively
- Scripts must be present in directory

```
$PBS_HOME/mom_priv
```

- On our test server

```
/var/spool/torque/mom_priv
```

- And must be available on all compute nodes

Torque features

- prologue/epilogue scripts are executed on the first node of an allocation as root user
- Script files named `prologue` and `epilogue`

```
/var/spool/torque/mom_priv/prologue
/var/spool/torque/mom_priv/epilogue
```

Torque features

- Prologue scripts can be used to prepare nodes before a job runs
    - Create scratch directories
    - Clear processes that may have been left behind by a previous job
- Epilogue scripts can be used to clean up nodes after a job completes
    - Clear temporary files
    - Kill processes left behind
- Script only runs on first node on allocation
- Therefore script needs to loop over nodes and execute commands remotely with ssh

Submitting a Job

- Create a job script
- As a regular user

```
#!/bin/sh

#PBS -l walltime=0:30:00
#PBS -N mytestjob
#PBS -q batch
#PBS -l nodes=1:ppn=2

cd $PBS_O_WORKDIR

./myexecutable input_$mynum > myoutput
```

Submitting a Job

- `qsub` options are specified as `#PBS` lines
- Commands in the script are executed on the first node in a multi-node allocation

Submitting a Job

- Submit the job

```
$ qsub testjob.sh
```

- Check status

```
$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
17.localhost               mytestjob        ebasheer               0 R batch
```

Submitting an Interactive Job

- Use `-I` option to qsub
- Example

```
$ qsub -I -q batch -lnodes=1:ppn=2
```

- Will allocate a node and launch a shell on the first node of the allocation
- `stdin`, `stdout`, and `stderr` of executable is connected to terminal where `qsub` was run

Submitting an Interactive Job

- Example

```
$ qsub -I -q batch -lnodes=1:ppn=2
qsub: waiting for job 123.headnode.ws to start
qsub: job 123.headnode.ws ready

[testuser@compute-05 ~]$
```

Torque considerations

- Care needs to taken when using programs that launch parallel executables. Like `mpirun`
- Unless integrated with Torque: They launch executables in a way that cannot be reliably controlled / monitored by the resource manager
    - Resource usage accounting may be incorrect
    - Processes may not be reliably killed when a job is terminated

Torque considerations

- Some MPI libraries support Torque integration (example OpenMPI)
- Build time option
- Launches executables using the Torque TM (Task Manager) API

Torque commands

- `pbsnodes`
  - Examine node statistics
  - Set node state
    - Take node offline
- `qsub`
- `qalter`
  - Modify attributes of job in queue
- `qstat`
  - View status of jobs, queues and server
- `pbsdsh`

Batch Processing

Maui Job Scheduler

Maui Job Scheduler

- pbs_sched
  - Allows simple configurations beyond the basic FIFO algorithm. Suffices when
    - Complex policies are not required
    - Resources are not heavily oversubscribed
    - Job resource requirements not very diverse
  - Does not allow fine grained control over the scheduling algorithms
- Torque can interface to a job scheduler
  - Scheduler decides which jobs should run during each scheduling cycle and directs torque based on policy settings

Installing Maui 3.3.1

- Download Maui from Adaptive Computing and extract
- Maui requires one of the source files to be patched to work with Torque 4.x
- Configure and build
  - Very important to pass the `-fno-strict-aliasing` option to gcc compiler
  - Spool directory must be specified where configuration, logs and accounting in maintained
- Edit start-up script templates and install

Maui Definitions

- Job
  - A request for a set of resources with requirements or constraints
  - Example:
    - 5 nodes of 8 cores each
    - nodes must have gigabit connectivity
    - Expected to run for 2 hours
- Advance reservation
  - A list of resources reserved for a particular period of time and attached to a job or user(s)
  - Example
    - Nodes `n1, n2, n4` reserved for job `24` for a duration of `12` hours beginning Sep 19, 11:47

Maui Scheduling Iteration

- Determine if job is eligible for consideration in scheduling decision
- Priority assigned for each job based on several weighted factors. time in queue, amount of resources requested, owner, ...
- Begin with highest priority jobs
- Determine if resources are available and place a reservation
- Communicate to resource manager to start jobs on reserved resources

Configuring Maui

- `/var/spool/maui/maui.cfg`
- `PARAMETER VALUE` pairs
- System wide parameters
  - `RMCFG[name]`
    - Specify resource manager
  - `SERVERHOST`
    - Host where maui daemon runs
    - used by clients to locate server daemon
  - `ADMIN1 user1 user2 ...`
    - user1 is user as which maui should be run
    - `ADMIN1` users must be Torque server managers and operators

