

Distributed file systems and grid computing

Christian Mertes
mertes@in.tum.de

Dr. Laszlo Kajan
I12 - Department for Bioinformatics and Computational Biology
Technische Universität München

26. Jun 2012

Overview

- Introduction to grid computing
- What is a distributed file system
 - ▶ Samba
 - ▶ NFS
- What is a grid engine
 - ▶ Open Grid Scheduler

Introduction

A simple example case

A biologist of your research group gives you the results of a sequencing run of an unknown bacteria.

Introduction

A simple example case

A biologist of your research group gives you the results of a sequencing run of an unknown bacteria.

Question

What would you do as a bioinformatician?

Introduction

A simple example case

A biologist of your research group gives you the results of a sequencing run of an unknown bacteria.

Question

What would you do as a bioinformatician?

Answer

You want to **run blast** on **several databases** and with **different settings**

Introduction

What kind of problems a system is faced by such a task?

Introduction

What kind of problems a system is faced by such a task?

- 1 how is the software served?

Introduction

What kind of problems a system is faced by such a task?

- ① how is the software served?
- ② how can the user/program find and access the data?

Introduction

What kind of problems a system is faced by such a task?

- ① how is the software served?
- ② how can the user/program find and access the data?
- ③ how and where are the different jobs executed?

Introduction

What kind of problems a system is faced by such a task?

- ① how is the software served?
- ② how can the user/program find and access the data?
- ③ how and where are the different jobs executed?
- ④ can independent jobs run in parallel?

Introduction

What kind of problems a system is faced by such a task?

- 1 how is the software served?
- 2 how can the user/program find and access the data?
- 3 how and where are the different jobs executed?
- 4 can independent jobs run in parallel?
- 5 how are permissions handled?

Introduction

What kind of problems a system is faced by such a task?

- 1 how is the software served?
- 2 how can the user/program find and access the data?
- 3 how and where are the different jobs executed?
- 4 can independent jobs run in parallel?
- 5 how are permissions handled?

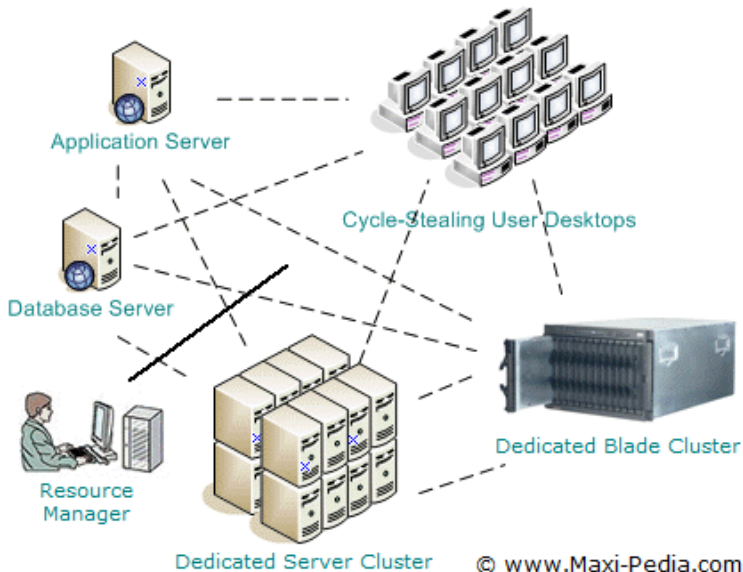
Introduction

What kind of problems a system is faced by such a task?

- 1 how is the software served?
- 2 how can the user/program find and access the data?
- 3 how and where are the different jobs executed?
- 4 can independent jobs run in parallel?
- 5 how are permissions handled?

The solution to all these questions is:

GRID COMPUTING



Definition of *Grid Computing*

IBM defines grid computing as:

“The ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet.

A grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across ‘multiple’ administrative domains based on their (resources) availability, capacity, performance, cost and users’ quality-of-service requirements”.

What is a distributed file system

A distributed file system is:

any **file system** that allows access to files from **multiple hosts** sharing via a **computer network**.

What is a distributed file system

A distributed file system is:

any **file system** that allows access to files from **multiple hosts** sharing via a **computer network**.

Advantages:

- data can easily be shared
- an user can access his home directory from different machines
- central management reduce maintenance workload
- opportunity to use diskless workstations

What is a distributed file system

A distributed file system is:

any **file system** that allows access to files from **multiple hosts** sharing via a **computer network**.

