LoadLeveler Overview

Pidad D'Souza (pidsouza@in.ibm.com)
IBM, System & Technology Group

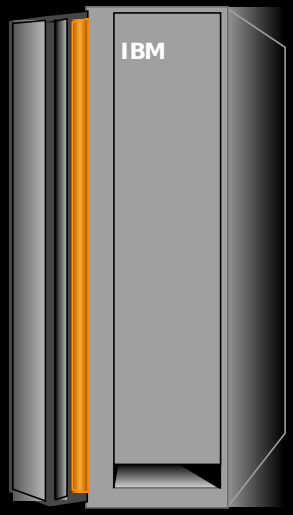
Who Needs a Job Scheduler?

Single Machine

Job 1
Job 2
….
Job N

OS multi-tasks single CPU:
time-shared scheduling

HPC Machine



Parallel Dimension

User 1:	User 2:	User 3:
Job 1	Job 1	Job 1
Job 2	Job 2	Job 2
….	….	….
Job N	Job N	Job N

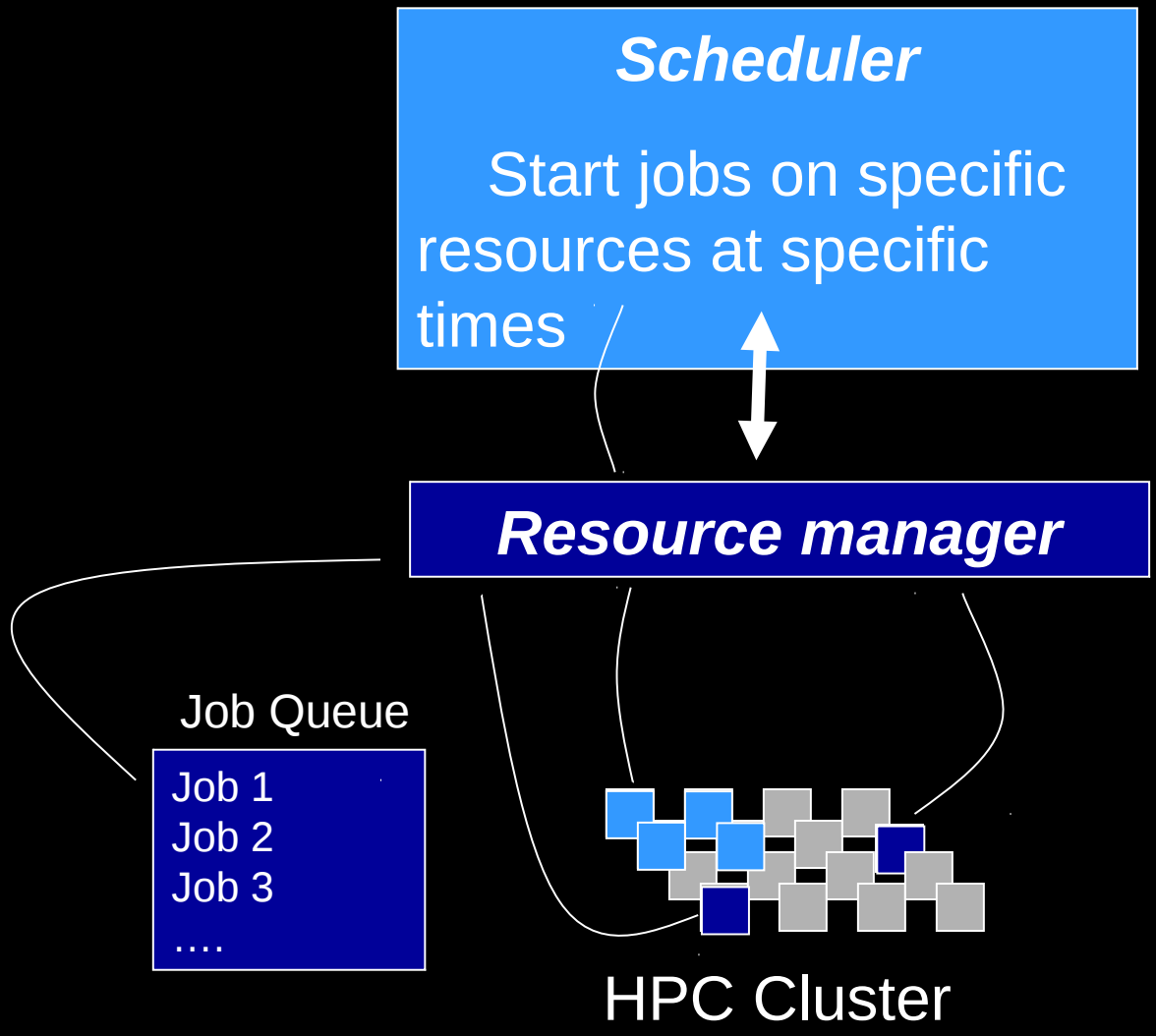
Many Machines and Users:
More Jobs
Parallel Dimension