Configuring Maui

- Some parameters
  - `SERVERMODE`
    - `TEST` allows to evaluate maui behavior without affecting the active scheduler
    - `NORMAL` for live scheduling

Configuring Scheduling

- Maui dynamically decides job priority based on based on job factors
- Scheduling can be tuned by assigning weights to components and subcomponents
- Priority change calculation usually of form

```
      CW * (SCW1*val1 + SCW2*val2 + …)
where,
CW -> component weight
SCW -> subcomponent weight
```

Configuring Maui

- If no weights are specified
- Only subcomponent with non-zero weight is `QUEUETIMEWEIGHT`
- Effectively makes Maui behave as a FIFO scheduler with backfill enabled

Configuring Scheduling Fairshare

- Fairshare takes into account usage history for a user/group for priority calculation
- For example: Jobs of a user who had run more jobs in the past weeks gets a lower priority than jobs of one who hasn’t run any in the same period
- Balances out usage: avoids monopolization
- Different user/groups can be assigned varying weights in the fairshare.
  - Allows utilization to be steered towards defined targets

Configuring Scheduling Fairshare Global Settings

`FSPOLICY DEDICATEDPES`

- Base usage history on utilized, processor-equivalents. Roughly, the number of processors held up due to a job

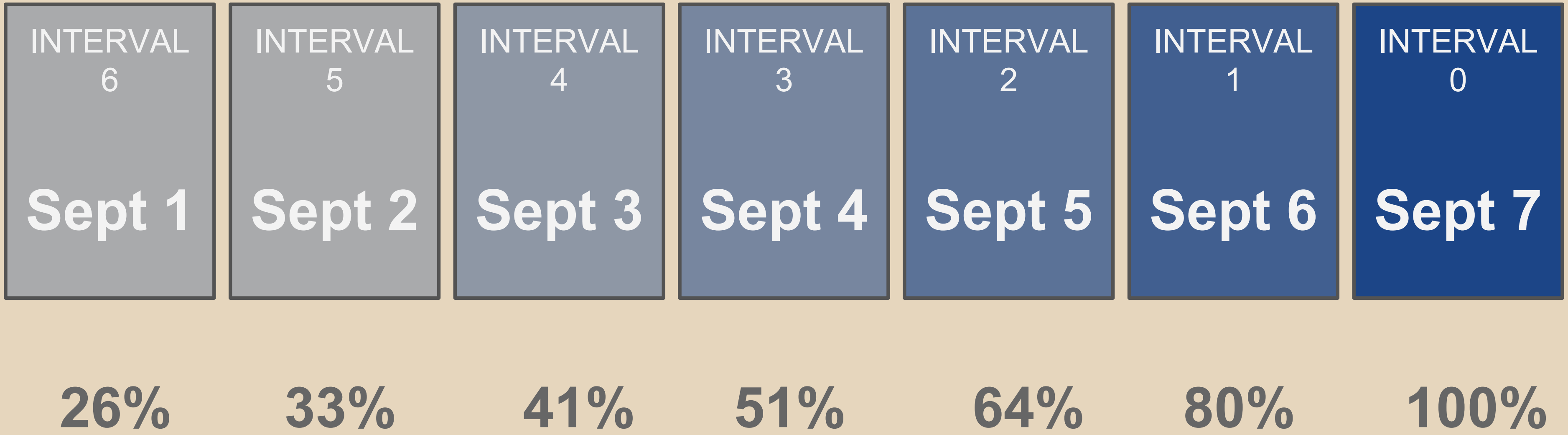
`FSINTERVAL`

`FSDEPTH`

`FSDECAY`

- Maui keeps track of historical usage for a duration of `FSINTERVAL * FSDEPTH` with a decay of `FSDECAY ^ N` for interval N

Configuring Scheduling Fairshare Global Settings



INTERVAL 6	INTERVAL 5	INTERVAL 4	INTERVAL 3	INTERVAL 2	INTERVAL 1	INTERVAL 0
Sept 1	Sept 2	Sept 3	Sept 4	Sept 5	Sept 6	Sept 7
26%	33%	41%	51%	64%	80%	100%

- With `FSDEPTH=7 FSINTERVAL=86400 FSDECAY=0.80`
- Fairshare history maintained for 7 24-hour periods
- Oldest interval contributes only 26% to the current fairshare usage calculation
- Latest interval contributes 100% to the calculation

Configuring Scheduling Fairshare

- Priority change calculation

```
FSWEIGHT * (FSUSERWEIGHT * user-usage +
            FSGROUPWEIGHT * group-usage +
            ...)
```

where

`user-usage:` Is difference between real usage and target set for that user

group-usage: same as above for groups

Configuring Scheduling Fairshare

- FS target is percentage system utilization
- Target per user or group can be set

```
USERCFG[john]   FSTARGET=10.0
GROUPCFG[staff] FSTARGET=50.0
```