Advantages:

- data can easily be shared
- an user can access his home directory from different machines
- central management reduce maintenance workload
- opportunity to use diskless workstations

disatvantages:

- network traffic
- private data insecure if not proper installed

Samba

Some key points about Samba:

- it is a free reimplementation of the **SMB**-Protocol (Server Message Block)
- it was introduced in 1991 by Andrew Tridgell
- it provides a **printer** and **file service**
- it supports almost all platforms
- it can serve as a Windows Primary Domain Controller

Configuring Samba

The configuration file of the Samba server is: `/etc/samba/smb.conf`

The file is separated into sections and contains 3 special sections:

`global`, `home`, `printer`

Configuring Samba

The configuration file of the Samba server is: `/etc/samba/smb.conf`

The file is separated into sections and contains 3 special sections:

`global`, `home`, `printer`

You can add any other sections to share some folders.

For a section common parameters are:

`comment` hold the label to be given to the share

`path` specify the full path of the share directory

`browseable` whether the client can list the content of the share directory

`writable` whether the client can write something to the share directory

`valid user` define who can access the share sources

Configuring Samba

Samba supports four security levels: share, user, server, and domain.

share most insecure level. (open for everyone)

user uses /etc/passwd for authentication for each user

server same as user but it uses another server to authenticate

domain uses the Windows Primary Domain Controller for authentication

Network File System

Some key points about NFS:

- it builds upon **RPC** (Remote Procedure Call)
- NFS is defined as an **open standard** (everybody can implement it)
- Sun Microsystems original developed NFS in 1984.
- at the beginning it used **UDP**; now the transport is done via **TCP**
- the current version is **NFSv4**

NFS vs Samba

A short comparison between NFS and Samba

NFS vs Samba

A short comparison between NFS and Samba

pros for NFS

- NFS is much faster (due to TCP/IP transmission)
- easily configured and maintained
- less CPU-load for the server

NFS vs Samba

A short comparison between NFS and Samba

pros for NFS

- NFS is much faster (due to TCP/IP transmission)
- easily configured and maintained
- less CPU-load for the server

cons for NFS

- only a very poor security options
- NFS is only natively supported on *uni systems
- only usable in a closed local network (due to security reasons)

Configuring NFS

The configuration of NFS is done in: `/etc/exports`

After each change in `/etc/exports` inform server via `exportfs`

Configuring NFS

The configuration of NFS is done in: `/etc/exports`

After each change in `/etc/exports` inform server via `exportfs`

Each line indicates a volume that is shared and how it is shared

`/etc/exports` Syntax:

```
directory machine1(option11,option12) {machine2(option21,option22)}
```

directory any absolute directory-path on the server

machine an IP or DNS (* can be used as wildcard)

option some examples: `ro,rw,no_root_squash ...`

see *man exports* for details

Other distributed file systems

There are a range of other distributed file systems out there:

- sshfs
 - ▶ uses **ssh** to connect to remote server
 - ▶ mounted folder is usable as it is locally
 - ▶ not suitable in **multi-user environments**

Other distributed file systems

There are a range of other distributed file systems out there:

- sshfs
 - ▶ uses **ssh** to connect to remote server
 - ▶ mounted folder is usable as it is locally
 - ▶ not suitable in **multi-user environments**

- AFS (Andrew File System)
 - ▶ uses Kerberos for authentication
 - ▶ caches often used files on localhost to reduce network traffic
 - ▶ supports mirroring for read-only files

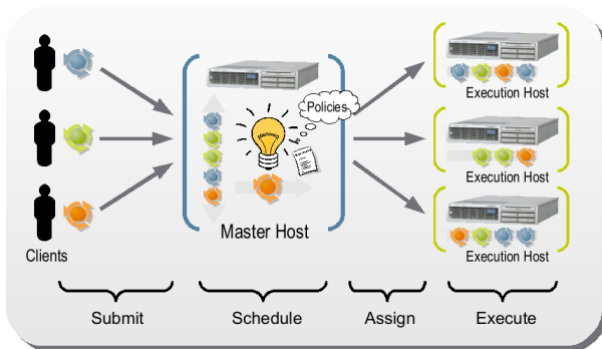
Other distributed file systems

There are a range of other distributed file systems out there:

