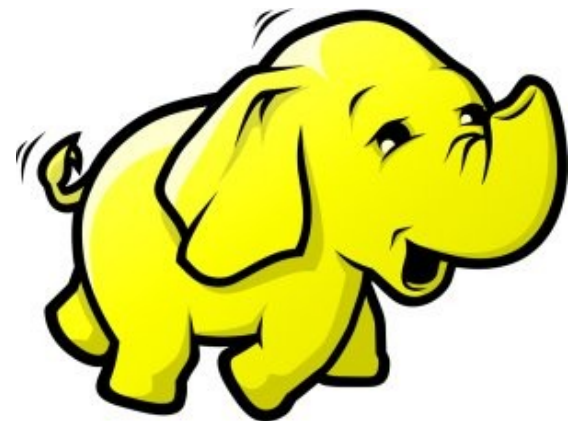




# Hadoop 叢集設定解說

## Setup Hadoop Fully Distributed Mode

**Jazz Wang**  
**Yao-Tsung Wang**  
**jazz@nchc.org.tw**



# Yahoo's Hadoop Cluster

## 雅虎的大象軍團

- ~10,000 machines running Hadoop in US
- The largest cluster is currently 2000 nodes
- Nearly 1 petabyte of user data (compressed, unreplicated)
- Running roughly 10,000 research jobs / week



# Hadoop Pseudo-Distributed Mode

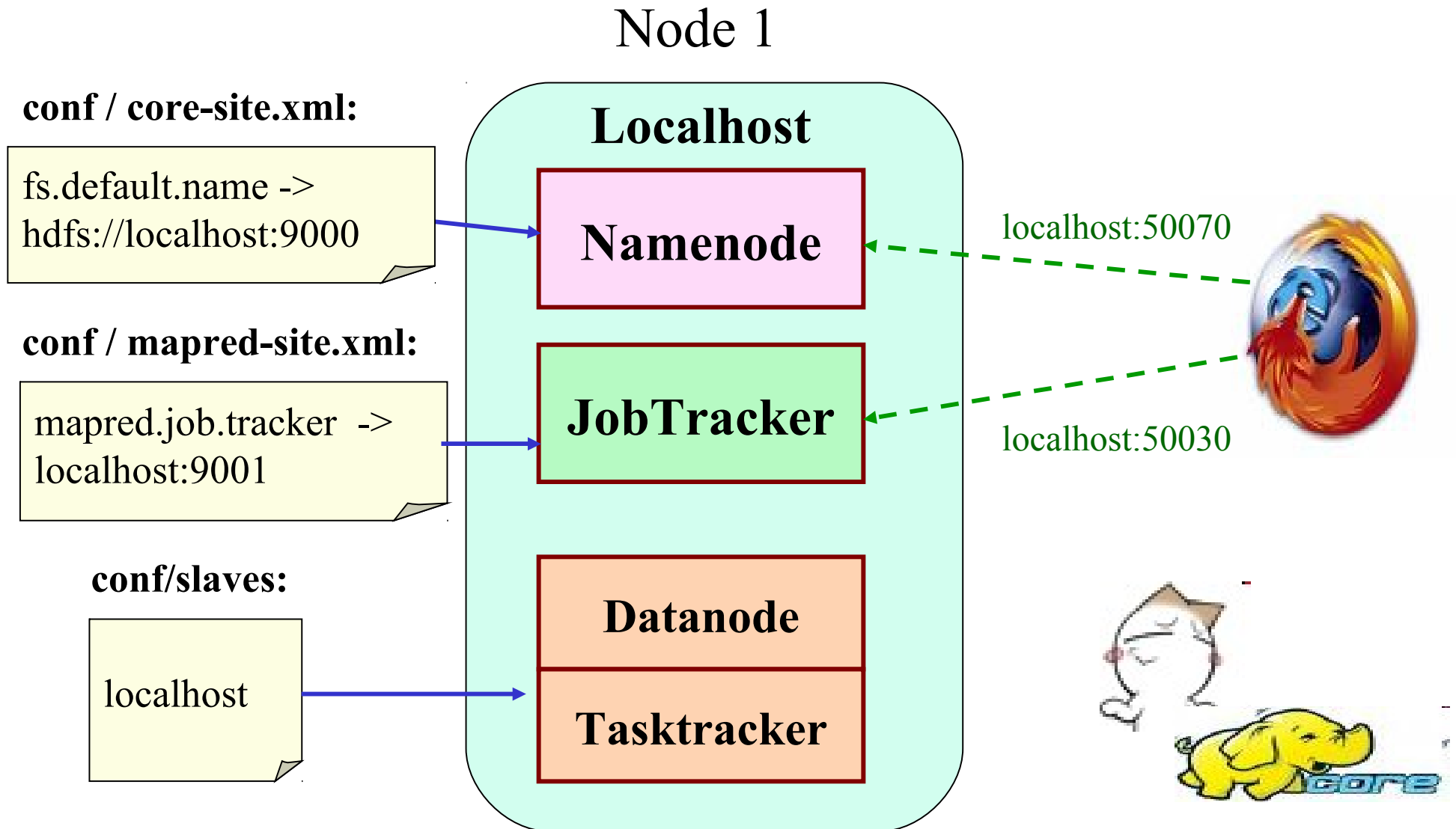
## 我們已經實作過單機模式

- Step 1: Setup SSH key exchange
- Step 2: Install Java
- Step 3: Download Hadoop Source Package
- Step 4: Configure `hadoop-env.sh`
  - `export JAVA_HOME=/usr/lib/jvm/java-6-sun`
- Step 5: Configure `*-site.xml`
  - Set Namenode to `hdfs://localhost:9000`
  - Set Jobtracker to `localhost:9001`
  - `bin/hadoop namenode -format`
- Step 6: Format HDFS
- Step 7: Start Hadoop
  - `bin/start-all.sh`
- Step 8: Complete!! Let's check the status of Hadoop
  - Job admin <http://localhost:50030/> HDFS <http://localhost:50070/>