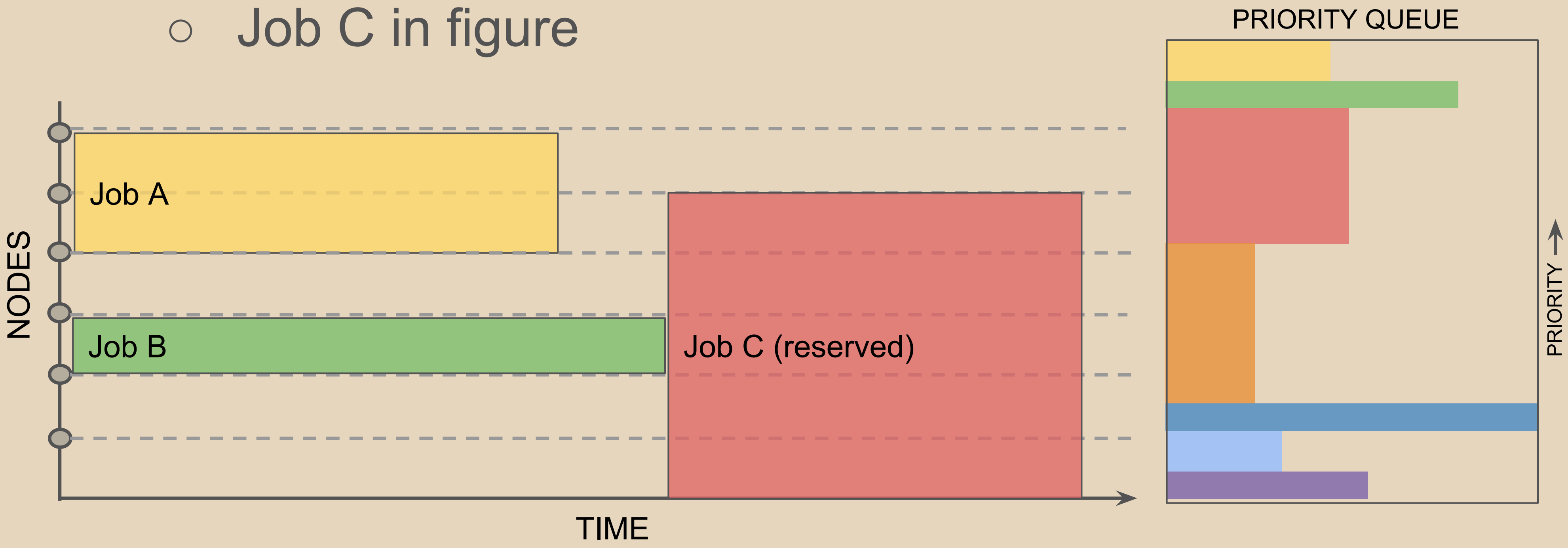
Configuring Scheduling Prioritization by Job Size

- Job priority can be weighted by resources requested by job
- Priority change calculation

```
RESWEIGHT * (NODEWEIGHT * node-reqest +
             PROCWEIGHT * proc-request +
            ...)
```

