Hálózatba kapcsolt erőforrás platformok és alkalmazásaik

Simon Csaba
TMIT
2018

Message Passing Interface

Message Passing Pros and Cons

- Pros
  - Simpler and cheaper hardware
  - Explicit communication makes programmers aware of costly (communication) operations
- Cons
  - Explicit communication is painful to program
  - Requires manual optimization
    - If you want a variable to be local and accessible via LD/ST, you must declare it as such
    - If other processes need to read or write this variable, you must explicitly code the needed sends and receives to do this

Message Passing: A Program

- Calculating the sum of array elements

```
#define ASIZE 1024
#define NUMPROC 4
double myArray[ASIZE/NUMPROC];
double mySum=0;
for(int i=0;i<ASIZE/NUMPROC;i++)
  mySum+=myArray[i];
if(myPID=0){
  for(int p=1;p<NUMPROC;p++){
    int pSum;
    recv(p,pSum);
    mySum+=pSum;
  }
  printf("Sum: %lf\n",mySum);
}else
  send(0,mySum);
```

Must manually split the array

"Master" processor adds up partial sums and prints the result

"Slave" processors send their partial results to master

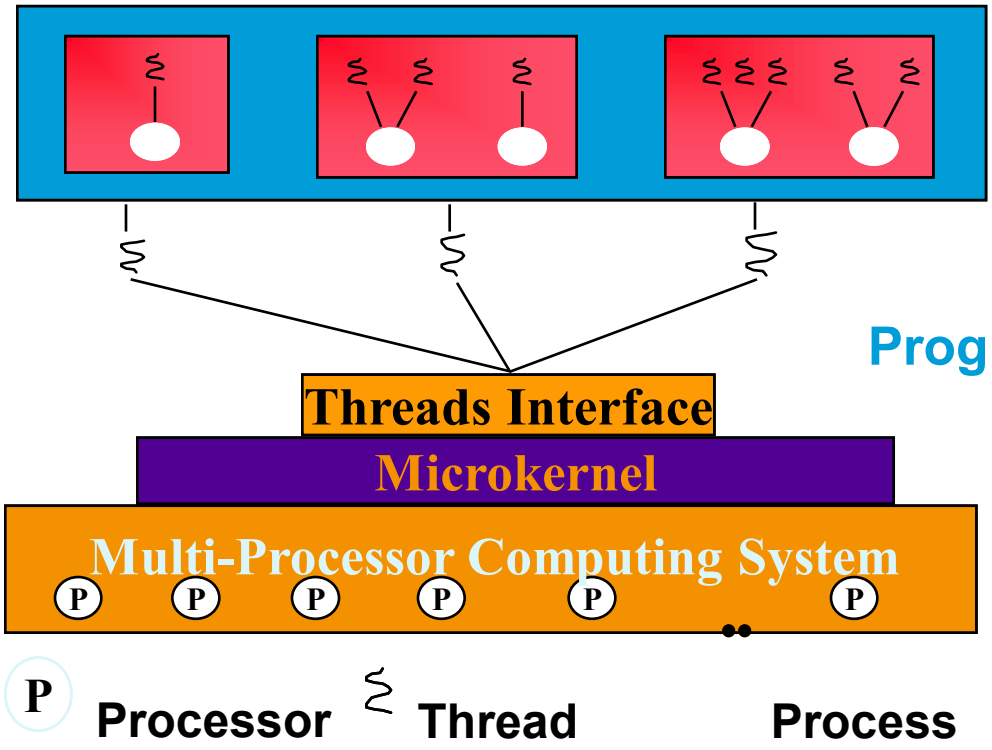
MPI programming example

- https://hpcc.usc.edu/support/documentation/examples-of-mpi-programs

Shared Memory Pros and Cons

- Pros
  - Communication happens automatically
  - More natural way of programming
    - Easier to write correct programs and gradually optimize them
  - No need to manually distribute data (but can help if you do)
- Cons
  - Needs more hardware support
  - Easy to write correct, but inefficient programs (remote accesses look the same as local ones)