User may impact a distant job

Scheduler runs jobs according to:
Scheduling Theory
Site-defined Policy

© 2012 IBM Corporation

Scheduling Terms



Batch Scheduler

© 2012 IBM Corporation

LoadLeveler Workload Management

Job Management

Build, Submit, Schedule, Monitor

Change Priority

Terminate

Workload Balancing

Maximize resource use

Control

Centralized - system admin
Individual - user

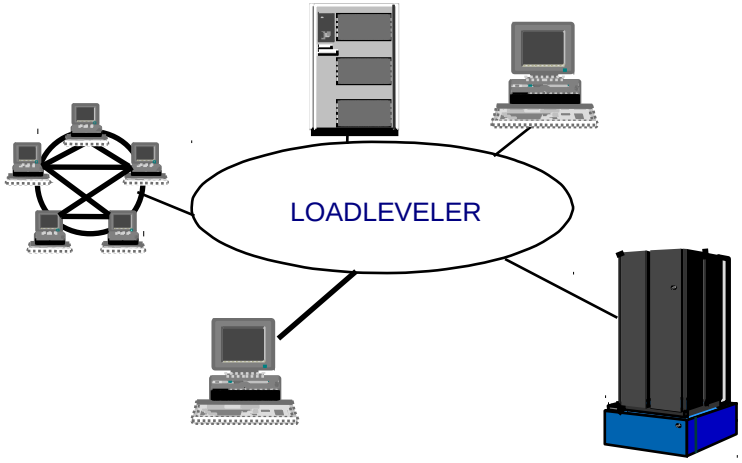
Usability

Web UI

Command line interface

Supports NFS, DFS, AFS, and GPFS

Consolidated Job Accounting



The Right Jobs Go To The Right Nodes

© 2012 IBM Corporation

Supported Hardware

- IBM System p7™
- IBM System p6™
- IBM System p5™
- IBM eServer pSeries®
- IBM BladeCenter POWER-based servers
- IBM Cluster 1600
- IBM OpenPower™

- IBM System servers with AMD Opteron or Intel® EM64T processors
- IBM System x™ servers
- IBM BladeCenter® Intel processor-based servers
- IBM Cluster 1350™

- Servers with Intel 32-bit and Intel Extended Memory 64 Technology (EM64T) –
- Servers with Advanced Micro Devices (AMD) 64-bit technology

- BlueGene

© 2012 IBM Corporation

Supported O/S

- AIX 5L, Version 3 Release 4
- AIX 6.1, AIX 7.1
- Red Hat Enterprise Linux (RHEL) 5 and RHEL 4 on IA-32 servers
- RHEL 5 and RHEL 4 on AMD Opteron or Intel EM64T processors
- RHEL 6 and RHEL 5 on IBM POWER servers
- SUSE Linux Enterprise Server (SLES) 11 and SLES 10 on IA-32 servers
- SLES 11 and SLES 10 on IBM POWER servers
- SLES 11 and SLES 10 on AMD Opteron or Intel EM64T processors