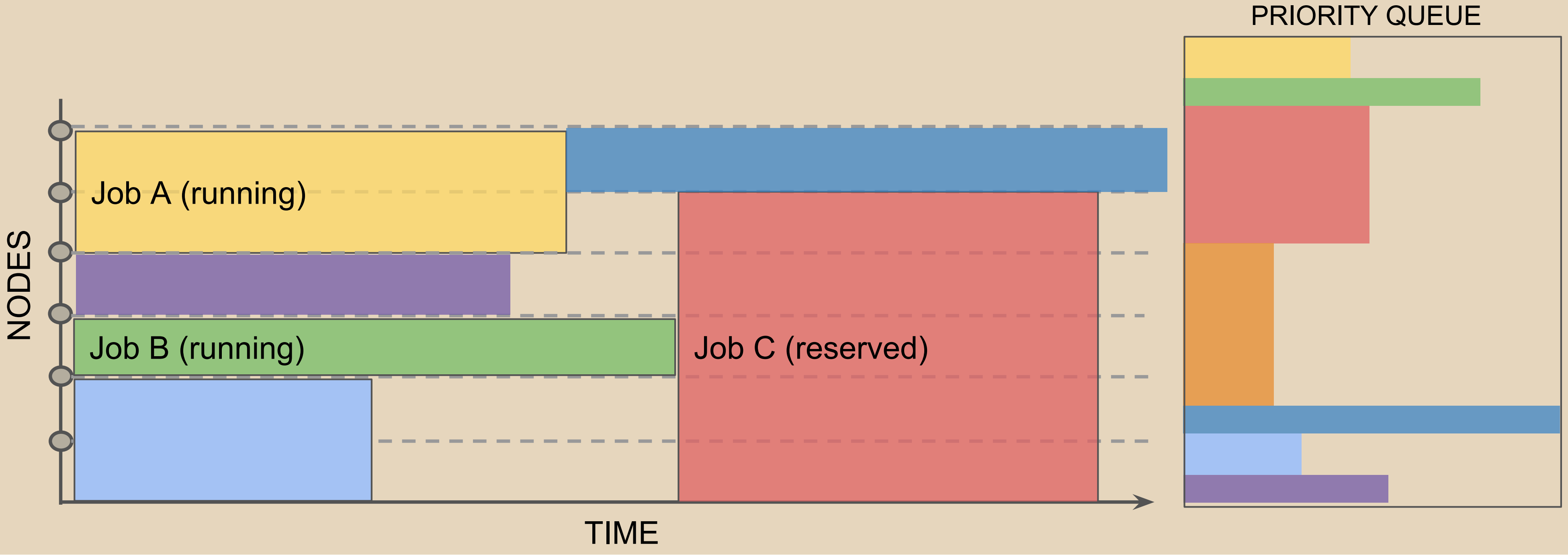
Configuring Scheduling Backfill

- Maui determines earliest that highest priority job can start based on walltime limits and resources requested
- Places a reservation for that job
    - Job C in figure



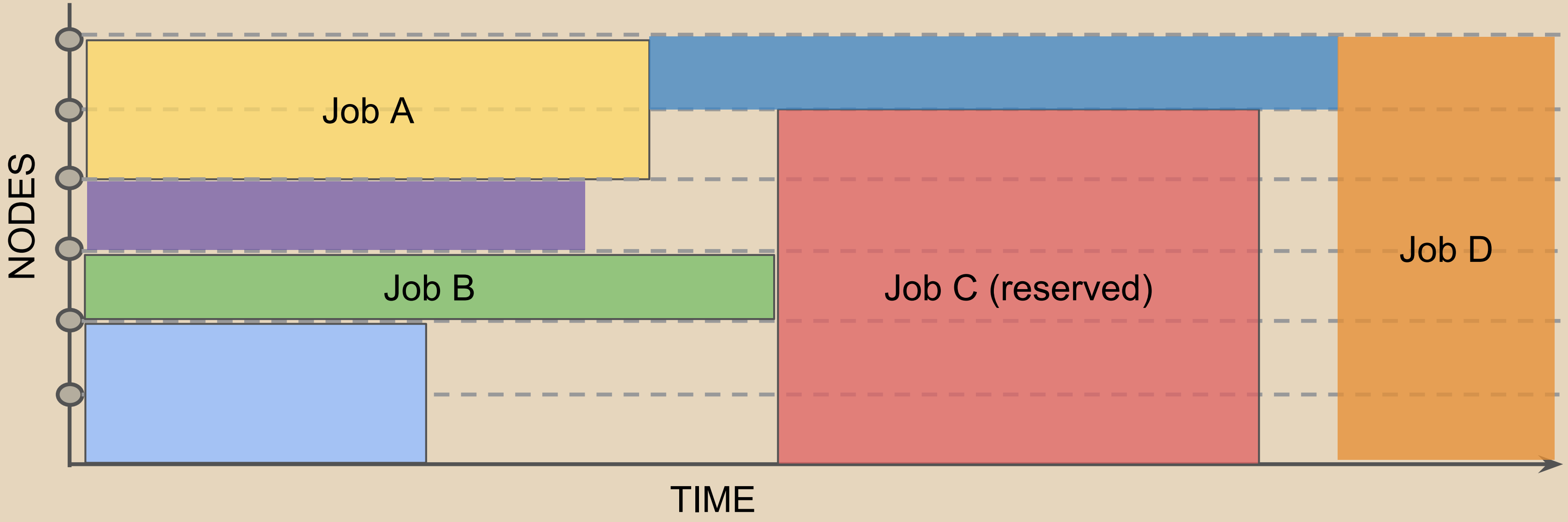
Configuring Scheduling Backfill

- Backfill enables Maui to consider lower priority jobs
    - Without affecting the reservation for the highest priority job