# Diagram of Pseudo-Distributed Mode

## Hadoop 單機環境示意圖



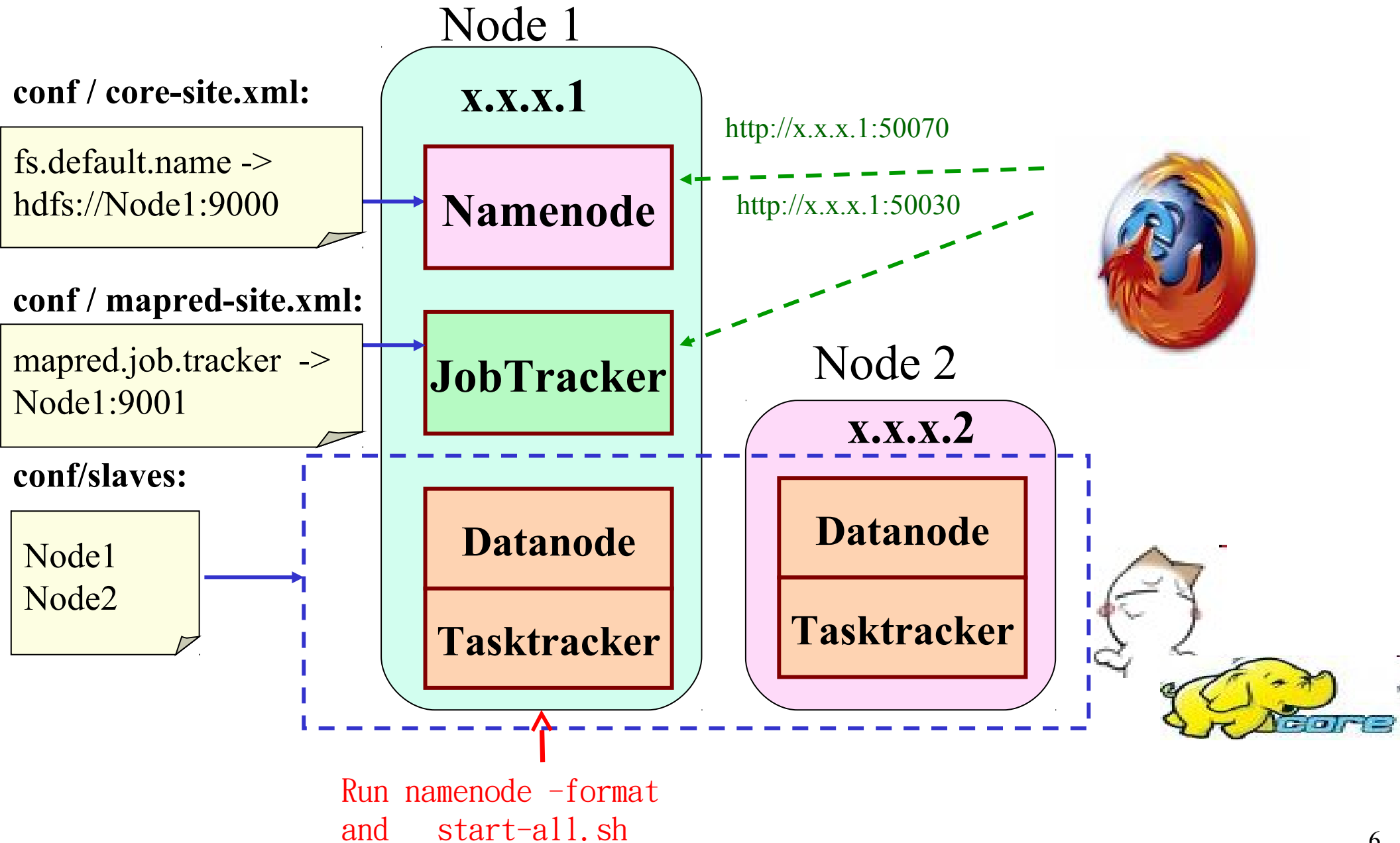
# Hadoop Fully-Distributed Mode

## 我們接著要用兩台電腦實作叢集模式

- Step 1: Setup SSH key exchange
- Step 2: Install Java
- Step 3: Download Hadoop Source Package
- Step 4: Configure `hadoop-env.sh`
  - `export JAVA_HOME=/usr/lib/jvm/java-6-sun`
- **Step 5: Configure `*-site.xml`**
  - **Set Namenode to `hdfs://x.x.x.1:9000`**
  - **Set Jobtracker to `x.x.x.2:9001`**
- **Step 6: Configure Slaves**
- **Step 7: Synchronization of all slaves**
- Step 8: Format HDFS
  - `bin/hadoop namenode -format`
- Step 9: Start Hadoop
  - **On NameNode : `bin/start-dfs.sh`**
  - **On JobTracker : `bin/start-mapred.sh`**
- Step 10: Complete!! Let's check the status of Hadoop
  - Job admin `http://x.x.x.2:50030/` HDFS `http://x.x.x.1:50070/`