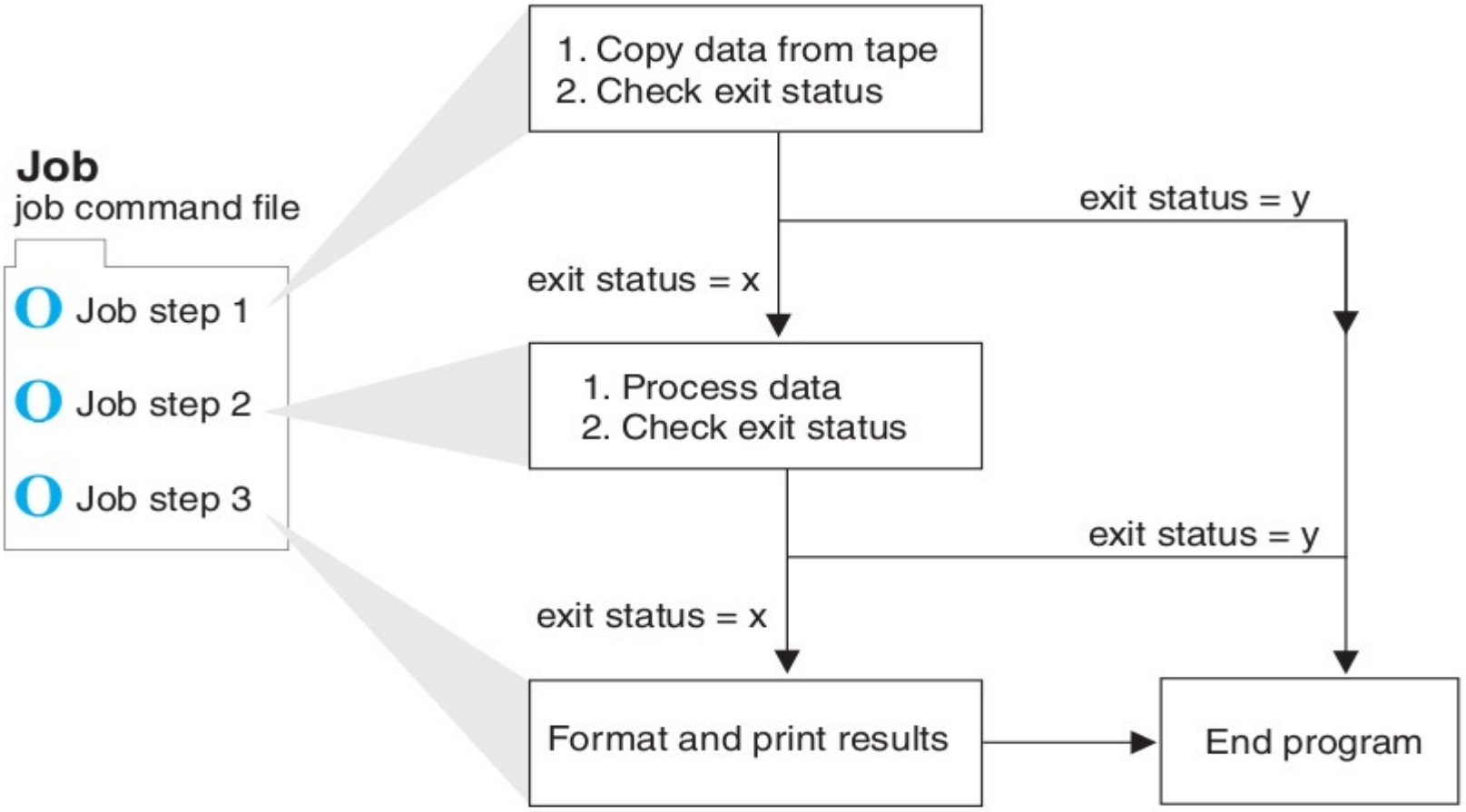
© 2012 IBM Corporation

LoadLeveler External

- Control Files
  - Configuration Files or database
  - Administration Files or database
  - Job Command Files
- Command Line Interface
  - Basic Job Functions (submit, query, cancel, etc)
  - Administrative Functions
- Graphical Interface
  - Web-based user interface (sample only)
- Application Programming Interface
  - Allow application programs to be written by users and administrators to interact with the LoadLeveler environment

© 2012 IBM Corporation

Job Definition



© 2012 IBM Corporation

Job Definition

For example, Figure illustrates a stream of job steps:
1. Copy data from tape
2. Check exit status
Job
exit status = y
job command file
Q Job step 1
exit status = x
Q Job step 2
1. Process data
2. Check exit status
Q Job step 3
exit status = y
exit status = x
Format and print results
End program