Configuring Scheduling Backfill

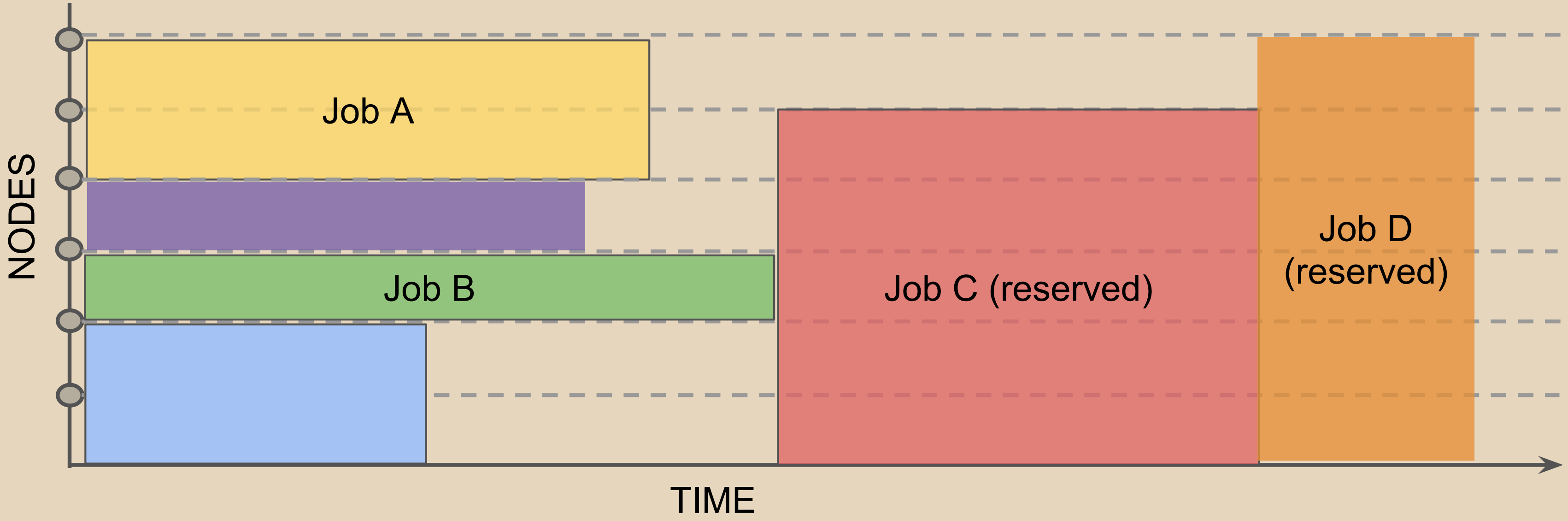
- Can sometimes cause high priority jobs for which there is no reservation to be delayed
  - Job D in figure



Configuring Scheduling Backfill

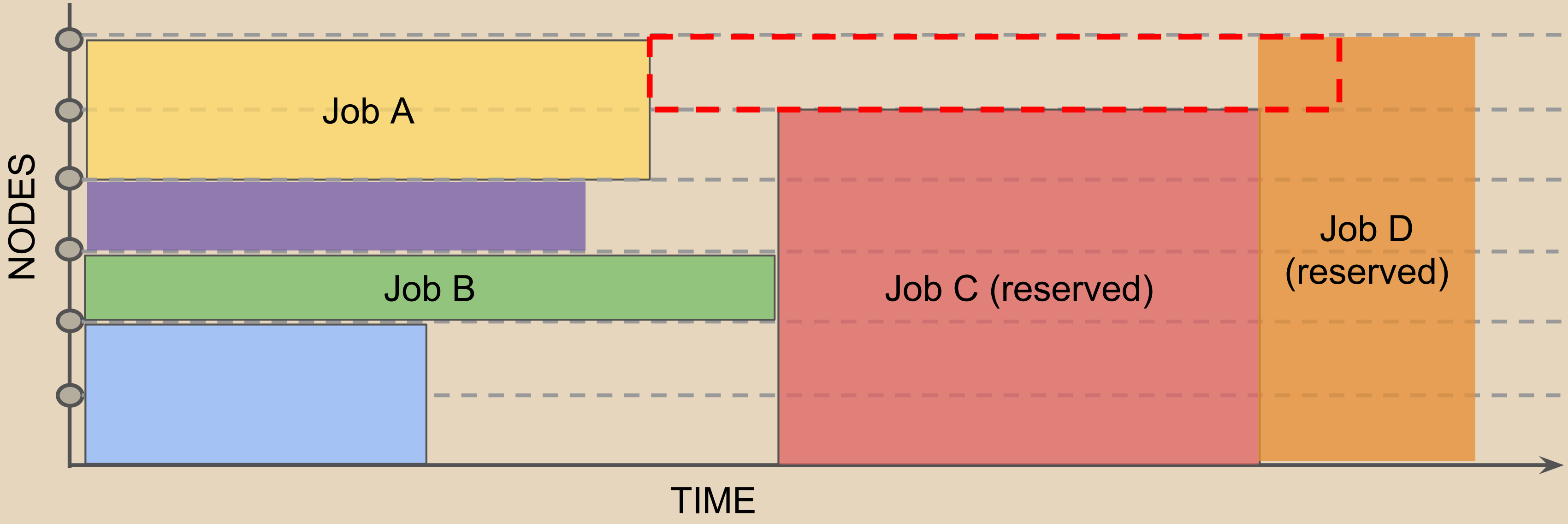
- Depth in the priority queue to make reservations can be set (default is 1)
- Example

```
RESERVATIONDEPTH 2
```