# Use case #1

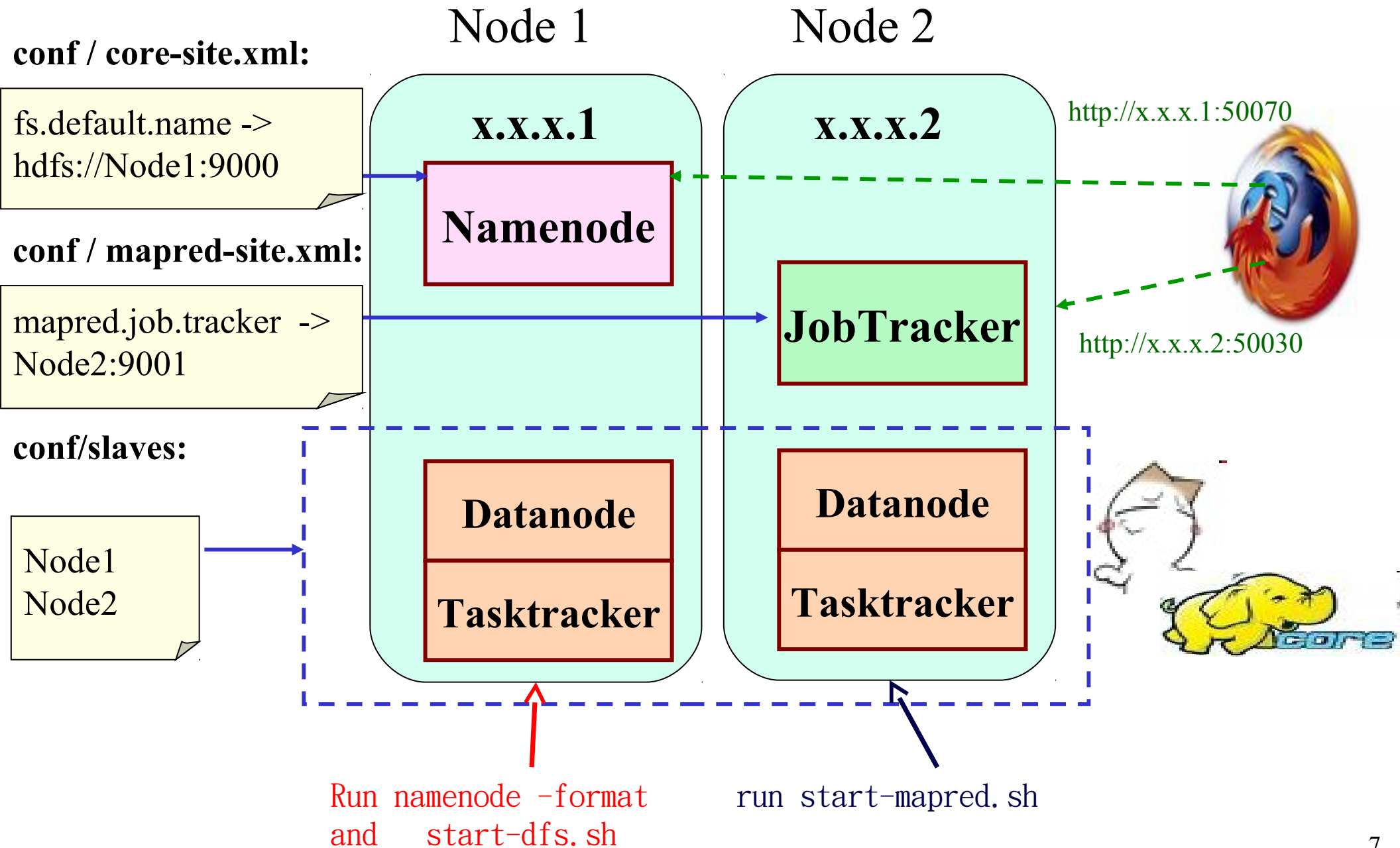
## 設定情境一





# Use case #2

## 設定情境二



# Use case #3

## 設定情境三

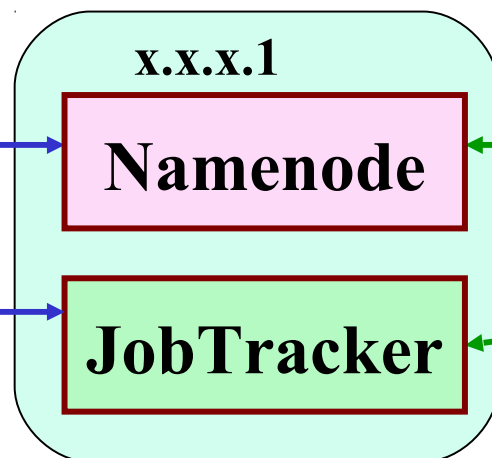
**conf / core-site.xml:**

fs.default.name ->  
hdfs://Node1:9000

**conf / mapred-site.xml:**

mapred.job.tracker ->  
Node1:9001

Node 1



http://x.x.x.1:50070

http://x.x.x.1:50030



**conf/slaves:**

Node2  
.....  
NodeN

Node 2

**x.x.x.2**

**Datanode**

**Tasktracker**

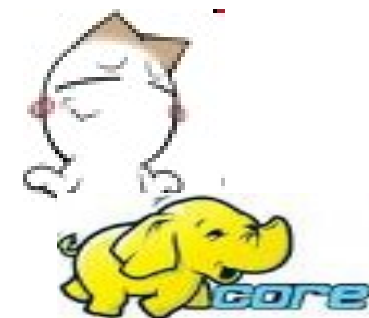
...