Computing Elements



Applications

Programming Paradigms

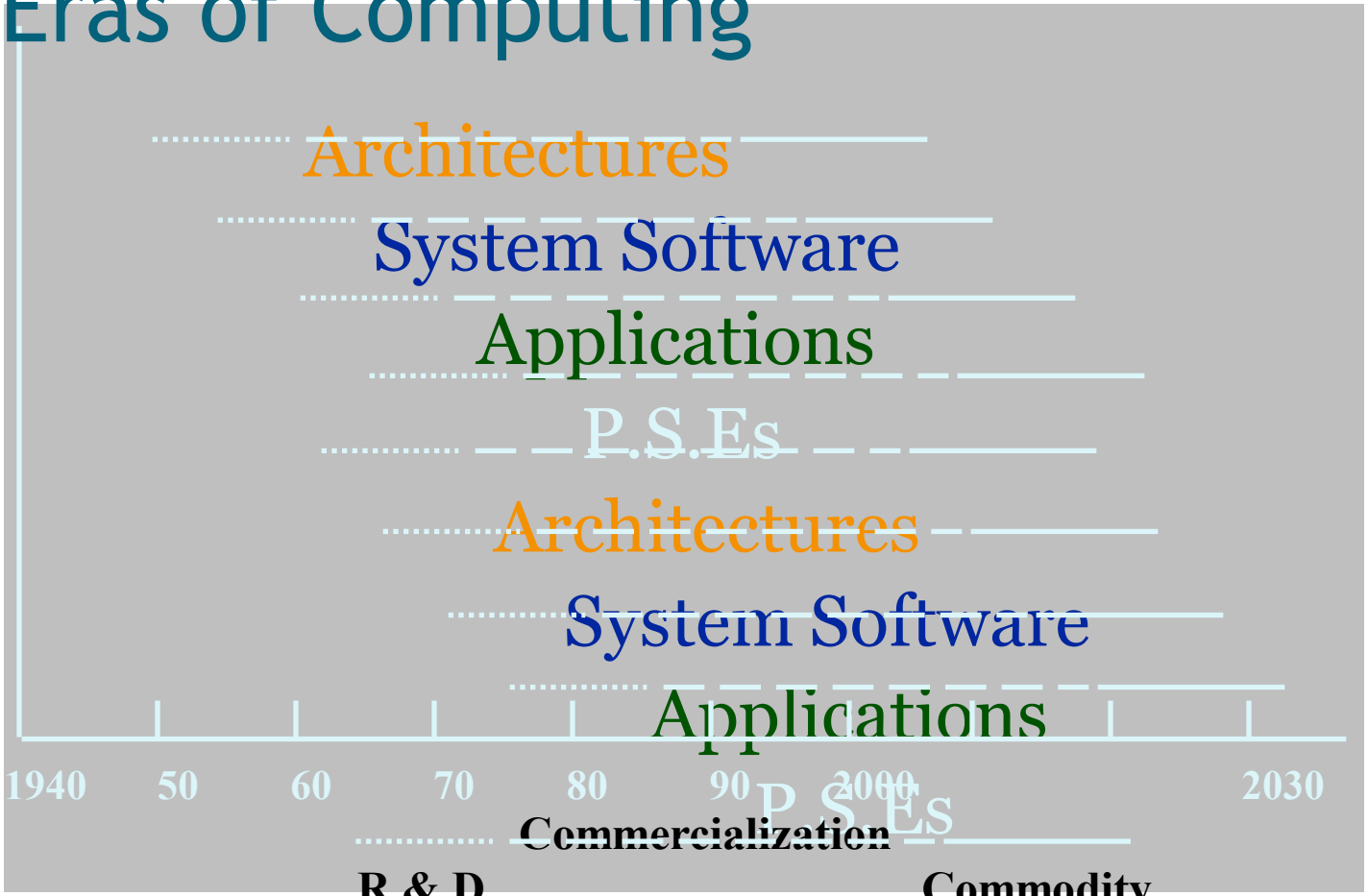
Operating System

Hardware

Two Eras of Computing

Sequential Era

Parallel Era



High-Performance Computing / Introduction

Source: James R. Knight/Yale Center for Genome Analysis

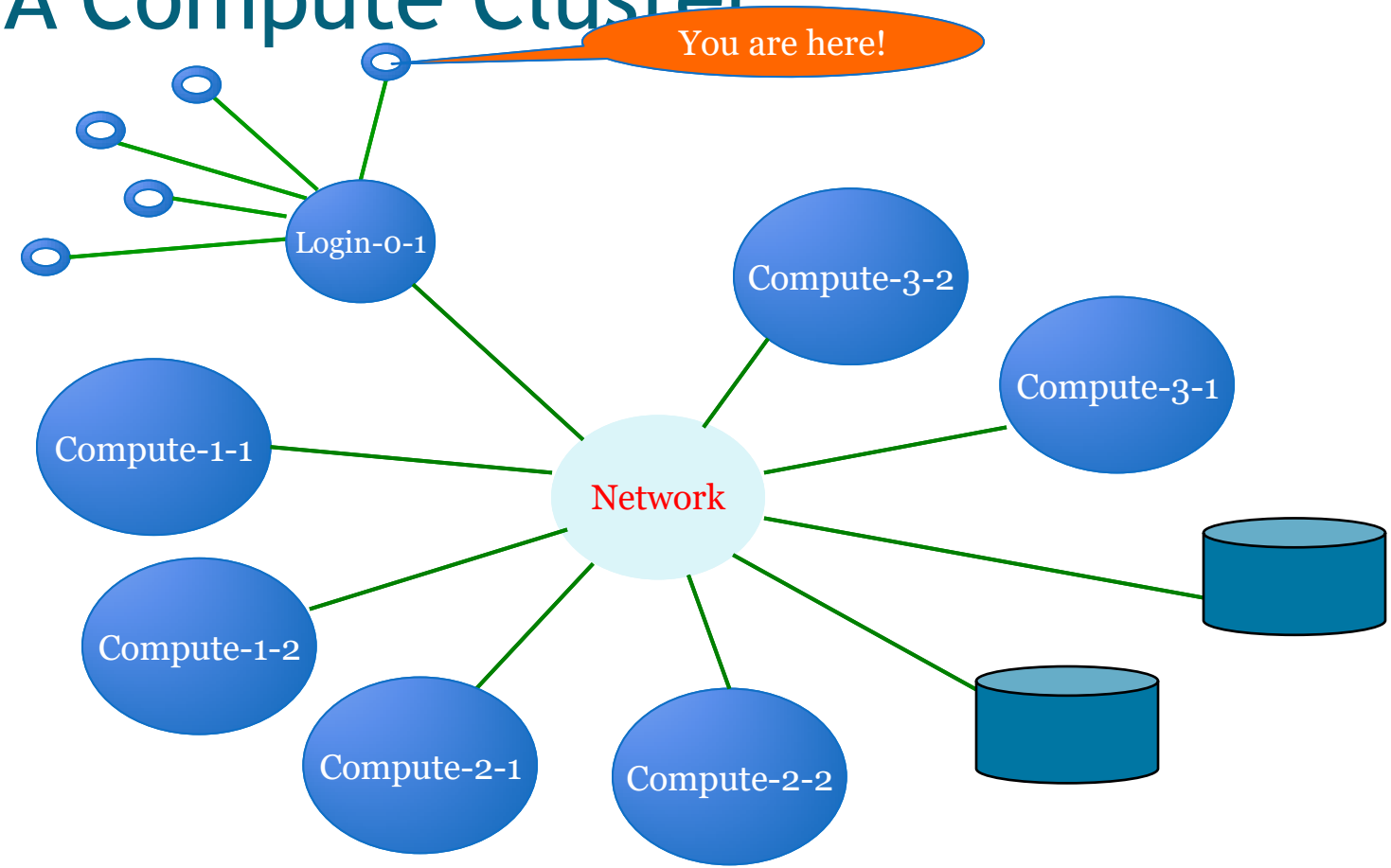
1950’s – The Beginning...

2016 – Looking very similar...