Configuring Scheduling Backfill

- But, may lower utilization by reducing backfill aggressiveness



Maui Commands

- `checknode`
  - Information about nodes
  - Jobs running, reservations
- `checkjob`
- `showstats`
- `mdiag`
  - Diagnostics
- `showq`
- `setres, releaseres`
  - Place administrative reservations
- `showbf`

Configuring Scheduling Job-Exclusive Access

- By default Maui allows multiple jobs to launch on a node
  - When each of the jobs requests less than the available resources on the node
  - Example: A two jobs with `nodes=1:ppn=1` can launch on the same node that has `np=2`
- Default interpretation of PBS specification of `nodes=<x>:ppn=<y>` is
  - Allocate `<x>*<y>` tasks with at-least `<y>` tasks per node

Configuring Scheduling
Job-Exclusive Access

- The above behavior means that nodes are jobs are not given exclusive access to nodes by default
- On multi-core machines, it is generally best to allow exclusive access to nodes
    - That is only one job can run on a node regardless of the resources requested/utilized
    - Multiple jobs on a multi-core node may cause contention for resources and reduce performance
    - This also allows nodes to be correctly allocated for single processes that spawn multiple threads

Configuring Scheduling Job-Exclusive Access

- Maui can be configured to allocate exactly <x> nodes with <y> tasks per node
- Set

```
JOBNODEMATCHPOLICY    EXACTNODE
```

- And prevent multiple jobs from launching on a single node

```
NODEACCESSPOLICY   SINGLEJOB
```

Configuring Scheduling Shared Access

- Shared node access is the default
- Scheduling can be tuned to evaluate node availability based on actual memory utilization
  - Prevent jobs from being launched if
    available memory < requested memory
    Even if processors are available
  - Caveat is that currently, Torque cannot enforce resident memory limits, but only virtual memory
  - Limiting virtual memory can cause programs to be terminated even though there is sufficient physical memory available

Advance Reservations

- Resources can be reserved for a period of time
    - To make available to certain users, or..
    - To prevent jobs from running on them, so that maintenance can be carried out
    - A “system reservation” can be made by the admin to keep the entire system free of jobs for a period of maintenance
- Advantages
    - Does not require stopping of job submission
    - Jobs held in queue
    - Jobs can be backfilled until reservation begins

Power and Cooling

Uninterruptible Power Supplies

Uninterruptible Power Supplies

- Two broad scenarios
- Site with generally reliable power supply
    - Such sites can afford to maintain backup only for headnodes and storage nodes
    - Compute nodes don’t require backup
    - Objective of maintaining data integrity
- Site with an unreliable primary supply and backup supply such as generators
    - Provide backup during the switch-over phase
    - Condition power to protect equipment from overvoltages, brown-outs, phase-reversals etc.
    - Backup all IT equipment including network infrastructure

Data Center Power and Cooling

- UPS sizing
    - Determine your IT equipment load that needs to be protected in **Watts**
    - Manufacturers may provide data sheets that estimate power consumption
        - Actual consumption depends whether node is Idle or under heavy computational load
        - Ratings stamped on power supplies show the maximum load that *it* can handle
            - Not useful for determining server load
        - Power factor of the server power supply needs to be noted
        - Power data can also be collected through IPMI or

Data Center Power and Cooling

- UPS sizing
  - UPS capacities usually sized in kVA
  - Server load given in Watts can be converted to VA with

$$VA = Watts / powerfactor$$

  - Rule of thumb: UPSes should not operate at more that 80% capacity. Inefficient
  - UPS backup time depends on size of battery packs
  - UPS manufacturer quotes backup time for particular load

Data Center Power and Cooling

- UPS considerations
  - For UPS capacity plan for future expansion of IT equipment
  - Cooling equipment do not usually have back up power
    - Servers may shut down before battery runs out
  - Time for backup power supply generators must be taken into account
  - Or, time for a graceful shutdown
  - UPS batteries wear out reducing the backup time.