Node N

**x.x.x.n**

**Datanode**

**Tasktracker**





# Use case #4

## 設定情境四

### conf / core-site.xml:

fs.default.name ->  
hdfs://Node1:9000

Client

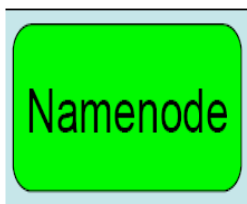


http://x.x.x.2:50030

### conf / mapred-site.xml:

mapred.job.tracker ->  
Node2:9001

G

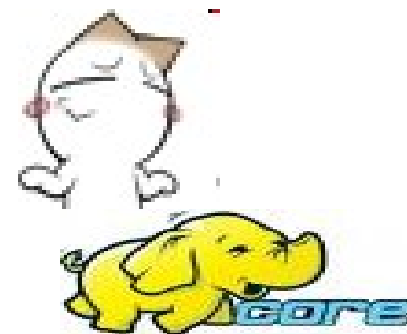
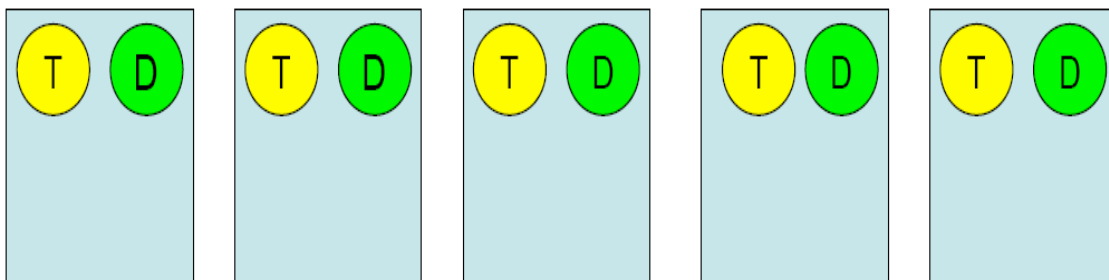


HTTP Monitoring UI

http://x.x.x.1:50070

### conf/slaves:

Node3  
.....  
NodeN

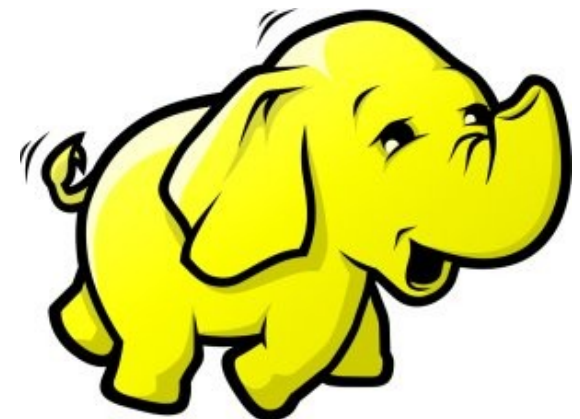




# Hadoop 叢集佈署工具

Hadoop Deployment Tool : SmartFog and DRBL

**Jazz Wang**  
**Yao-Tsung Wang**  
**jazz@nchc.org.tw**



# Programmer v.s. System Admin.



Source: <http://www.funnyjunksite.com/wp-content/uploads/2007/08/programmer.jpg>



Source: <http://www.sysadminday.com/images/people/136-3697.JPG>



**PART 1**

# PC Cluster 101

**Jazz Wang**

**Yao-Tsung Wang**

**[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)**



Powered by **DRBL**



At First, We have “4 + 1” PC Cluster

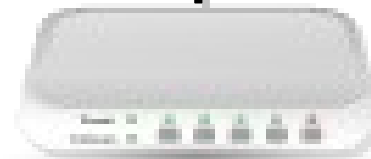
It'd better be  
**2<sup>n</sup>**



Manage  
**Scheduler**

**Then, We connect 5 PCs with  
Gigabit Ethernet Switch**

**GiE Switch**



**10/100/1000  
Mbps**

**WAN**



**Add 1 NIC  
for WAN**





## Compute Nodes

**4 Compute Nodes** will communicate via **LAN Switch**. Only **Manage Node** have **Internet Access** for Security!



WAN



Manage Node

# Compute Nodes

## Basic System Setup for Cluster

Messaging

**MPICH**

**GCC**

**Bash**

**Perl**

Account Mgmt.

**SSHD**

**NIS**

**YP**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

On **Manage Node**,  
We need to install **Scheduler** and  
**Network File System** for sharing  
Files with **Compute Node**

Job Mgmt.

**OpenPBS**

File Sharing

**NFS**

**Extra**

Messaging

MPICH

GCC

Bash

Perl

Account Mgmt.

SSHD

NIS

YP

GNU Libc



Kernel Module

Linux Kernel

Boot Loader

# Challenges of Cluster Computing

- **Hardware**

- **Ethernet Speed / PC Density**
- **Power / Cooling / Heat**
- **Network and Storage Architecture**

- **Software**

- **Job Scheduler ( Cluster level )**
- **Account Management**
- **File Sharing / Package Management**

- **Limitation**

- **Shared Memory**
- **Global Memory Management**

# Common Method to deploy Cluster



**1. Setup one  
Template  
machine**

**2. Cloning  
to  
multiple  
machine**



**3. Configure  
Settings**



**4. Install  
Job  
Scheduler**



**5. Running  
Benchmark**

# Challenges of Common Method

**Add New User Account ?**

**Upgrade Software ?**

**How to share user data ?**

**Configuration Synchronization**



# How to deploy 4000+ Nodes ????

資料標題：Scaling Hadoop to 4000 nodes at Yahoo!

資料日期：September 30, 2008

<b>Total Nodes</b>	<b>4000</b>
<b>Total cores</b>	<b>30000</b>
<b>Data</b>	<b>16PB</b>

	<b>500-node cluster</b>		<b>4000-node cluster</b>	
	<b>write</b>	<b>read</b>	<b>write</b>	<b>read</b>
<b>number of files</b>	990	990	14,000	14,000
<b>file size (MB)</b>	320	320	360	360
<b>total MB processes</b>	316,800	316,800	5,040,000	5,040,000
<b>tasks per node</b>	2	2	4	4
<b>avg. throughput (MB/s)</b>	<b>5.8</b>	<b>18</b>	<b>40</b>	<b>66</b>

# Advanced Methods to deploy Cluster

- **SSI ( Single System Image )**
  - **Multiple PCs as Single Computing Resources**
  - **Image-based**
    - **homogeneous**
    - **ex. SystemImager, OSCAR, Kadeploy**
  - **Package-based**
    - **heterogeneous**
    - **easy update and modify packages**
    - **ex. FAI, DRBL**
- **Other deploy tools**
  - **Rocks : RPM only**
  - **cfengine : configuration engine**