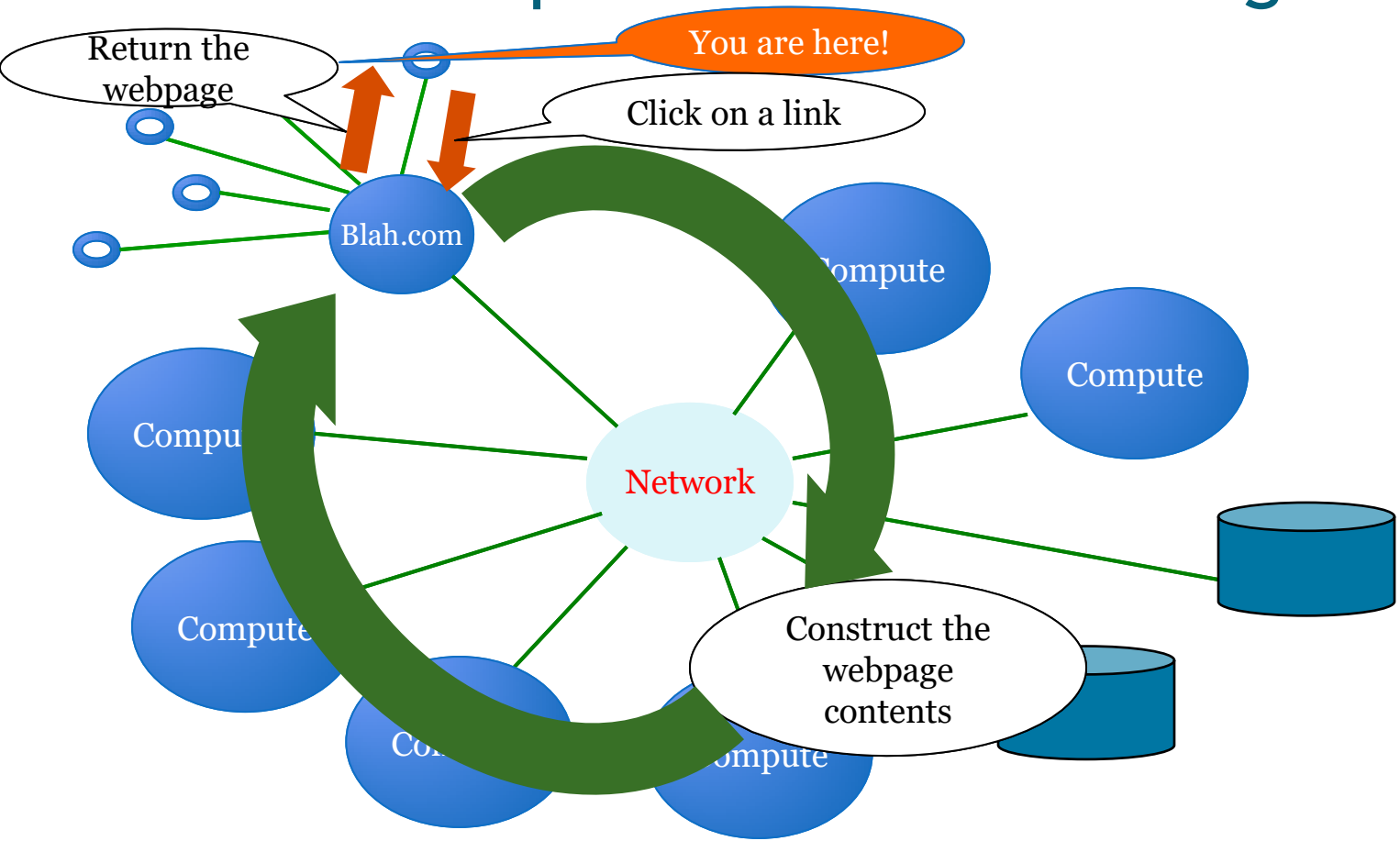
…but there are differences

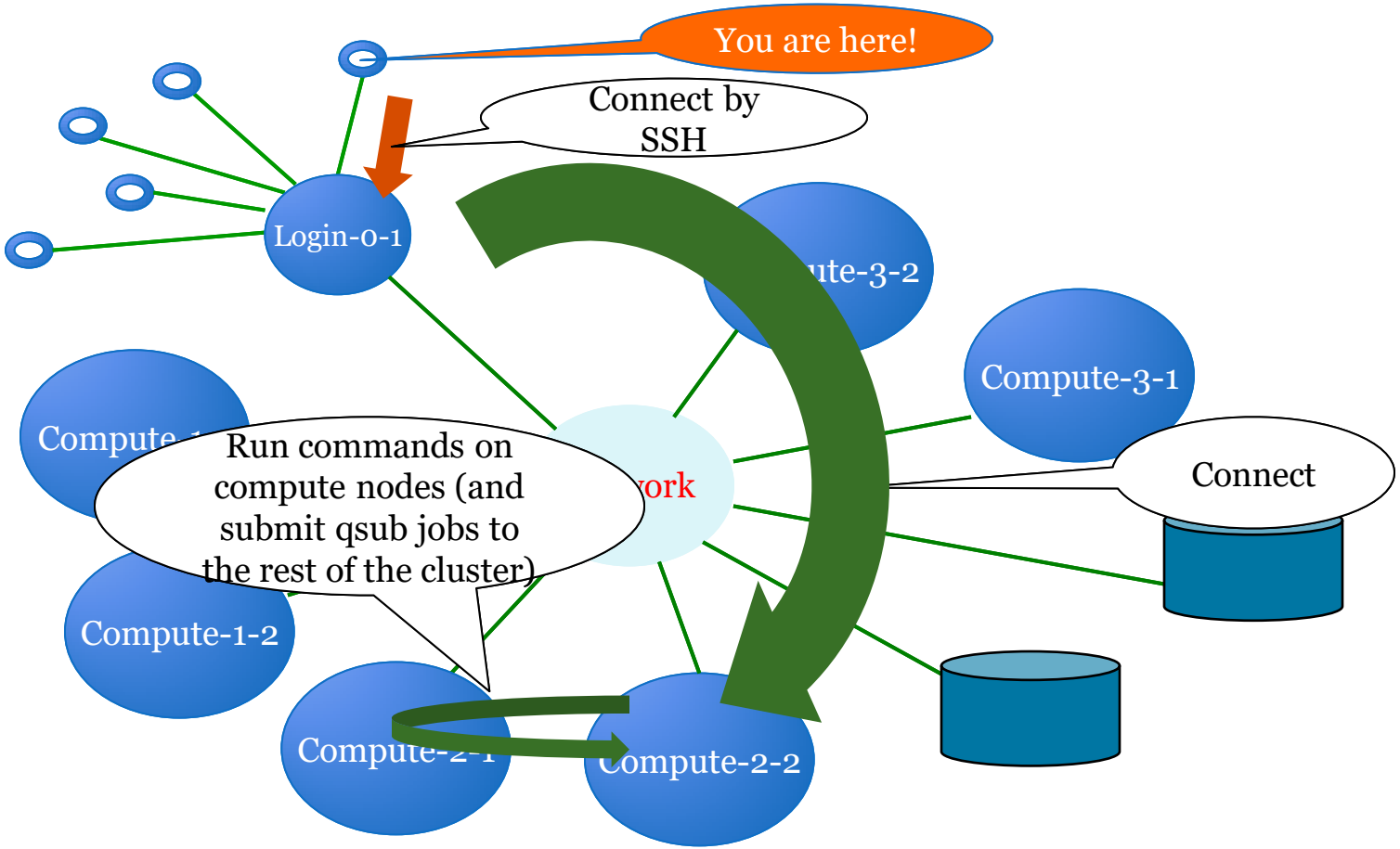
- Not a single computer but thousands of them, called a **cluster**
  - Hundreds of physical "computers", called **nodes**
  - Each with 4-64 CPU's, called **cores**
- Nobody works in the server rooms anymore
  - IT is there to fix what breaks, not to run computations (or help you run computations)
  - Everything is done by remote connections
- Computation is performed by submitting **jobs** for running
  - This actually hasn't changed…but how you run jobs has…

A Compute Cluster



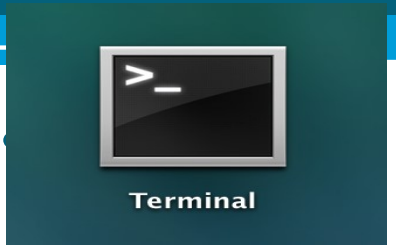
You Use a Compute Cluster! Surfing the Web




You are here!
Connect by SSH
Login-0-1
Compute-3-2
Compute-3-1
Connect
Run commands on compute nodes (and submit qsub jobs to the rest of the cluster)
Compute-1-2
Compute-2-2

1970's – Terminals, In the Beginning...

2016 – Pretty much the same



- Terminal app on Mac
- Look in the “Other” folder in Launchpad

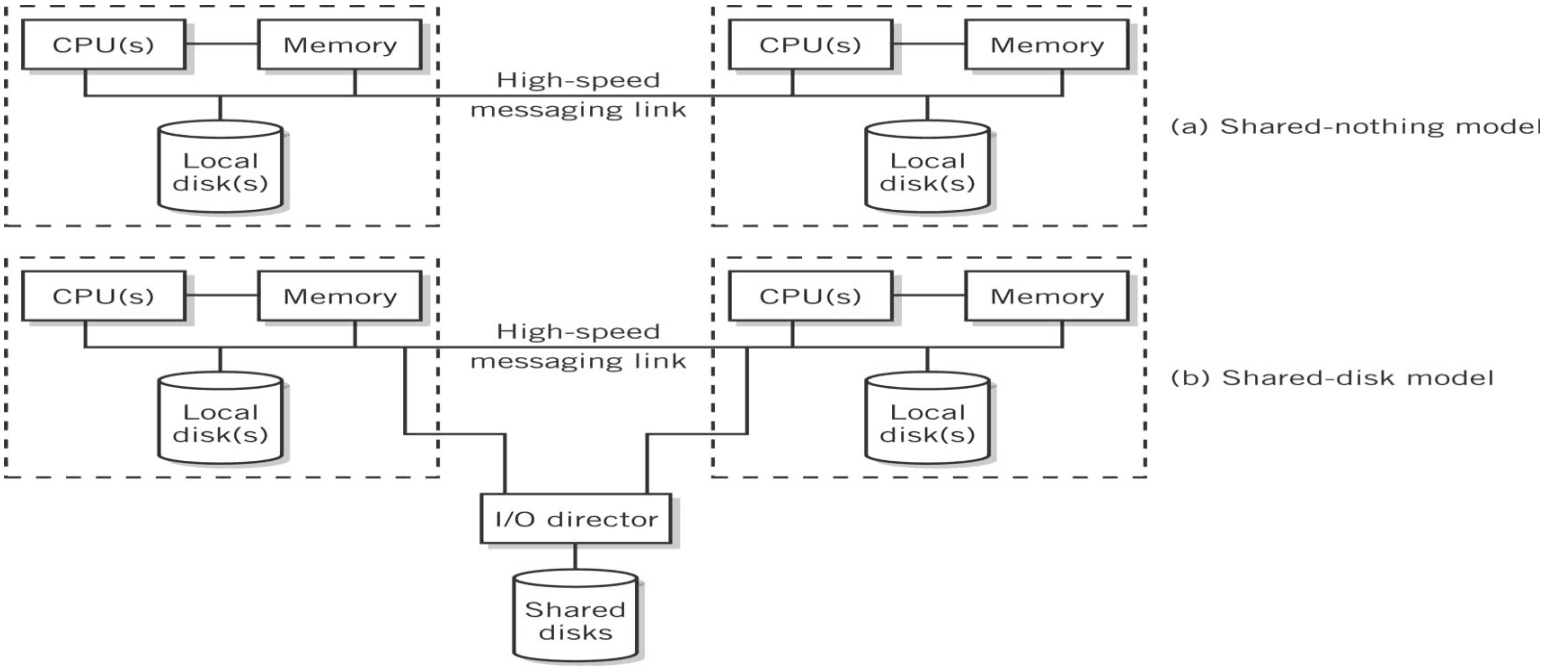
```
jamesknight — jk2269@login-0-0:~ — ssh — 95×37
Last login: Thu Jan  8 17:03:29 on ttys000
Jamess-MacBook-Pro-2:~ jamesknight$ ssh jk2269@louise.hpc.yale.edu
jk2269@louise.hpc.yale.edu's password:
Last login: Thu May 15 15:38:39 2014 from vpn172022117249.its.yale.internal
                    _                 _
                   | |               (_)
                   | |     ___   _   _  ___   ___
                   | |    / _ \| | | | / __|/ _ \
                   | |___| (_) | |_| | \__ \  __/
                   \_____/\___/ \__,_|_|___/\___|


                          at Yale University

          ================== ATTENTION ===================
Use of Yale's electronic systems is governed by applicable laws
and policies (http://www.yale.edu/policy/).  Violators and
unauthorized access may be prosecuted.
          ================================================

 * No sensitive information may be stored on Louise. Please see:
      http://www.yale.edu/its/secure-computing/data/compliance/hipaa.html
      for more information.
 * Documentation pertaining to the use of the system
      can be found here: http://maguro.cs.yale.edu/hpc.html
      and here:          http://hpc.research.yale.edu/
      and here:          http://hpc.yale.edu/
 * The script /usr/local/cluster/bin/myquota.sh
      will give your current storage usage & limits.
 * The script /usr/local/cluster/bin/myjobs.sh
      will give your current running jobs & resources for new jobs
 * Questions, comments or criticisms should be sent to:
      robert.bjornson@yale.edu or jason.ignatius@yale.edu
        ================================================
[jk2269@login-0-0 ~]$
```