© 2012 IBM Corporation

Job Definition – Role of machines

Job

Set of job steps

Job Steps

Each job step can specify different executables

Job steps can be serial or parallel

Job Command file

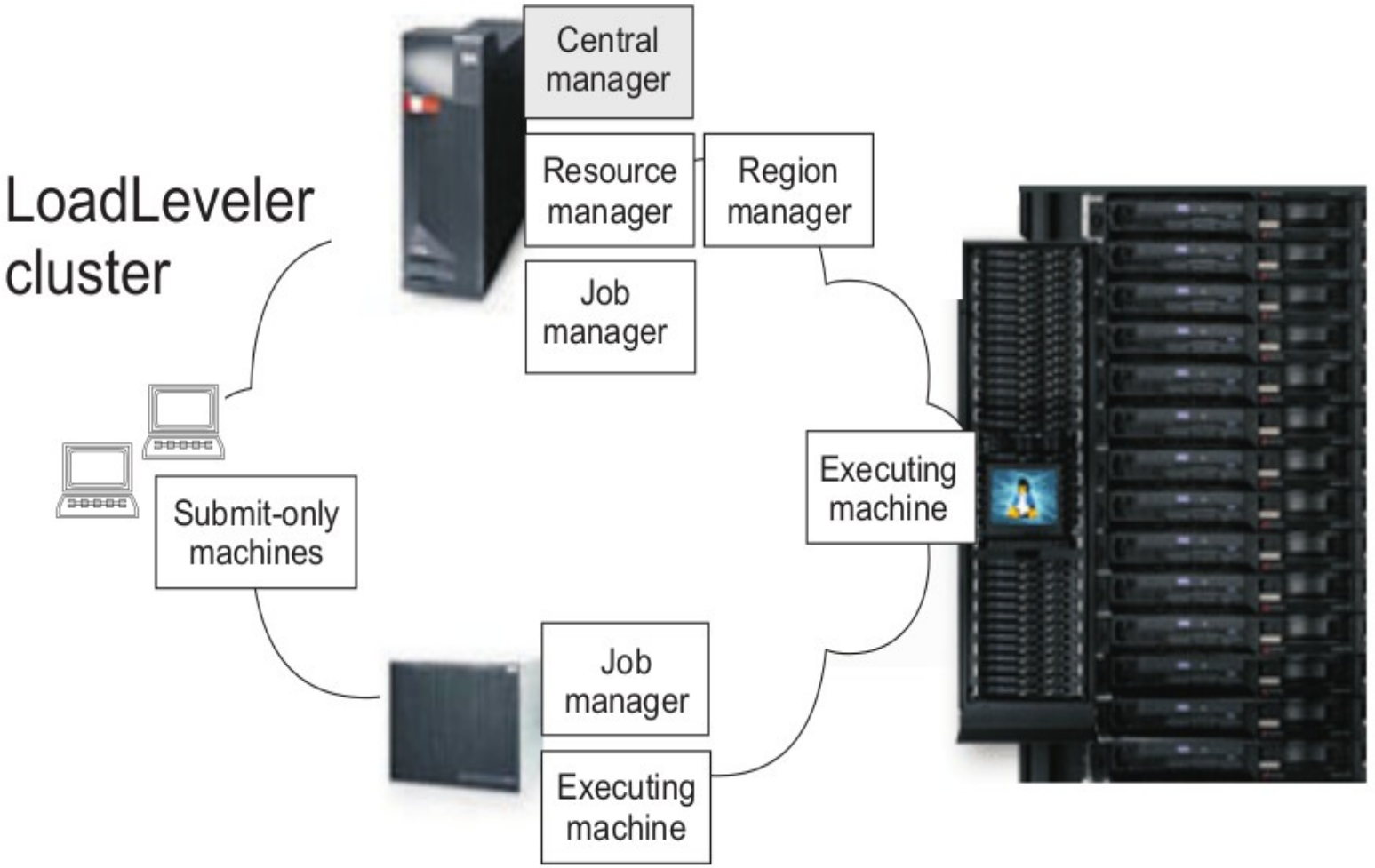
Job steps are defined

Can have one or more job steps

All job steps not necessary to run on same machine

© 2012 IBM Corporation

Machine Definition



© 2012 IBM Corporation

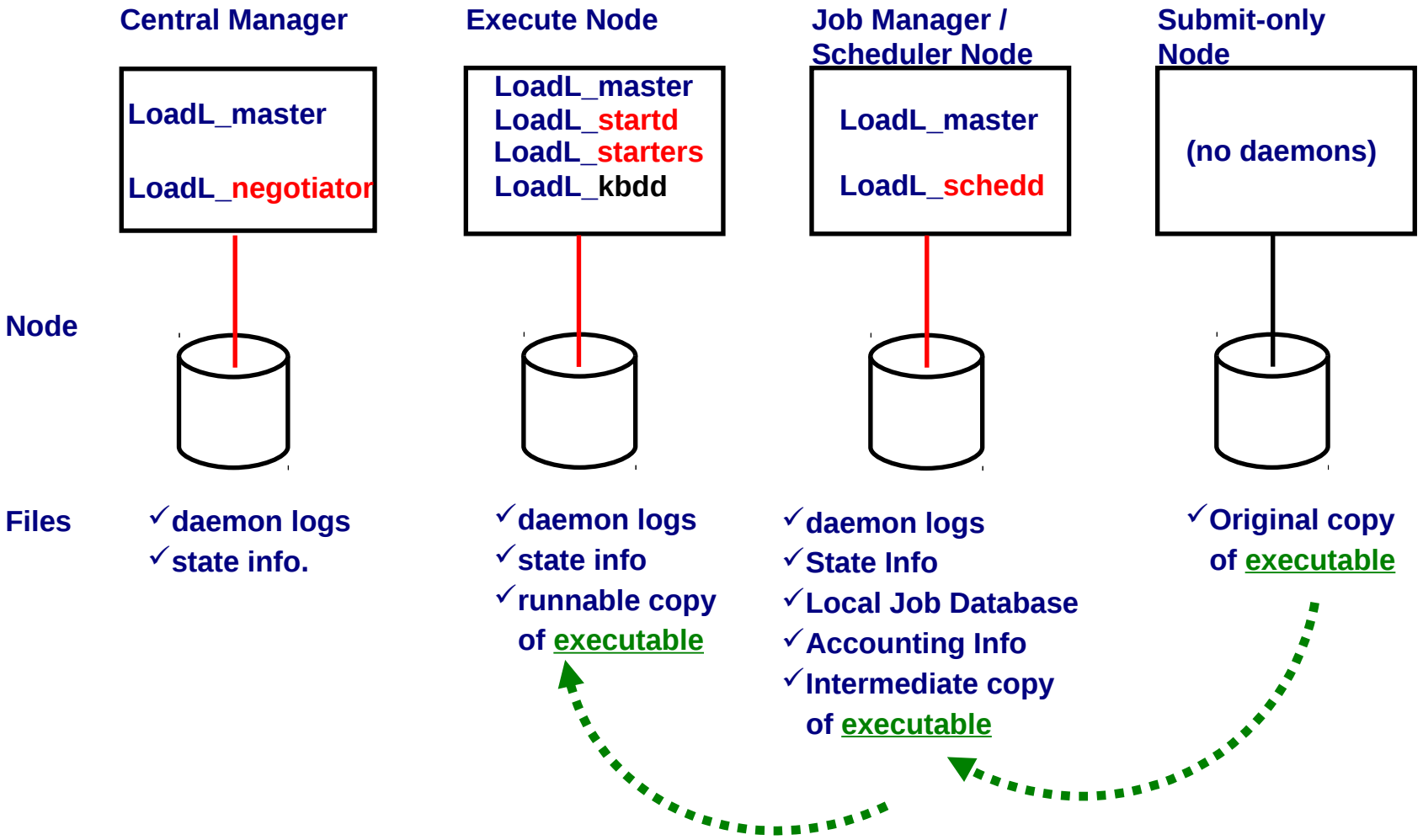
Machine Definition – Role of machines

LoadLeveler Cluster

1. Job Manager/Scheduler Node (public or local)
   Manages jobs from submission through completion
   Receive submission from user, send to Central Manager, schedule jobs
2. Central Manager
   Central resource manager and workload balancer
   Examine requirement and find the resources
3. Execute Node
   Runs work (serial job steps or parallel job tasks) dispatched by the Central Manager
4. *Resource Manager*
   Collect status from executing and job manager
5. *Region Manager*
   Monitor node and adaptor status of executing machines
6. Submit-only Node
   Submits jobs to LoadLeveler from outside the cluster.
   Runs no daemons.