    This much be taken into account with periodic inspection and replacement of batteries

Data Center Power and Cooling

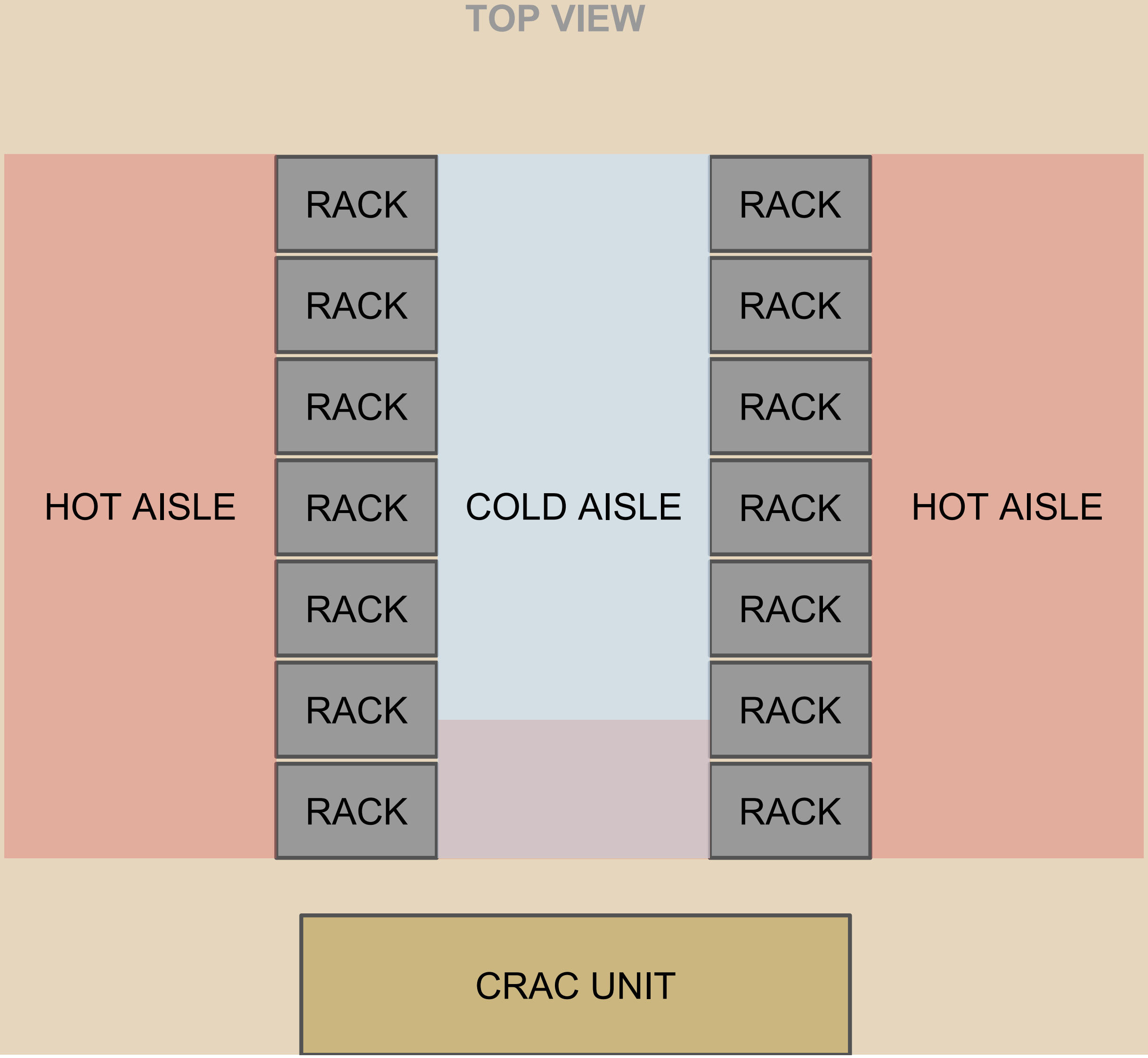
- Cooling
  - Cooling capacity for air conditioners (called CRACs) usually measured in BTU/hr
    - Tons commonly used in some countries
  - Manufacturers usually specify thermal load for servers in BTU/hr
    - Or can be calculated with