# Comparison of Cluster Deploy Tools

	Distribution	Support Diskless/ Sysmless	Type	Node configuration tools	Cluster management tools	Database installation
<b>System Imager</b>	<b>ALL</b>	<b>Yes</b>	<b>Image</b>	<b>Yes</b>	<b>No</b>	<b>No</b>
<b>OSCAR</b>	<b>RPM- based</b>	<b>Yes</b>	<b>Image</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>
<b>Kadeploy</b>	<b>ALL</b>	<b>No</b>	<b>Image</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>
<b>DRBL</b>	<b>ALL</b>	<b>Yes</b>	<b>Package</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>
<b>FAI</b>	<b>Debian- Based</b>	<b>Yes</b>	<b>Package</b>	<b>Yes</b>	<b>No</b>	<b>No</b>



**PART 2-1 :**

# Hadoop Deployment Tool

**Jazz Wang**

**Yao-Tsung Wang**

**[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)**



Powered by **DRBL**



- Make Hadoop deployment *agile*
- Integrate with dynamic cluster deployments

Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)

12 June 2008

# SmartFrog - HPLabs' CM tool

- Language for describing systems to deploy  
—everything from datacentres to test cases
  - Runtime to create *components* from the model
  - Components have a lifecycle
  - LGPL Licensed, Java 5+
- <http://smartfrog.org/>

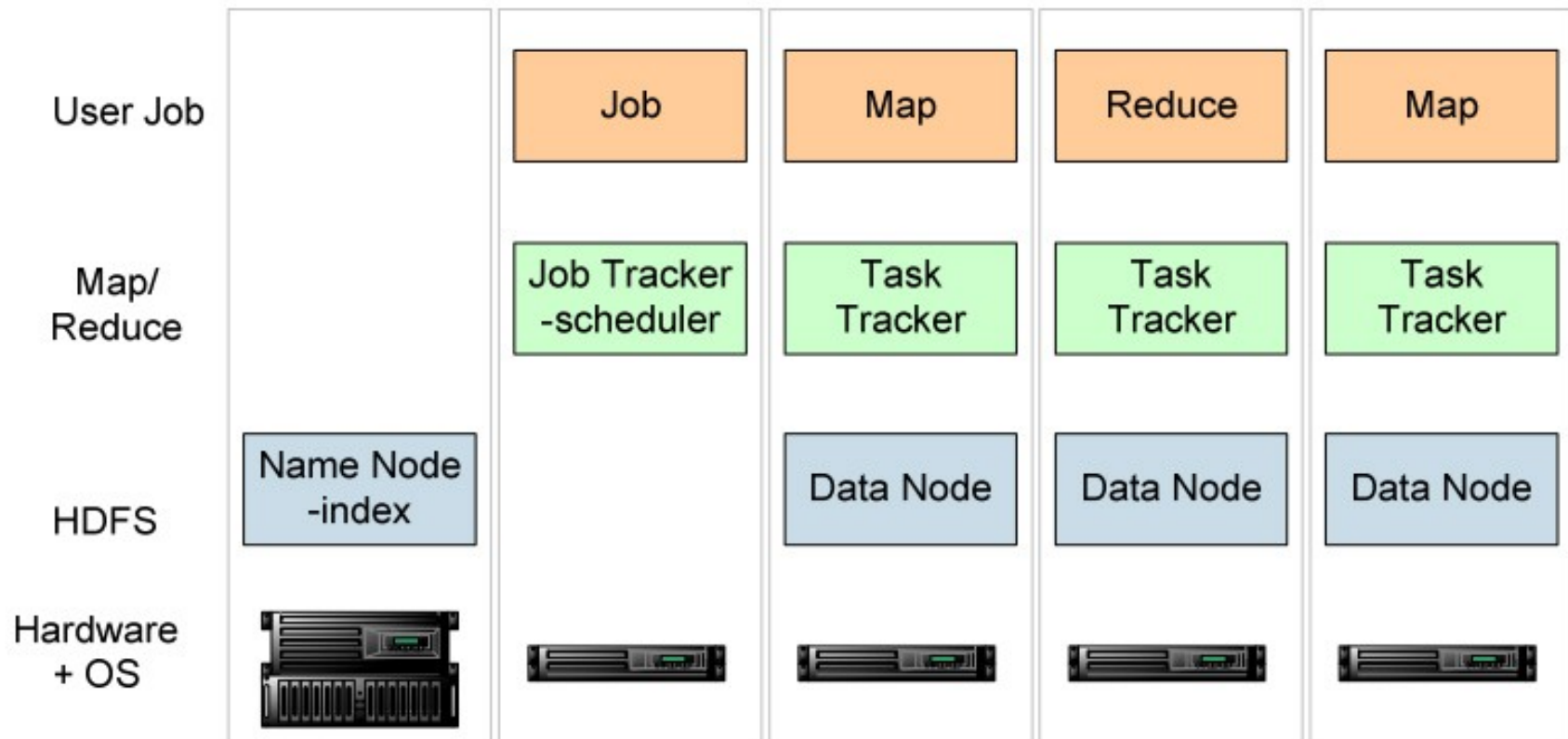
Source: Deploying hadoop with smartfrog

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)





# Basic problem: deploying Hadoop



*one namenode, 1+ Job Tracker, many data nodes and task trackers*

**Source: Deploying hadoop with smartfrog**

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)

# The hand-managed cluster

- Manual install onto machines
- SCP/FTP in Hadoop zip
- copy out hadoop-site.xml and other files
- edit /etc/hosts, /etc/rc5.d, SSH keys ...
- Installation scales  $O(N)$
- Maintenance, debugging scales worse