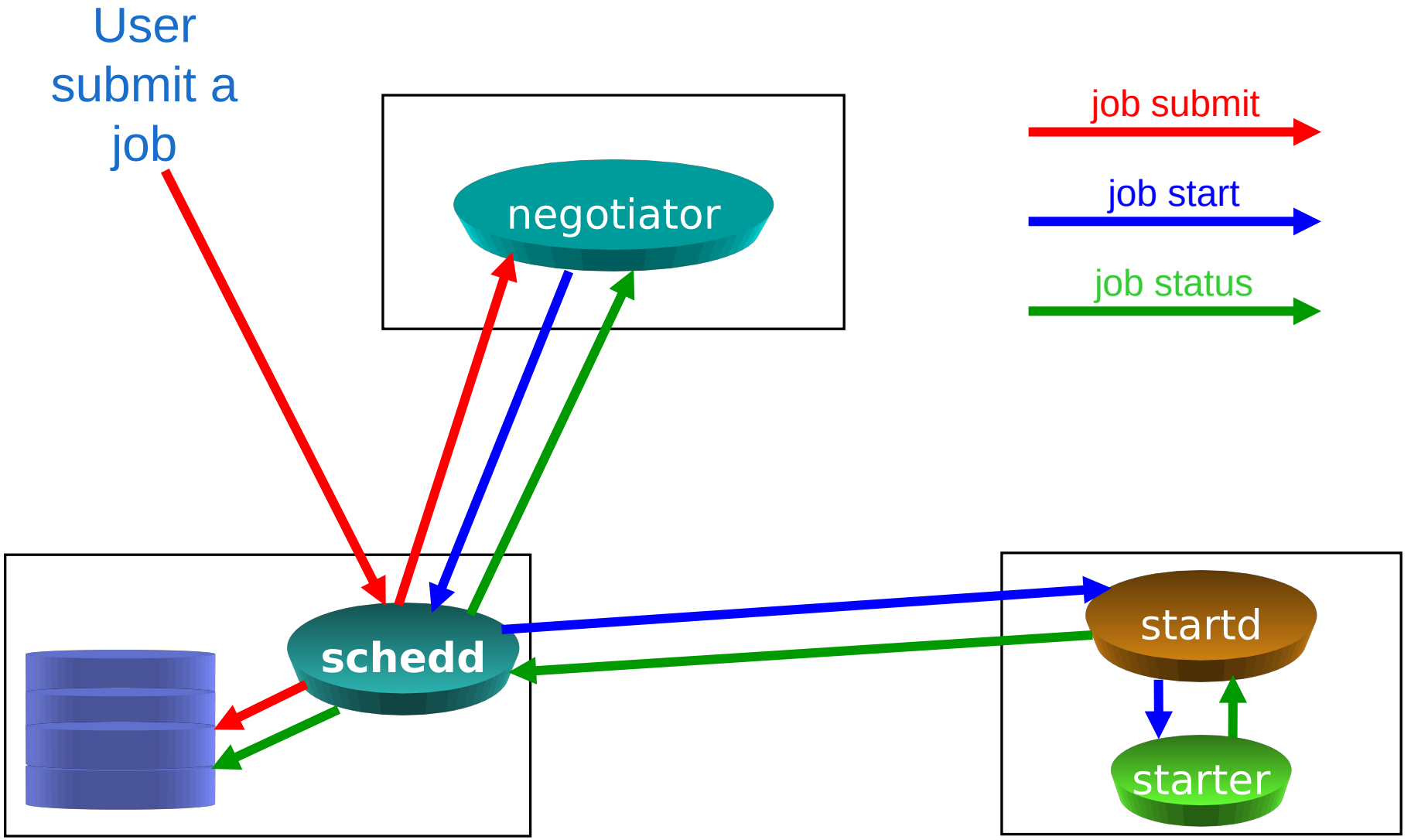
© 2012 IBM Corporation

Machine Definition



© 2012 IBM Corporation

LoadLeveler Job Cycle



© 2012 IBM Corporation

Defer and Hold vs Backfill

- Defer and Hold (default) Scheduling
  - Top job waits short time for resources to free
  - Defer job if resources are not available

- Backfill
  - Top job starts if enough resources are available
  - If not enough resources to start now then determine future start time when enough resources will be available to start job
  - Lower priority jobs which do not interfere with the start time of the top job are backfilled onto available resources

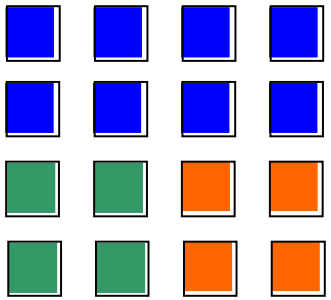
- Backfill is recommended for scheduling parallel jobs

© 2012 IBM Corporation

Backfill Scheduler

Job Queue

Job	Nodes	Wall Clock
Job A	8	2
Job B	12	1
Job C	8	3
Job D	4	1
Job E	4	5

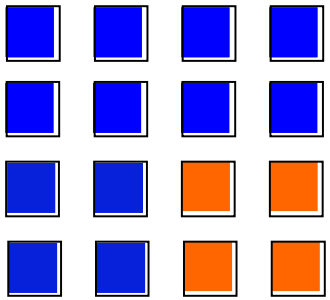


© 2012 IBM Corporation

Backfill Scheduler

Job Queue

Job	Nodes	Wall Clock
~~Job A~~	~~8~~	~~2~~
Job B	12	1
Job C	8	3
~~Job D~~	~~4~~	~~1~~
Job E	4	5



© 2012 IBM Corporation

Preemption

- Allows lower priority jobs to be preempted so that higher priority jobs can run
- Supported for backfill scheduler only
- Supported on AIX and Linux (suspend not available on Linux)
- User initiated preemption
  - Administrator only llpreempt command
  - Suspended jobs must be manually resumed
- System initiated preemption
  - Automatically enforced by LoadLeveler (except in reservation)
  - Uses PREEMPT_CLASS rules
  - Automatically resumed when resources become available

© 2012 IBM Corporation

Reservation

- Reserve Computing Resources (Nodes) for
  - Running a Workload
  - Maintenance
- Supported with backfill scheduler only
- A reservation is a set of nodes reserved for a period of time
  - Unique reservation ID
  - Owner
  - Start time
  - Duration
  - List of reserved nodes
  - List of users that could use reservation

© 2012 IBM Corporation

Fair Share Scheduling

- Divide cluster resources “fairly” among users or group of users.
- It’s all through priority. No change to job scheduling algorithm.
- Allocate a proportion of resources to users or groups of users.
- Let job priority change according to allocated and used shares.