Cluster Models



Beowulf Clusters

- Simple and highly configurable
- Low cost
- Networked
  - Computers connected to one another by a private Ethernet network
  - Connection to an external network is through a single gateway computer
- Configuration
  - COTS – Commodity-off-the-shelf components such as inexpensive computers
  - Blade components – computers mounted on a motherboard that are plugged into connectors on a rack
  - Either shared-disk or shared-nothing model

Blade and Rack of Beowulf Cluster

Cluster computing concept

Cluster Computing - Research Projects

- Beowulf (CalTech and NASA) - USA
- CCS (Computing Centre Software) - Paderborn, Germany
- Condor - Wisconsin State University, USA
- DQS (Distributed Queuing System) - Florida State University, US.
- EASY - Argonne National Lab, USA
- HPVM -(High Performance Virtual Machine),UIUC&now UCSB,US
- *far* - University of Liverpool, UK
- Gardens - Queensland University of Technology, Australia
- MOSIX - Hebrew University of Jerusalem, Israel
- MPI (MPI Forum, MPICH is one of the popular implementations)
- NOW (Network of Workstations) - Berkeley, USA
- NIMROD - Monash University, Australia
- NetSolve - University of Tennessee, USA
- PBS (Portable Batch System) - NASA Ames and LLNL, USA
- PVM - Oak Ridge National Lab./UTK/Emory, USA

Cluster Computing - Commercial Software

- Codine (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- LoadLeveler - IBM Corp., USA
- LSF (Load Sharing Facility) - Platform Computing, Canada
- NQE (Network Queuing Environment) - Craysoft Corp., USA
- OpenFrame - Centre for Development of Advanced Computing, India
- RWPC (Real World Computing Partnership), Japan
- Unixware (SCO-Santa Cruz Operations,), USA
- Solaris-MC (Sun Microsystems), USA
- ClusterTools (A number for free HPC clusters tools from Sun)
- A number of commercial vendors worldwide are offering clustering solutions including IBM, Compaq, Microsoft, a number of startups like TurboLinux, HPTI, Scali, BlackStone.....)

Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Motivation for using Clusters

- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.
- Workstation clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Cycle Stealing

- Usually a workstation will be *owned* by an individual, group, department, or organisation - they are dedicated to the exclusive use by the *owners*.
- This brings problems when attempting to form a cluster of workstations for running distributed applications.

Cycle Stealing

- Typically, there are three types of owners, who use their workstations mostly for:
  1. Sending and receiving email and preparing documents.
  2. Software development - edit, compile, debug and test cycle.
  3. Running compute-intensive applications.

Cycle Stealing

- Cluster computing aims to steal spare cycles from (1) and (2) to provide resources for (3).
- However, this requires overcoming the *ownership hurdle* - people are very protective of *their* workstations.
- Usually requires organisational mandate that computers are to be used in this way.
- Stealing cycles outside standard work hours (e.g. overnight) is easy, stealing idle cycles during work hours without impacting interactive use (both CPU and memory) is much harder.

Type of Clusters

- HA
- Load distribution

High-Performance Computing / Introduction

Source: James R. Knight/Yale Center for Genome Analysis

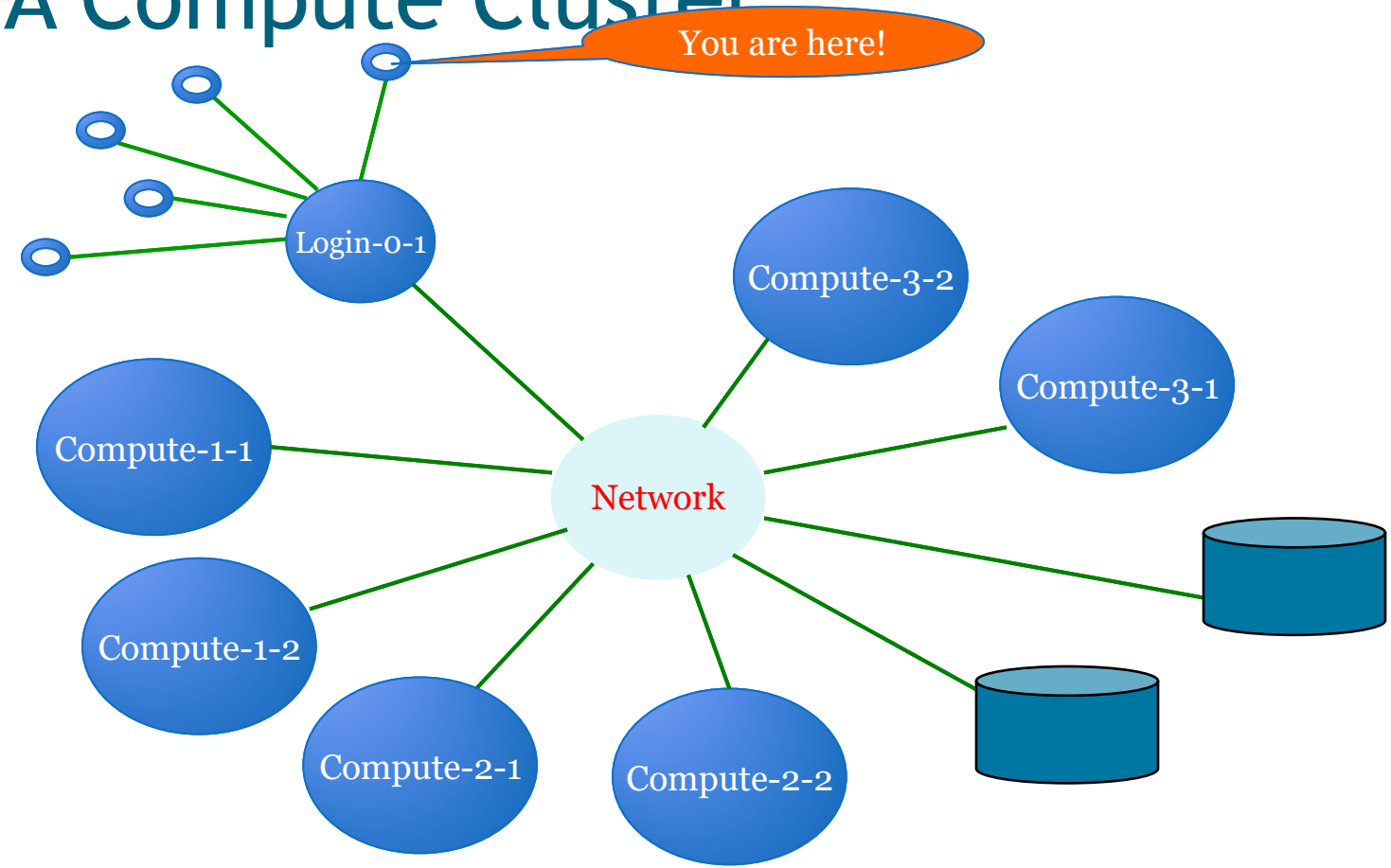
1950’s – The Beginning...

2016 – Looking very similar...