BTU/hr = load in Watts **x** 3.41

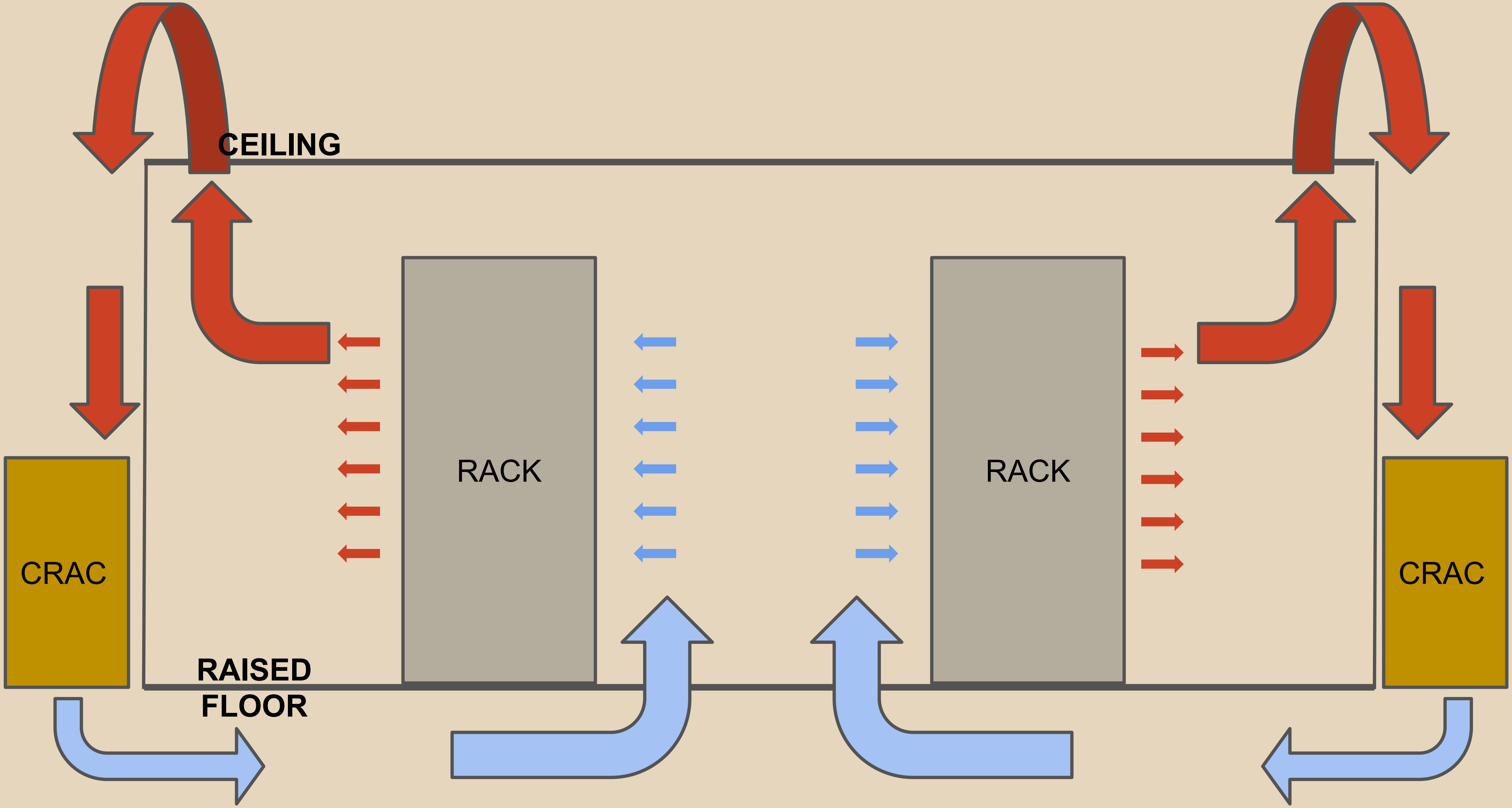
Data Center Power and Cooling

- Rack mount servers have front to back airflow facilitated by fans
- Common datacenter arrangement is Hot aisle/cold aisle separation
- Cold air blown through perforated tiles in raised floor
- Rows of racks with
    - Front of racks facing each other, enclosing cold air in the aisle
    - Back of racks facing each other and enclosing hot server exhaust

Data Center Power and Cooling



Data Center Power and Cooling



Data Center Power and Cooling

- Considerations
  - Too much air velocity from floor tiles may cause air to blow past the servers and directly into hot air ducts
    - Tiles with lesser perforations in these cases to slow down air flow
    - Tiles adjusted according to cooling load required in that vicinity
  - Racks near the CRAC units could be prone to low pressure of cold air due to the high velocity of the air close to the CRAC outlet.

Data Center Power and Cooling

- Considerations
  - Too much air velocity from floor tiles may cause air to blow past the servers and directly into hot air ducts
    - Tiles with lesser perforations in these cases to slow down air flow
    - Tiles adjusted according to cooling load required in that vicinity
  - Prevent mixing of air in cold/hot aisle
    - No gaps must be left in between racks in the rows
    - Unpopulated parts of racks must be closed off

Data Center Power and Cooling

- Considerations
  - Depending on capability of cooling systems and power dissipation
    - Rack may need to to filled less densely
    - Example: Every other position left unpopulated
- Raised floor height depends on thermal load and the rate at which cold air needs to be moved