Source: Deploying hadoop with smartfrog

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# The locked-down cluster

- PXE Preboot of OS images
- RedHat Kickstart to serve up (see [instalinux.com](http://instalinux.com))
- Maybe: LDAP to manage state, or custom RPMs

## Requires:

uniform images, central LDAP service, good ops team, stable configurations, home-rolled RPMs

Source: [Deploying hadoop with smartfrog](#)

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# CM-tool managed cluster

## Configuration Management tools

- State Driven: observe system state, push it back into the desired state
- Workflow: apply a sequence of operations to change a machine's state
- Centralized: central DB in charge
- Decentralized: machines look after themselves

CM tools are the only way to manage big clusters

Source: [Deploying hadoop with smartfrog](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# Model the system in the SmartFrog language

```
TwoNodeHDFS extends OneNodeHDFS {  
  
    localDataDir2 extends TempDirwithCleanup {  
  
    }  
  
    datanode2 extends datanode {  
        dataDirectories [LAZY localDataDir2];  
        dfs.datanode.https.address "https://localhost:0";  
    }  
}
```

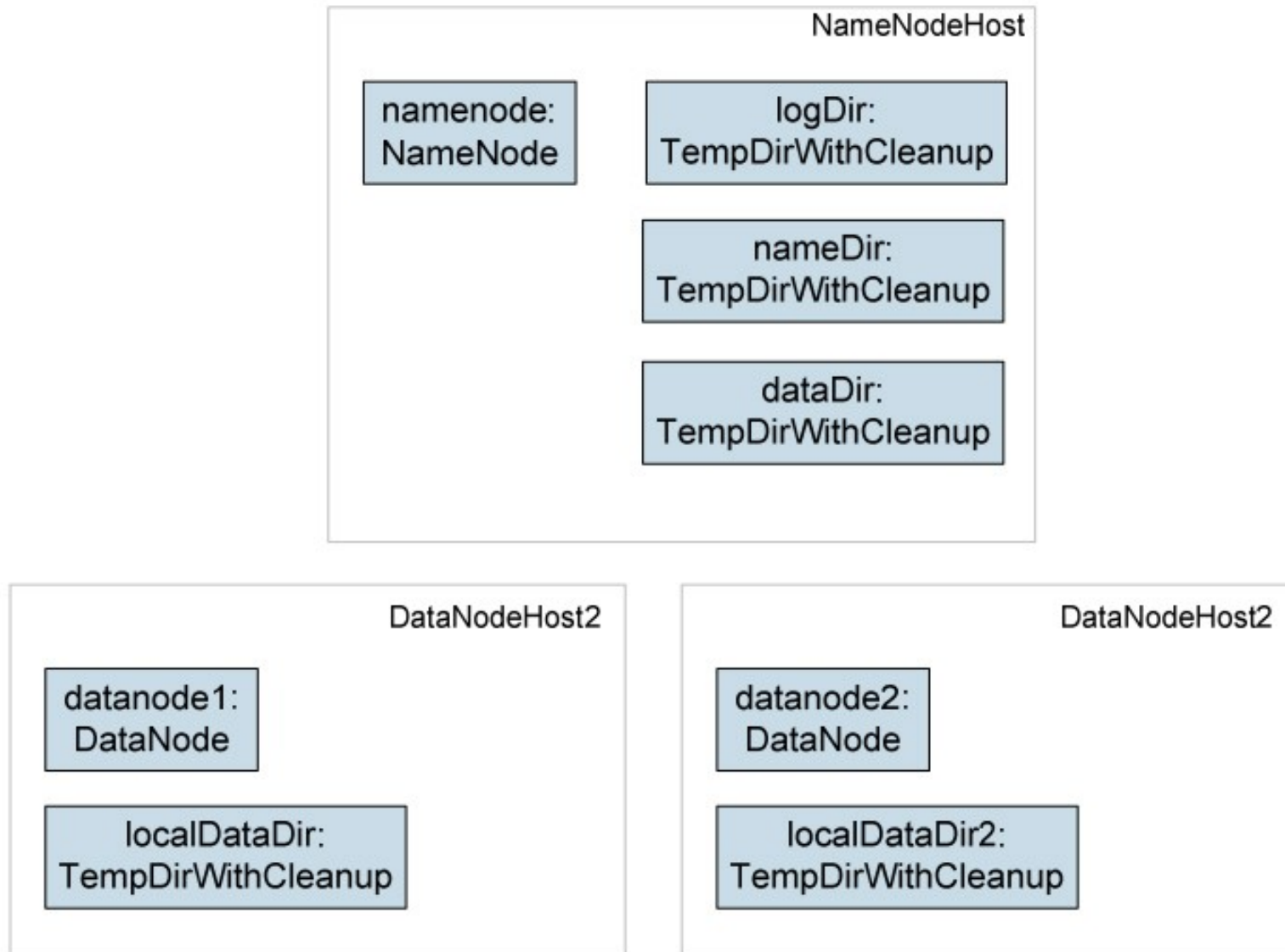
Inheritance, cross-referencing, templating

Source: [Deploying hadoop with smartfrog](#)

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# The runtime deploys the model



Source: Deploying hadoop with smartfrog

[http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)



# Steps to deployability

1. Configure Hadoop from an SmartFrog description
2. Write components for the Hadoop nodes
3. Write the functional tests
4. Add *workflow* components to work with the filesystem; submit jobs
5. Get the tests to pass

Source: Deploying hadoop with smartfrog

12 [http://people.apache.org/~stevell/slides/deploying\\_hadoop\\_with\\_smartfrog.pdf](http://people.apache.org/~stevell/slides/deploying_hadoop_with_smartfrog.pdf)





**PART 2-2 :**

# Introduction to DRBL

**Jazz Wang**

**Yao-Tsung Wang**

**[jazz@nchc.org.tw](mailto:jazz@nchc.org.tw)**



Powered by **DRBL**

# What is DRBL ??

- **Diskless Remote Boot in Linux**
- Network is cheap, and our time is expansive
- In simple words, DRBL is .....
  - Replace IDE/SATA cable with network cable
  - 40+ student PCs connected to one DRBL server



**Diskfull  
PC**



=



+



+



**Diskless  
PC**



**Server**

# 1st, We install Base System of **GNU/Linux** on **Management Node**.

**You can choose:**

**Redhat, Fedora, CentOS, Mandriva,  
Ubuntu, Debian, ...**



2nd, We install **DRBL package** and  
configure it as **DRBL Server**.

