Overview of MOSIX

Prof. Amnon Barak
Computer Science Department
The Hebrew University

http:// www.MOSIX.org

Background

Clusters and multi-cluster private Clouds are popular platforms for running distributed applications

In most cases, users want to run multiple jobs concurrently, with minimal burden how the resources are managed

- **Users prefer not to:**
  - **Modify applications**
  - **Copy files or login to different nodes**
  - **Lose jobs when some nodes are disconnected**
- **Users don't know (and doesn't care):**
  - **What is the configuration, the status and the locations of the nodes**
  - **Availability of resources, e.g. CPU speed, load, free memory, etc.**

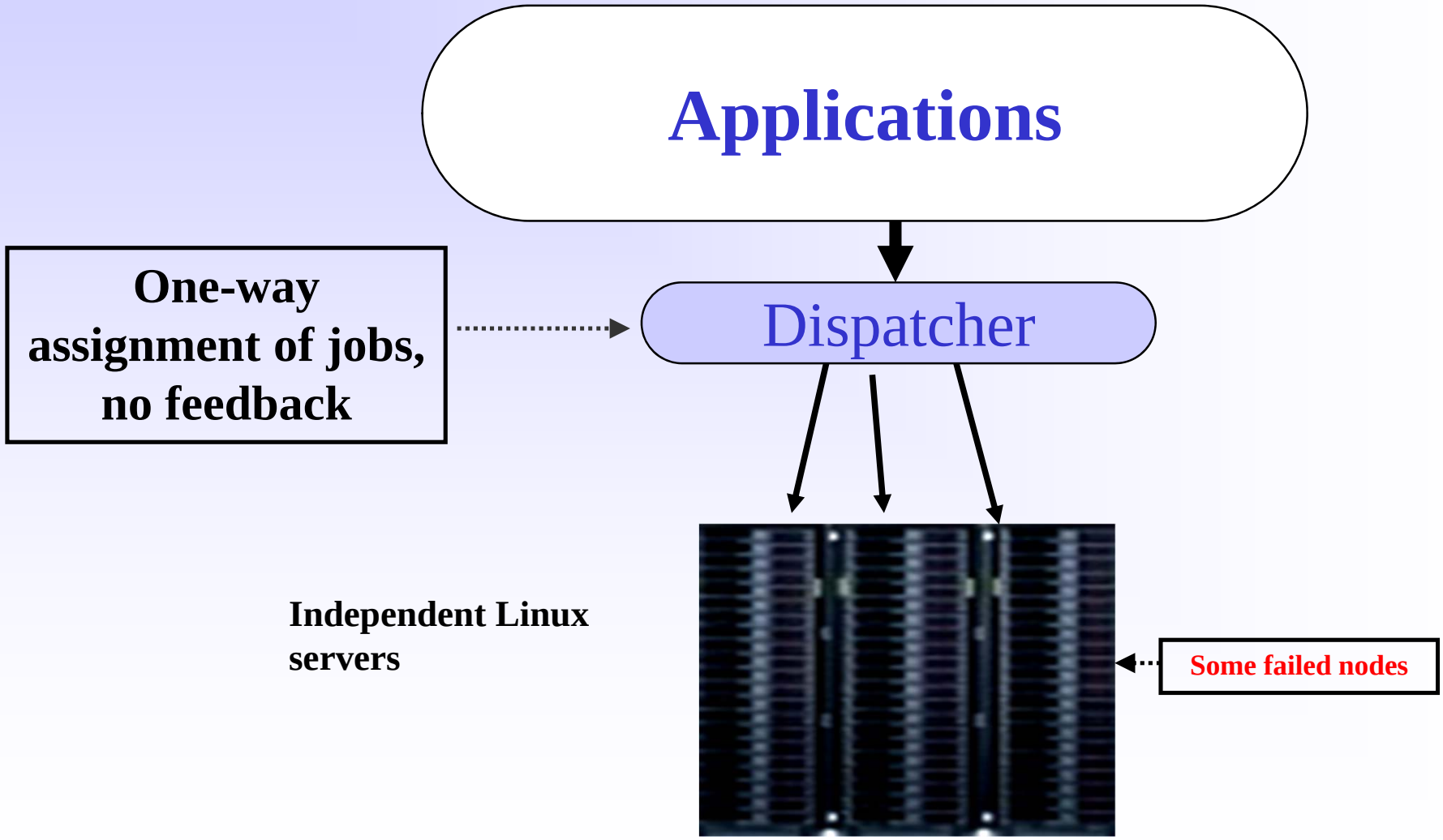
Traditional management packages

Most cluster management packages are batch dispatchers that place the burden of management on users