- sshfs
 - ▶ uses **ssh** to connect to remote server
 - ▶ mounted folder is usable as it is locally
 - ▶ not suitable in **multi-user environments**
- AFS (Andrew File System)
 - ▶ uses Kerberos for authentication
 - ▶ caches often used files on localhost to reduce network traffic
 - ▶ supports mirroring for read-only files
- GlusterFS
 - ▶ GlusterFS **aggregates** various storage servers **over Ethernet**
 - ▶ File-based mirroring and replication
 - ▶ File-based striping supported
 - ▶ File-based **load balancing**
 - ▶ Volume failover

What is a grid engine

A grid engine is also call a *Distributed Resource Management* system



What is a grid engine

A DRM/grid engine should provide:

- Policy based allocation of distributed resources
- Batch queuing & scheduling
- Load balancing & remote job execution
- Detailed job accounting statistics
- Fine-grained user specifiable resources
- Suspend/resume/migrate jobs
- Tools for reporting Job/Host/Cluster status
- Job Arrays
- Integration & tight control of parallel jobs

Open Grid Scheduler

Some key points about the Open Grid Scheduler

- original developed under the name **Sun Grid Engine** by *Sun Microsystems*
- in 2010, after the purchase of *Sun* by *Oracle*, the community started the open source project **Open Grid Scheduler**
- normally use for high-performance computing clusters

Open Grid Scheduler

Some key points about the Open Grid Scheduler

- original developed under the name **Sun Grid Engine** by *Sun Microsystems*
- in 2010, after the purchase of *Sun* by *Oracle*, the community started the open source project **Open Grid Scheduler**
- normally use for high-performance computing clusters
- if you used the BIM-Cluster, you used already the SGE

Example of a job submission on SGE

A user can specify **needed resources**. (eg. cores, memory)

→ The job will run only if the requirements can be met.

Example of a job submission on SGE

A user can specify **needed resources**. (eg. cores, memory)

→ The job will run only if the requirements can be met.

A job is executed if a **slot is free** and the job has the **highest priority**

→ Different profiles can be used for calculating a priority

Example of a job submission on SGE

A user can specify **needed resources**. (eg. cores, memory)

→ The job will run only if the requirements can be met.

A job is executed if a **slot is free** and the job has the **highest priority**

→ Different profiles can be used for calculating a priority

Output- and errorstream is commonly redirected to the users \$HOME

→ Therefore commonly the \$HOME is mounted on all nodes

Example of a job submission on SGE

A user can specify **needed resources**. (eg. cores, memory)

→ The job will run only if the requirements can be met.

A job is executed if a **slot is free** and the job has the **highest priority**

→ Different profiles can be used for calculating a priority

Output- and errorstream is commonly redirected to the users \$HOME

→ Therefore commonly the \$HOME is mounted on all nodes

Most of the time user want to run his new developed code in his \$HOME

→ same as before

Example of a job submission on SGE

```
#!/bin/bash

### SGE Options
#$-P short_proj
#$-l vf=3800m
#$-S /bin/bash
#$-pe serial 1
#$-e $HOME/errors/error_${TASK_ID}_${JOB_ID}
#$-o $HOME/results/results_${TASK_ID}_${JOB_ID}
#$-cwd $HOME/executenDir

# program to execute
java -Xmx3700m mainClass -input $TASK_ID
```


Example of a job submission on SGE

```
#!/bin/bash

### SGE Options
#$-P short_proj
#$-l vf=3800m
#$-S /bin/bash
#$-pe serial 1
#$-e $HOME/errors/error_${TASK_ID}_${JOB_ID}
#$-o $HOME/results/results_${TASK_ID}_${JOB_ID}
#$-cwd $HOME/executenDir

# program to execute
java -Xmx3700m mainClass -input $TASK_ID
```

submitting the skript

```
@master$ qsub -t 1-100:2 script_name.sh
```

Bibliography

chrisdag. Grid Engine Administration - Training Slides.

<http://bioteam.net/2009/09/sge-training-slides/>, Sep 2009.

N. Ferguson. An Introduction to NFS.

<http://linuxvm.org/present/SHARE98/S5521NFa.pdf>, Mar 2002.

C. Kesselman and I. Foster. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Nov. 1998. ISBN 1558604758.

URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1558604758>.

D. Y. Leung and D. R. Hou. An Introduction to Samba Server. http://www.comp.hkbu.edu.hk/~comp2650/tutorial/notes/lab_notes_8.pdf, Jun 2012.

C. Smith. Linux NFS-HOWTO. <http://nfs.sourceforge.net/nfs-howto/>, May 2006.