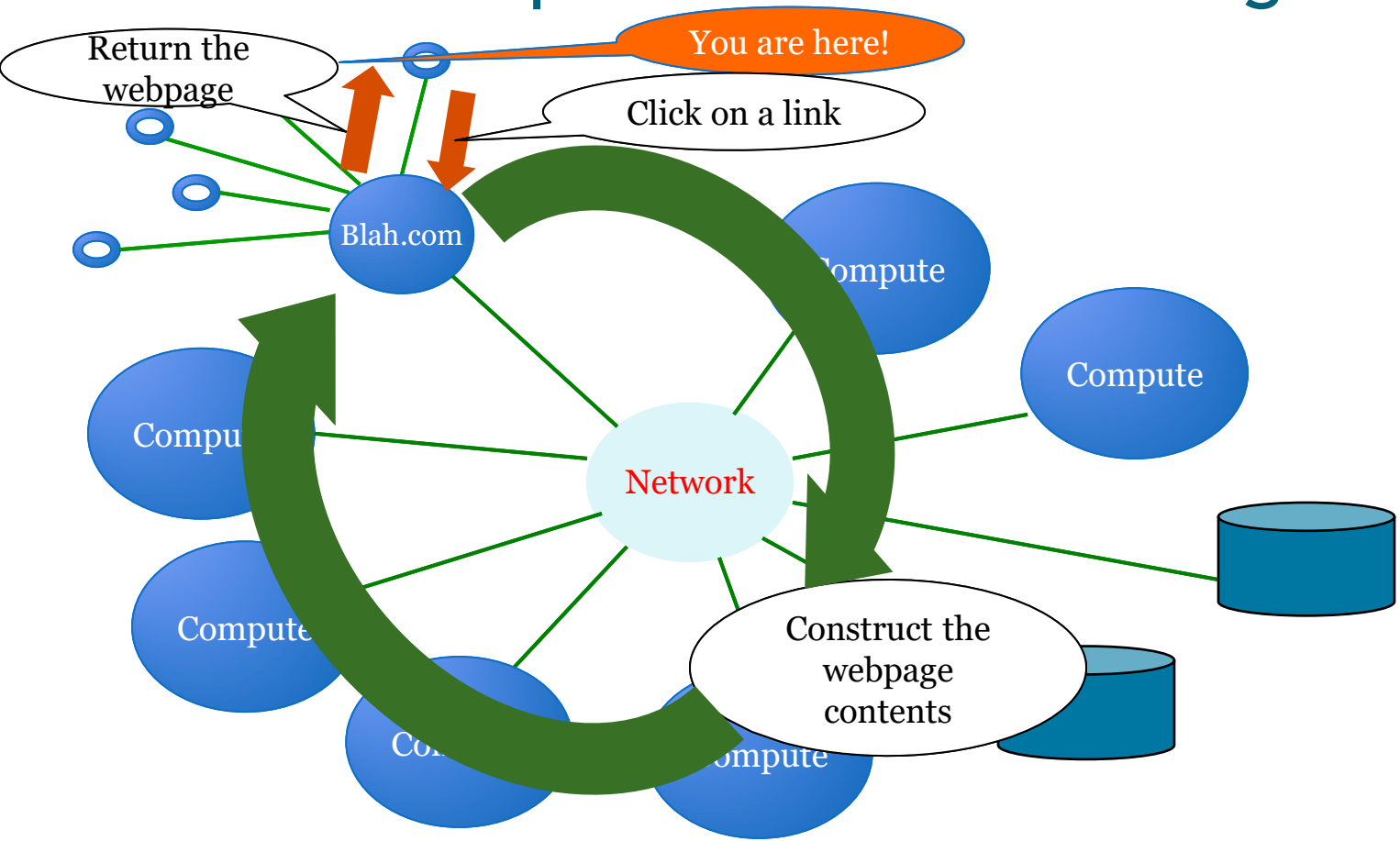
…but there are differences

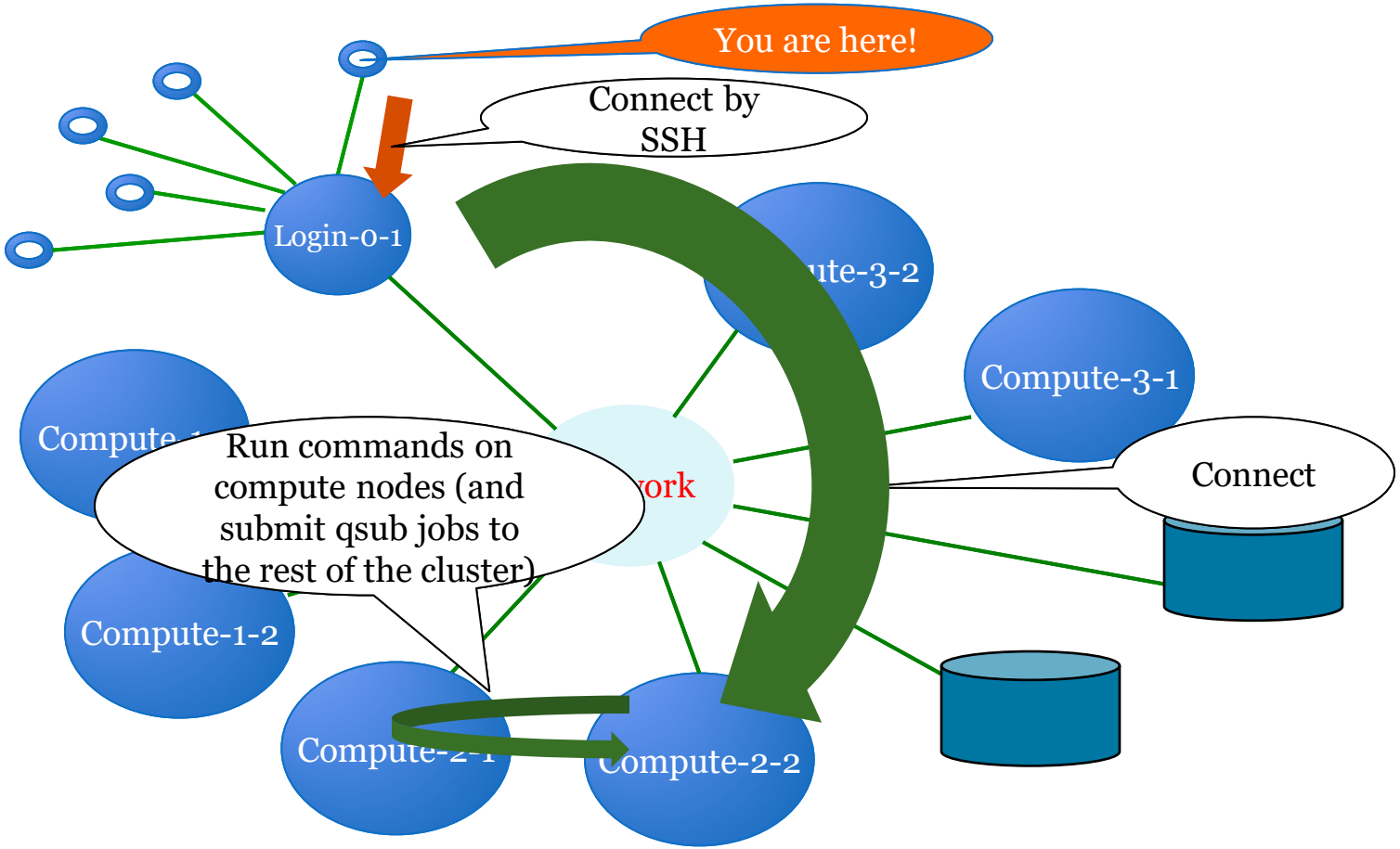
- Not a single computer but thousands of them, called a **cluster**
  - Hundreds of physical "computers", called **nodes**
  - Each with 4-64 CPU's, called **cores**
- Nobody works in the server rooms anymore
  - IT is there to fix what breaks, not to run computations (or help you run computations)
  - Everything is done by remote connections
- Computation is performed by submitting **jobs** for running
  - This actually hasn't changed…but how you run jobs has…

A Compute Cluster



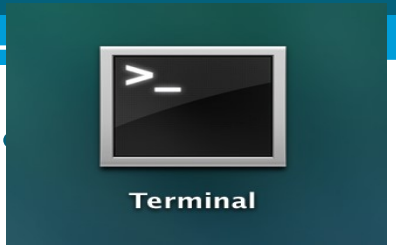
You Use a Compute Cluster! Surfing the Web




You are here!
Connect by SSH
Login-0-1
Compute-3-2
Compute-3-1
Connect
Run commands on compute nodes (and submit qsub jobs to the rest of the cluster)
Compute-1-2
Compute-2-2

1970's – Terminals, In the Beginning...