There are lots of service needed:  
**SSHD, DHCPD, TFTP, NFS Server,**  
**NIS Server, YP Server ...**

Network Booting

Account Mgmt.

**NFS**

**TFTP**

**DHCPD**

**SSHD**

**NIS**

**YP**

**Perl**

**Bash**

**GNU Libc**

**DRBL Server**

based on existing  
Open Source and  
keep Hacking!

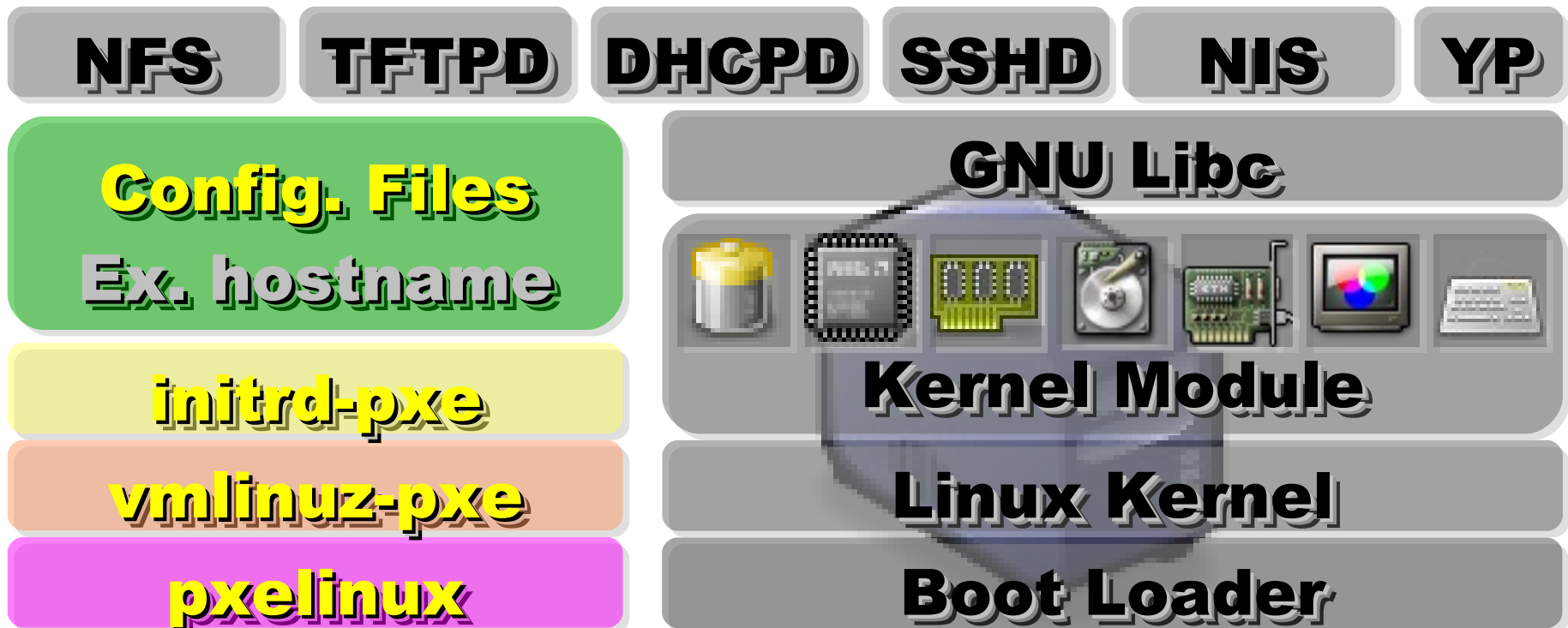


**Kernel Module**

**Linux Kernel**

**Boot Loader**

After running “**drblsrv -i**” & “**drblpush -i**”, there will be **pxelinux**, **vmlinux-pex**, **initrd-pxe** in TFTPROOT, and different **configuration files** for each Compute Node in NFSROOT



3rd, We enable **PXE** function in **BIOS** configuration.

**BIOS PXE**

**BIOS PXE**

**BIOS PXE**

**BIOS PXE**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

**Config. Files**

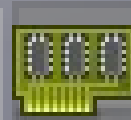
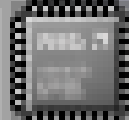
**Ex. hostname**

**initrd-pxe**

**vmlinux-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**



While Booting, **PXE** will query IP address from **DHCPD**.

**BIOS PXE**

**BIOS PXE**

**BIOS PXE**

**BIOS PXE**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

**Config. Files**  
**Ex. hostname**

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**



While Booting, **PXE** will query IP address from **DHCPD**.

**IP 1**

**IP 2**

**IP 3**

**IP 4**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

**Config. Files**  
**Ex. hostname**

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**

After PXE get its IP address, it will download booting files from **TFTPD**.