Some examples:

- **Use static assignment of jobs to nodes**
  - **May lose jobs when nodes are disconnected**
  - **May lose overdue jobs**
- **Not transparent to applications**
  - **May require to link application with special libraries**
  - **View the cluster as a set of independent nodes**
    - **One user per node, cluster partition for multi-users**

Traditional management packages



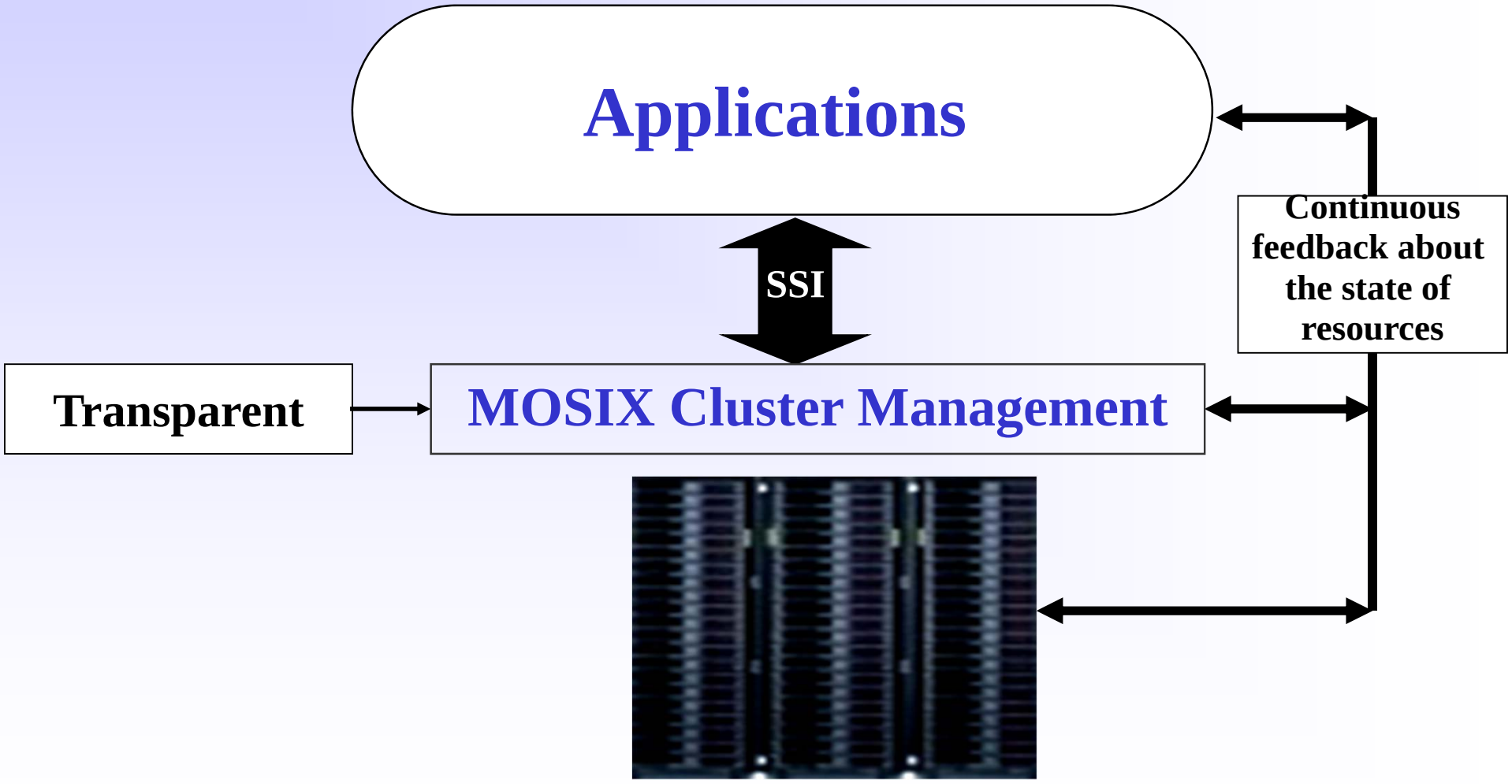
What is MOSIX (Multi-computer OS)

A cluster management system for Linux clusters and multi-clusters Clouds

Main feature: Single-Systems Image (SSI)