2016 – Pretty much the same



- Terminal app on Mac
- Look in the “Other” folder in Launchpad

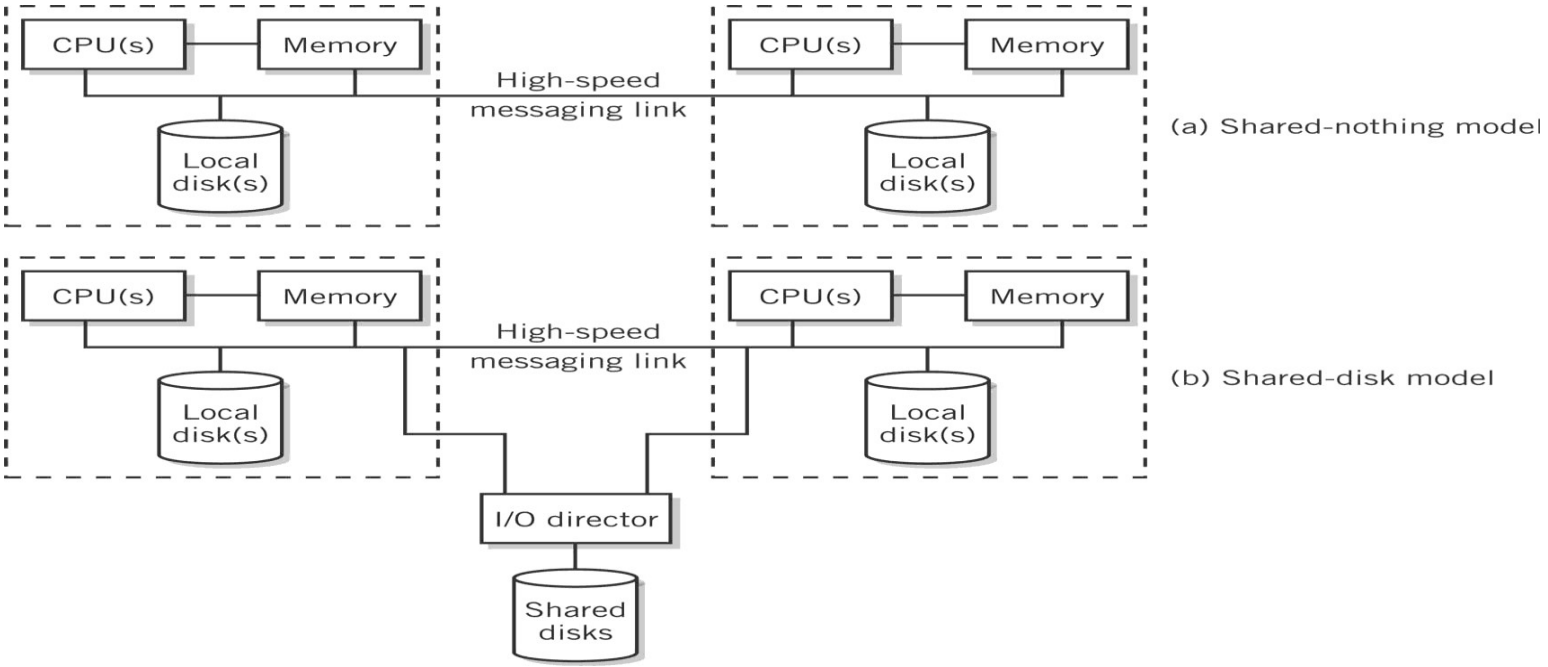
```
jamesknight — jk2269@login-0-0:~ — ssh — 95×37
Last login: Thu Jan  8 17:03:29 on ttys000
Jamess-MacBook-Pro-2:~ jamesknight$ ssh jk2269@louise.hpc.yale.edu
jk2269@louise.hpc.yale.edu's password:
Last login: Thu May 15 15:38:39 2014 from vpn172022117249.its.yale.internal
                    _                 _
                   | |               (_)
                   | |     ___   _   _  ___   ___
                   | |    / _ \| | | | / __|/ _ \
                   | |___| (_) | |_| | \__ \  __/
                   \_____/\___/ \__,_|_|___/\___|


                          at Yale University

          ================== ATTENTION ===================
Use of Yale's electronic systems is governed by applicable laws
and policies (http://www.yale.edu/policy/).  Violators and
unauthorized access may be prosecuted.
          ================================================

 * No sensitive information may be stored on Louise. Please see:
      http://www.yale.edu/its/secure-computing/data/compliance/hipaa.html
      for more information.
 * Documentation pertaining to the use of the system
      can be found here: http://maguro.cs.yale.edu/hpc.html
      and here:          http://hpc.research.yale.edu/
      and here:          http://hpc.yale.edu/
 * The script /usr/local/cluster/bin/myquota.sh
      will give your current storage usage & limits.
 * The script /usr/local/cluster/bin/myjobs.sh
      will give your current running jobs & resources for new jobs
 * Questions, comments or criticisms should be sent to:
      robert.bjornson@yale.edu or jason.ignatius@yale.edu
        ================================================
[jk2269@login-0-0 ~]$
```

Cluster Models



Beowulf Clusters

- Simple and highly configurable
- Low cost
- Networked
  - Computers connected to one another by a private Ethernet network
  - Connection to an external network is through a single gateway computer
- Configuration
  - COTS – Commodity-off-the-shelf components such as inexpensive computers
  - Blade components – computers mounted on a motherboard that are plugged into connectors on a rack
  - Either shared-disk or shared-nothing model

Blade and Rack of Beowulf Cluster

Cluster computing concept

Cluster Computing - Research Projects

- Beowulf (CalTech and NASA) - USA
- CCS (Computing Centre Software) - Paderborn, Germany
- Condor - Wisconsin State University, USA
- DQS (Distributed Queuing System) - Florida State University, US.
- EASY - Argonne National Lab, USA
- HPVM -(High Performance Virtual Machine),UIUC&now UCSB,US
- *far* - University of Liverpool, UK
- Gardens - Queensland University of Technology, Australia
- MOSIX - Hebrew University of Jerusalem, Israel
- MPI (MPI Forum, MPICH is one of the popular implementations)
- NOW (Network of Workstations) - Berkeley, USA
- NIMROD - Monash University, Australia
- NetSolve - University of Tennessee, USA
- PBS (Portable Batch System) - NASA Ames and LLNL, USA
- PVM - Oak Ridge National Lab./UTK/Emory, USA

Cluster Computing - Commercial Software

- Codine (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- LoadLeveler - IBM Corp., USA
- LSF (Load Sharing Facility) - Platform Computing, Canada
- NQE (Network Queuing Environment) - Craysoft Corp., USA
- OpenFrame - Centre for Development of Advanced Computing, India
- RWPC (Real World Computing Partnership), Japan
- Unixware (SCO-Santa Cruz Operations,), USA
- Solaris-MC (Sun Microsystems), USA
- ClusterTools (A number for free HPC clusters tools from Sun)
- A number of commercial vendors worldwide are offering clustering solutions including IBM, Compaq, Microsoft, a number of startups like TurboLinux, HPTI, Scali, BlackStone.....)

Motivation for using Clusters

- Surveys show utilisation of CPU cycles of desktop workstations is typically <10%.
- Performance of workstations and PCs is rapidly improving
- As performance grows, percent utilisation will decrease even further!
- Organisations are reluctant to buy large supercomputers, due to the large expense and short useful life span.

Motivation for using Clusters

- The development tools for workstations are more mature than the contrasting proprietary solutions for parallel computers - mainly due to the non-standard nature of many parallel systems.
- Workstation clusters are a cheap and readily available alternative to specialised High Performance Computing (HPC) platforms.
- Use of clusters of workstations as a distributed compute resource is very cost effective - incremental growth of system!!!

Cycle Stealing

- Usually a workstation will be *owned* by an individual, group, department, or organisation - they are dedicated to the exclusive use by the *owners*.
- This brings problems when attempting to form a cluster of workstations for running distributed applications.

Cycle Stealing

- Typically, there are three types of owners, who use their workstations mostly for:
  1. Sending and receiving email and preparing documents.
  2. Software development - edit, compile, debug and test cycle.
  3. Running compute-intensive applications.

Cycle Stealing

- Cluster computing aims to steal spare cycles from (1) and (2) to provide resources for (3).
- However, this requires overcoming the *ownership hurdle* - people are very protective of *their* workstations.
- Usually requires organisational mandate that computers are to be used in this way.
- Stealing cycles outside standard work hours (e.g. overnight) is easy, stealing idle cycles during work hours without impacting interactive use (both CPU and memory) is much harder.