**IP 1**

**IP 2**

**IP 3**

**IP 4**

**NFS**

**TFTPD**

**DHCPD**

**SSHD**

**NIS**

**YP**

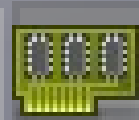
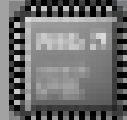
**Config. Files**  
**Ex. hostname**

**initrd-pxe**

**vmlinuz-pxe**

**pxelinux**

**GNU Libc**



**Kernel Module**

**Linux Kernel**

**Boot Loader**



NFS **TFTPD** DHCPD SSHD NIS YP

Config. Files  
Ex. hostname

initrd-pxe

vmlinuz-pxe

pxelinux

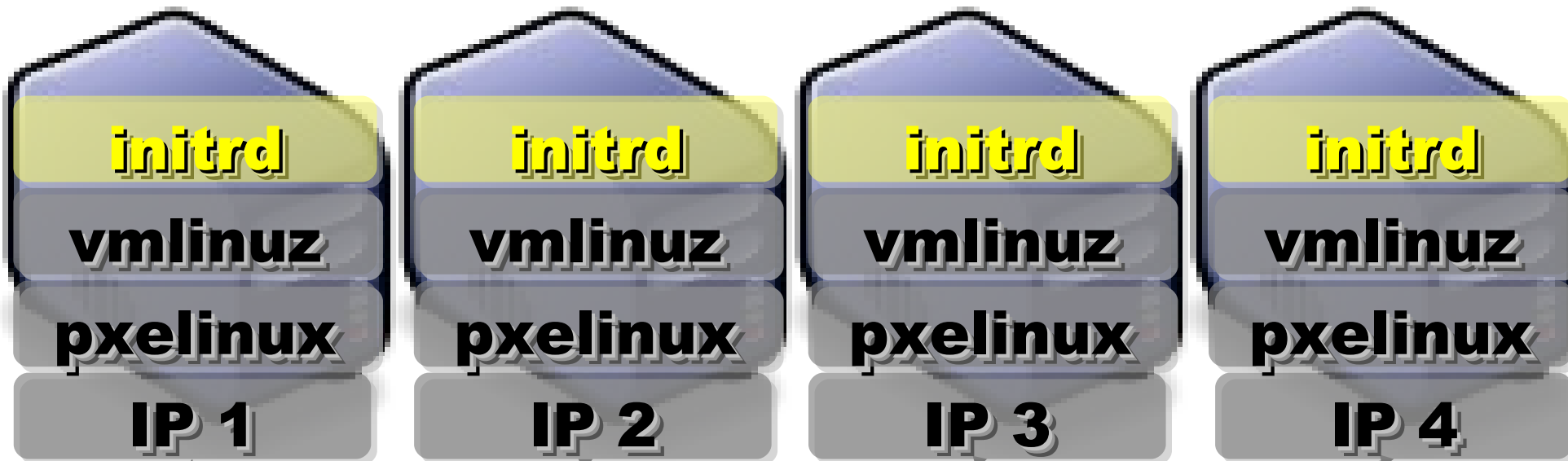
GNU Libc



Kernel Module

Linux Kernel

Boot Loader



Config. Files  
GNU Libc

**After downloading booting files, scripts in **initrd-pxe** will config **NFSROOT** for each Compute Node.**





**NFS**    **TFTPD**    **DHCPD**    **SSHD**    **NIS**    **YP**

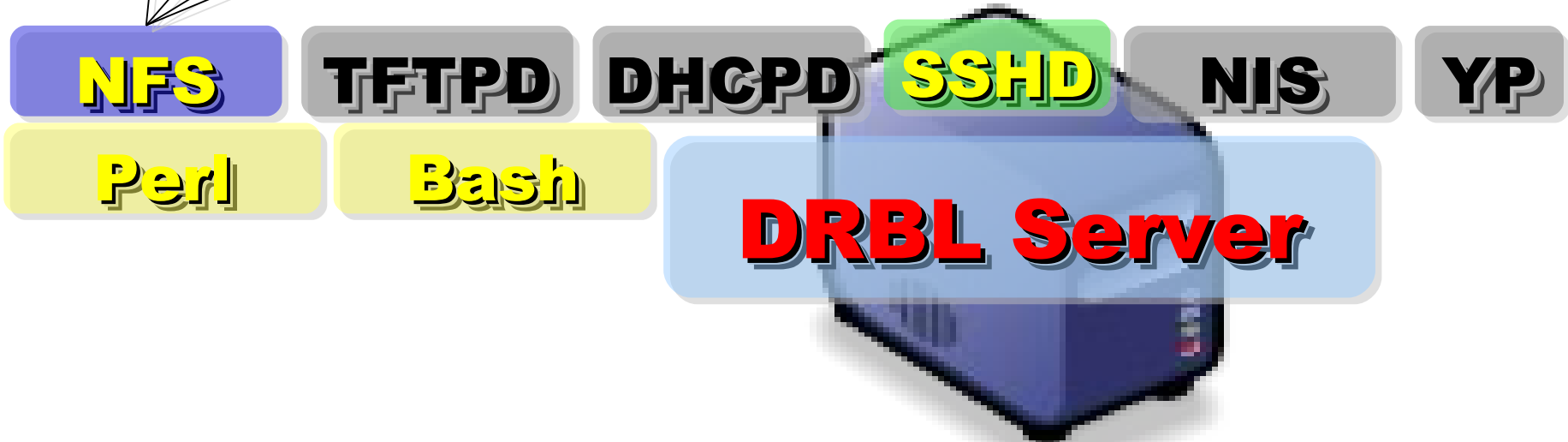
**Config. Files**  
Ex. hostname

initrd-pxe  
vmlinuz-pxe  
pxelinux





**Applications and Services** will also  
deployed to each **Compute Node**  
via **NFS** ....





With the help of **NIS** and **YP**,  
You can login each Compute Node  
with the **Same ID / PASSWORD**  
stored in **DRBL Server!**

SSH Client





## Questions?

Slides - <http://trac.nchc.org.tw/cloud>

**Jazz Wang**  
**Yao-Tsung Wang**  
**jazz@nchc.org.tw**



Powered by DRBL