© 2012 IBM Corporation

Consumable Resources

- Scheduler keeps track of consumable resources
- Amount of available resource is reduced by requested amount when a job is scheduled
- Amount of available resource is increased when a job completes
- Machine consumable resources
  - ConsumableCpus
  - ConsumableMemory
  - ConsumableVirtualMemory
  - Administrator defined
- Cluster consumable resources (e.g. software licenses)
- Only CPU and real memory are enforced

© 2012 IBM Corporation

Job Command File Basics

Command file contains job “directives”

Basic items include:
* Shell
* Class
* Input/output directories
* Notification control
* Queue keyword

2 ways to specify job executable:
* Executable keyword
* Script invocation after the keyword

© 2012 IBM Corporation

Basic Job Command File

```
#!/bin/ksh
# @ class = large
# @ queue
./exe
```


© 2012 IBM Corporation

More Job Command File Keywords

Requirements allow you to select:

* I/O directives
* Node requirements
* Wallclock limit
* Locally defined requirements

Notification Controls what LL sends about the job

* From *never* to *always*

notify_user tells LL where to send job info

* An email address

© 2012 IBM Corporation

Serial Job Command File

```
#!/bin/ksh
# @ error = ./out/job2.$(jobid).err
# @ output = ./out/job2.$(jobid).out
# @ wall_clock_limit = 180
# @ class = large
# @ notification = complete
# @ notify_user = pidsouza@in.ibm.com
# @ queue
./exe
```


© 2012 IBM Corporation

Communication on the System

Each node has a connection to the high-performance switch

There are 2 ways to use the switch

* ip mode "unlimited" channels slower communication perf
* User space mode limited number of channels faster than ip
* Can be selected in job command file

© 2012 IBM Corporation

Parallel Job Command File Keywords

node
How many nodes your job requires

tasks_per_node
How many tasks will run on each node

network
How your job will communicate

wall_clock_limit
An estimate of how long your job runs

© 2012 IBM Corporation

The Network Keyword

network.protocol = network_type, usage, mode

protocol: MPI, LAPI, PVM

network_type: sn_single or sn_all for switch adapter

usage: shared or not_shared

mode: IP, US

An example:

@ network.MPI = sn_single, shared, us

© 2012 IBM Corporation

Parallel Job Command File

```
#!/bin/ksh
# @ job_type = parallel
# @ node = 1
# @ tasks_per_node = 4
# @ arguments = -ilevel 6
# @ error = ./out/job3.$(jobid).err
# @ output = ./out/job3.$(jobid).out
# @ wall_clock_limit = 05:00
# @ class = demo
# @ notification = complete
# @ notify_user = pidsouza@in.ibm.com
# @ network.MPI = sn_all,shared,us
# @ queue
poe exe
```


© 2012 IBM Corporation

Basic Loadleveler Commands

llsubmit – submits a job to Loadleveler

llcancel – cancels a submitted job

llq – queries the status of jobs in the job queue

llstatus – queries the status of machines in the cluster

llclass - returns information about available classes

llprio - changes the *user priority* of a job step

© 2012 IBM Corporation

llstatus

```
[ibmhpc@hpcmgmt bm2]$ llstatus
Name                 Schedd InQ  Act Startd Run LdAvg Idle Arch      OpSys
hpcn01.tifr.res.in     Avail     0   0 Idle     0 0.00  9999 PPC64     Linux2

PPC64/Linux2             1 machines      0  jobs      0  running tasks
Total Machines           1 machines      0  jobs      0  running tasks

The Central Manager is defined on hpcn01

The BACKFILL scheduler is in use
```


© 2012 IBM Corporation

llclass (describing the class)

```
[root@hpcn02 ~]# llclass
Name                 MaxJobCPU     MaxProcCPU  Free   Max Description
                     d+hh:mm:ss    d+hh:mm:ss  Slots  Slots
---------------      ------------------ ----------------- -------  ------- --------------------
No_Class              undefined          undefined          1       1           yes
medium                undefined          undefined          5       5           yes
small                 undefined          undefined         32      32           yes
large                 undefined          1+00:00:00         2       2           yes
[root@hpcn02 ~]#

[root@hpcn02 ~]# llclass -l parallel | more

Name: parallel
Priority: 19
….
Class_comment: large MPP jobs
Wall_clock_limit: 2+00:00:05, 2+00:00:00
Def_wall_clock_limit: 2+00:00:05, 2+00:00:00 (172805 seconds, 172800 seconds)
```