Type of Clusters

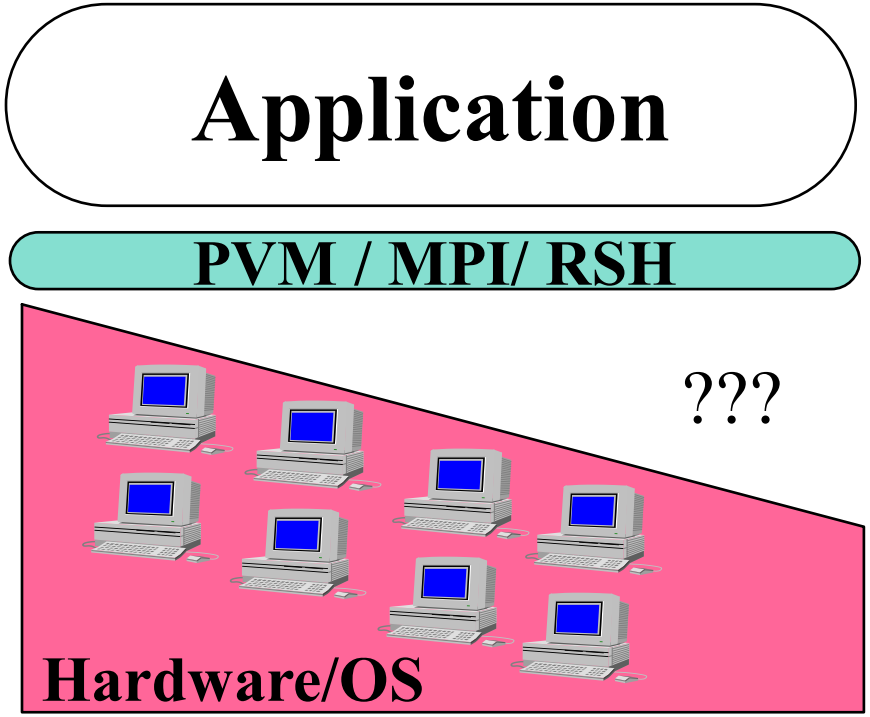
- HA
- Load distribution

P2P Computing vs Cluster/Grid Computing

- Differ in Target Communities
- Grid system deals with more complex, more powerful, more diverse and highly interconnected set of resources than P2P.

Cluster Work Schedulers

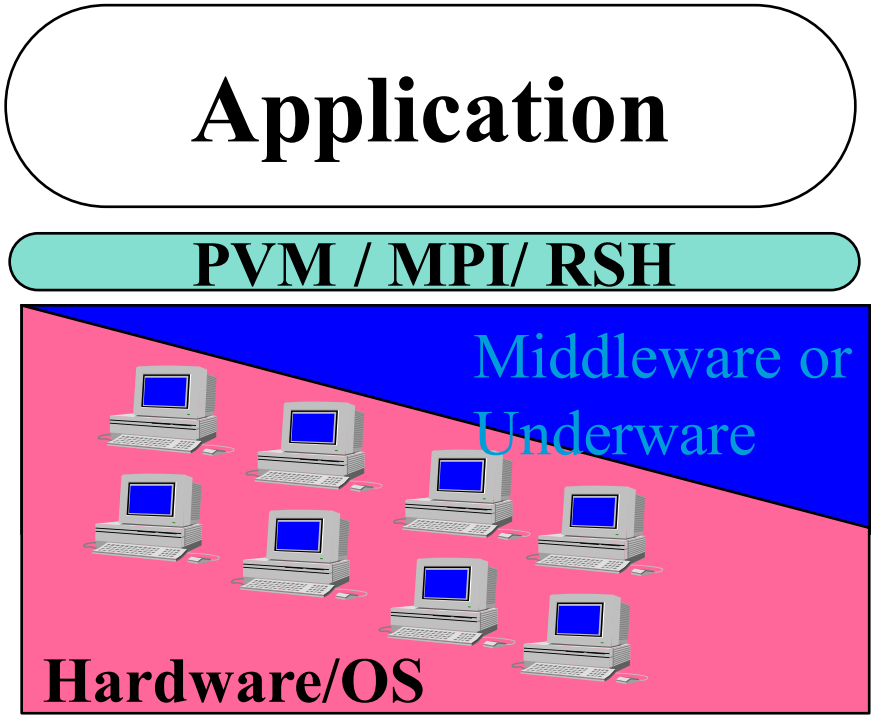
A typical Cluster Computing Environment



CC should support

- Multi-user, time-sharing environments
- Nodes with different CPU speeds and memory sizes (heterogeneous configuration)
- Many processes, with unpredictable requirements
- Unlike SMP: insufficient "bonds" between nodes
  - Each computer operates independently
  - Inefficient utilization of resources

The missing link is provide by cluster middleware/underware



SSI Clusters--SMP services on a CC

"*Pool Together*" the "*Cluster-Wide*" resources

- Adaptive resource usage for better performance
- Ease of use - almost like SMP
- Scalable configurations - by decentralized control

Result: *HPC/HAC at PC/Workstation prices*

What is Cluster Middleware ?

- An interface between between use applications and cluster hardware and OS platform.
- Middleware packages support each other at the management, programming, and implementation levels.
- Middleware Layers:
  - SSI Layer
  - Availability Layer: It enables the cluster services of
    - Checkpointing, Automatic Failover, recovery from failure,
    - fault-tolerant operating among all cluster nodes.

Middleware Design Goals

- Complete Transparency (Manageability)
  - Lets the see a single cluster system..
    - Single entry point, ftp, telnet, software loading…
- Scalable Performance
  - Easy growth of cluster
    - no change of API & automatic load distribution.
- Enhanced Availability
  - Automatic Recovery from failures
    - Employ checkpointing & fault tolerant technologies
  - Handle consistency of data when replicated..

Work schedulers - requirements

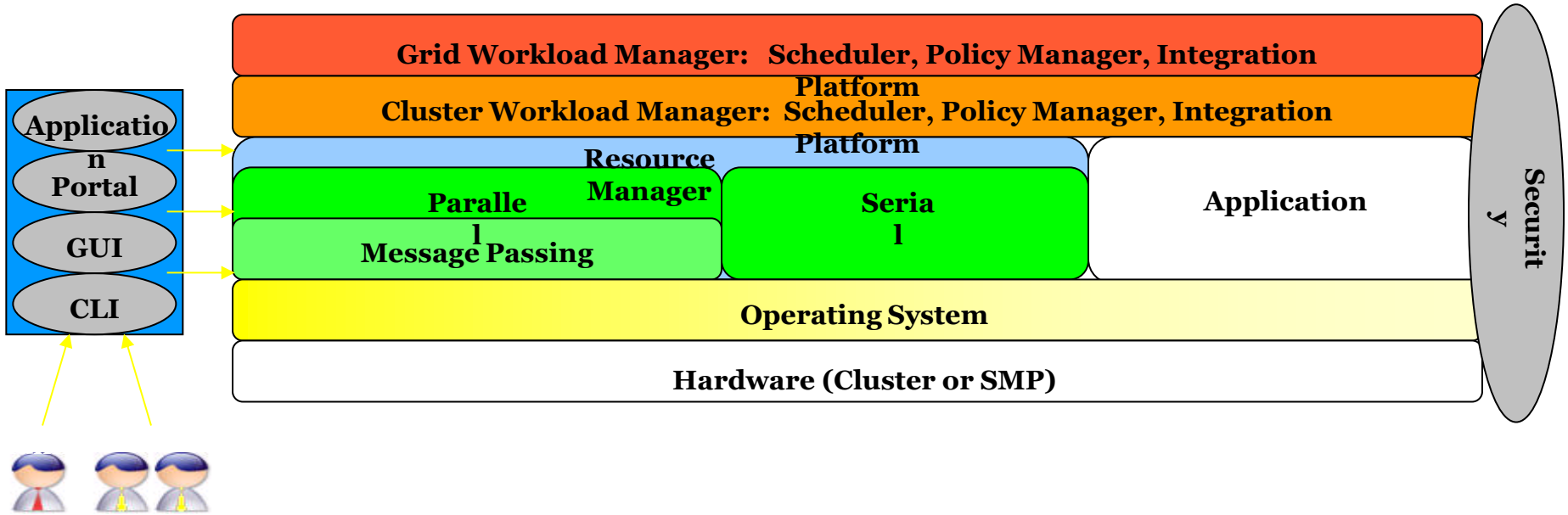
- Interactive or batch

- Stable
- Robust
- Efficient resource management
- Lightweigth
- Fair
- Avoids starvation

- SGE - Sun Grid Engine (Oracle Grid Engine, Open Grid Scheduler)
- SLURM (Simple Linux Utility for Resource Management)
- MOAB + Torque
- HTCondor
- ...

Redirect: MOAB

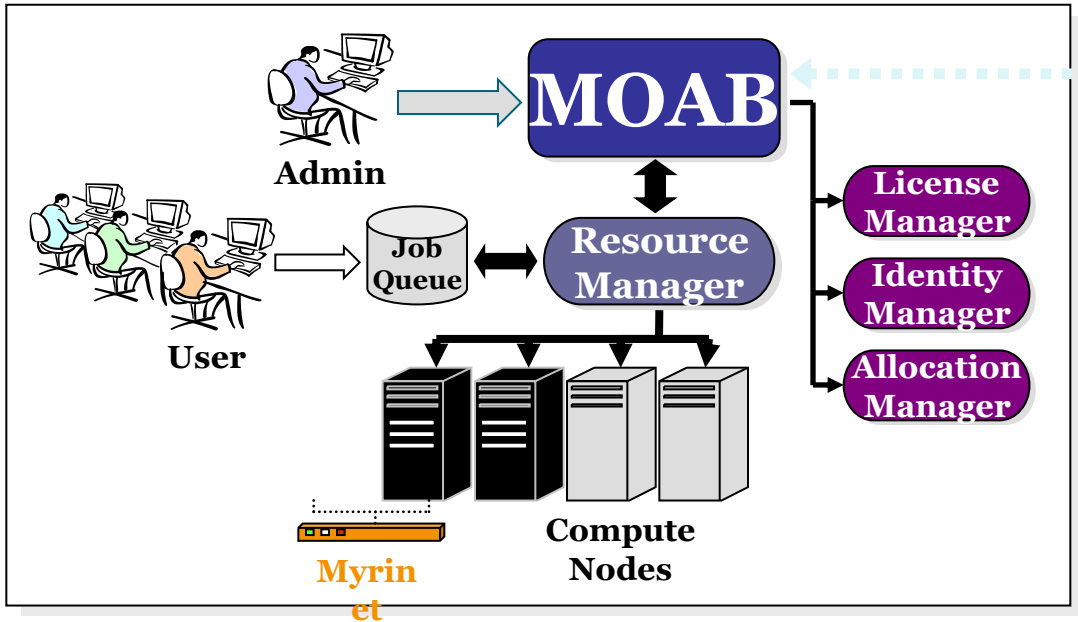
Cluster Stack / Framework:

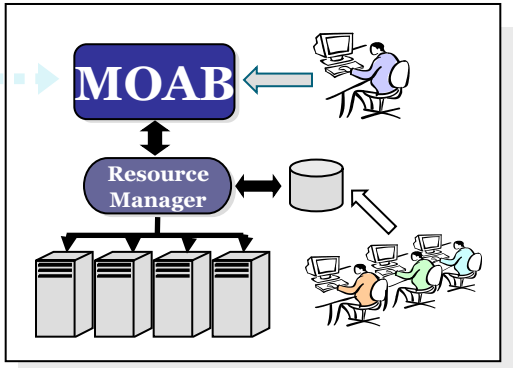


Resource Manager (RM)

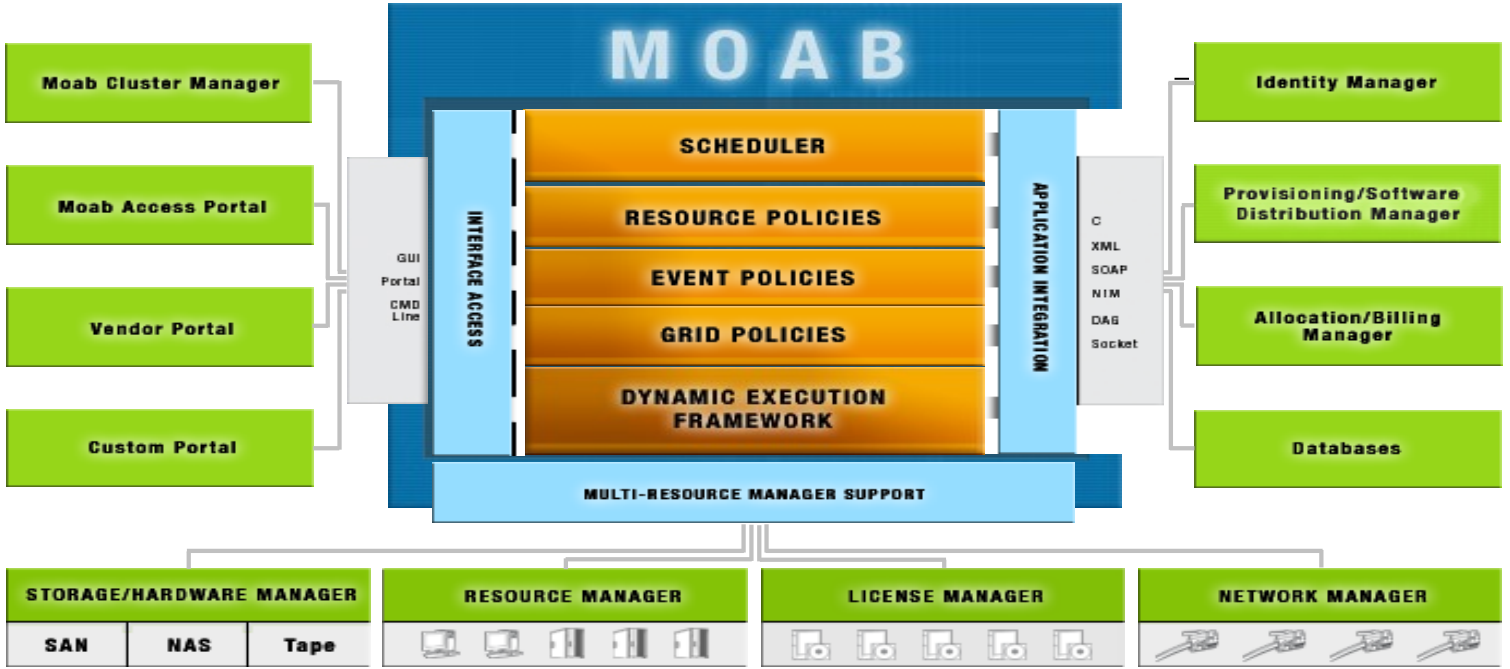
- While other systems may have more strict interpretations of a resource manager and its responsibilities, Moab's *multi-resource manager* support allows a much more liberal interpretation.
  - In essence, any object which can provide environmental information and environmental control can be utilized as a resource manager.
- Moab is able to aggregate information from multiple unrelated sources into a larger more complete *world view* of the cluster which includes all the information and control found within a standard resource manager such as TORQUE including:
  - Node
  - Job
  - Queue management services.

The Evolved Cluster





Moab Architecture



What Moab Does

- Optimizes Resource Utilization with Intelligent Scheduling and Advanced Reservations
- Unifies Cluster Management across Varied Resources and Services
- Dynamically Adjusts Workload to Enforce Policies and Service Level Agreements
- Automates Diagnosis and Failure Response

What Moab Does Not Do

- Does not does do resource management (usually)
- Does not install the system (usually)
- Not a storage manager
- Not a license manager
- Does not do message passing

Supported Platforms/Environments

- Resource Managers
  - TORQUE, OpenPBS, PBSPro, LSF, Loadleveler, SLURM, BProc, clubMASK, S3, WIKI
- Operating Systems
  - RedHat, SUSE, Fedora, Debian, FreeBSD, (+ all known variants of Linux), AIX, IRIX, HP-UX, OS/X, OSF/Tru-64, SunOS, Solaris, (+ all known variants of UNIX)
- Hardware
  - Intel x86, Intel IA-32, Intel IA-64, AMD x86, AMD Opteron, SGI Altix, HP, IBM SP, IBM x-Series, IBM p-Series, IBM i-Series, Mac G4 and G5