© 2012 IBM Corporation

llsubmit (multijob command file)

```
#!/bin/ksh
# @ job_name = Nt84bd
# @ comment = "BG Job by Size"
# @ error = $(home)/<userid>/Outputs/$(job_name).$(stepid).$(jobid).err
# @ output = $(home)/<userid>/Outputs/$(job_name).$(stepid).$(jobid).out
# @ environment = COPY_ALL
# @ notification = error
# @ notify_user = user@incois.com
#############################################################
# @ step_name = step_1
# @ wall_clock_limit = 24:00:00
# @ class = large
# @ queue
#############################################################
# @ step_name = step_2
# @ dependency = (step_1 == 0)
# @ wall_clock_limit = 24:00:00
# @ class = large
# @ job_type = bluegene
# @ queue
#############################################################
```


© 2012 IBM Corporation

llq

```
ibm@f2n1login1/home/ibm>llq
Id                       Owner      Submitted   ST PRI Class        Running On
------------------------ ---------- ----------- -- --- ------------ -----------
f2n1login1.341.0         sjo         2/215:31    R   50  large          f1n4

1 job step(s) in queue, 0 waiting, 0 pending, 1 running, 0 held, 0 preempted



ibm@f2n1login1/home/ibm>llq -s f2n1login1.341.0

===== EVALUATIONS FOR JOB STEP f2n1login1.341.0 =====

Step state                    : Running
Since job step status is not Idle, Not Queued, or Deferred, no attempt has
been made to determine why this job step has not been started.
```


© 2012 IBM Corporation

llq (job long description)

```
ibm@f2n1login1/home/ibm>llq -l f2n1login1.341.0 | more
Job Step Id: f2n1login1.341.0
Job Name: f2n1login1.341
Queue Date: Tue Feb  2 15:31:52 IST 2010
Status: Running

Dispatch Time: Wed Feb  3 02:31:45 IST 2010
Notifications: Never

SMT required: as_is
Parallel Threads:  0
            Env:
            In: /dev/null
            Out: hycom.out
            Err: hycom.out
Initial Working Dir: /gpfs1/sjo/hycom/INDx0.25/expt_01.5

Step Type: General Parallel
Node Usage: not_shared
```


© 2012 IBM Corporation

llckpt, llcancel

* llckpt

You can mark a job step for checkpoint by specifying checkpoint=yes or checkpoint=interval in the job command file

To restart a job from a checkpoint file, the original job command file should be used with the value of the restart_from_ckpt keyword set to yes. The name and location of the checkpoint file should be specified by the ckpt_dir and ckpt_file keywords.

* llcancel

This example cancels the job step 3 that is part of the job 18 that is scheduled by the machine named bronze:

llcancel bronze.18.3 (taken from man page)

© 2012 IBM Corporation

Advanced Topics

Job Preemption

Job Checkpointing

Loadleveler APIs (data access, scheduling)

Consumable resource control

Advance Reservation

Submit filter

© 2012 IBM Corporation

Features in LoadLeveler

- Backfill scheduling
- Multiple top dogs
- Batch and interactive pools
- Parallel Environment (batch and interactive) support
- OpenMPI
- Fair sharing scheduling
- **Co-scheduling**
- **Dependent job step**
- OpenMP thread level binding
- **Scheduling API**
- Scheduling by blocking or packing tasks (breadth vs. depth) or by task geometry
- Job process tracking
- **Checkpoint/Restart**
- **Reservation / Recurring**
- Job Preemption
- Consumable Resource Scheduling (with enforcement option thru WLM)
- Multi-cluster Support
- **Job Accounting**

© 2012 IBM Corporation

Tips for Efficient Job Processing

Assumptions

* One task per CPU
* Classes Configured

Get your job to the TOP of the queue

* Short run
* Small number of nodes
* Use ip communication over the switch
* Priority?
* Submit during low use periods (evening)

These are FREE!

* All above tips (except priority) will impact no other job

© 2012 IBM Corporation

More Tips for Efficient Job Processing

Allow your job to run as QUICKLY as possible:

Balance node operations

Keep data entirely in physical memory

Use processors of similar types (system admin?)

Use distributed data load and store

Profile your applications for efficient compiler use

© 2012 IBM Corporation

References

- TWS Loadleveler
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.l

© 2012 IBM Corporation