- **Users can login on any node and need not know where their programs run**
- **Automatic resource discovery**
  - **By continuous monitoring of the state of the resources**
- **Dynamic workload distribution by process migration**
  - **Automatic load-balancing**
  - **Automatic migration from slower to faster nodes and from nodes that run out of free memory**

MOSIX is a unifying management layer



In a MOSIX cluster

All the active nodes run like one server with many cores

MOSIX Version 4 (MOSIX-4)

- **Geared for distributed, concurrent computing, especially for running application with moderate amounts of I/O**
- **Main features:**
  - **Provides a SSI by process migration**
  - **Process migration within a cluster and among different clusters**
  - **Secure run time environment (sandbox) for guest processes**
  - **Supports checkpoint and recovery**

MOSIX processes

- **Applications that can benefit from migration**
  - **Created by the ``mosrun'‘ command**
  - **Processes are started from standard Linux executables, but run in an environment that allows each process to migrate from one node to another**
  - **Each MOSIX process has a unique home-node, which is usually the node in which the process was created**

Examples: running interactive jobs

Possible ways to run *myprog*:

- ***myprog*** **- run as a Linux process on the local node**
- **mosrun** ***myprog*** **- run as a MOSIX process in the local cluster**
- **mosrun -b** ***myprog*** **- assign the process to the least loaded node**
- **mosrun -b –m700** ***myprog*** **- assign the process only to a nodes with 700MB of free memory**

How does it work

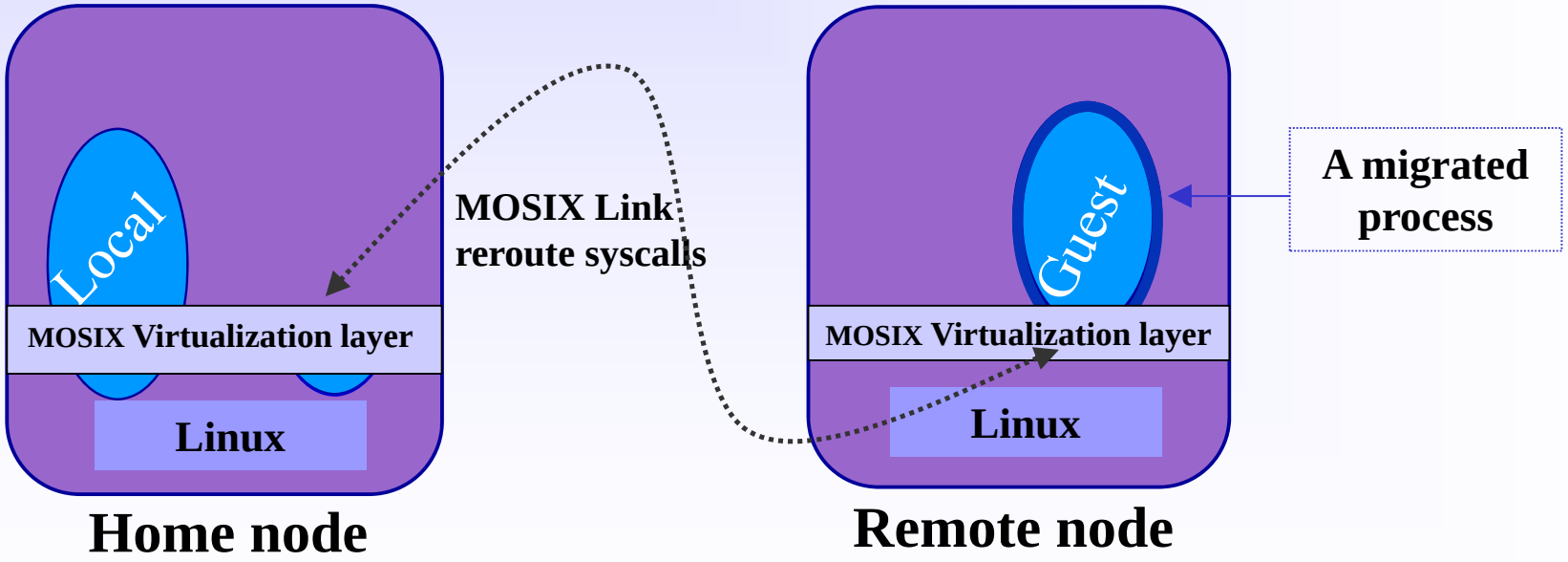
- Automatic resource discovery by a “gossip” algorithm
  - Provides each node with the latest info about the cluster/multi-cluster resources (e.g free nodes)
  - All the nodes disseminate information about relevant resources: speed, load, memory, local/remote I/O, IPC
  - Info exchanged in a random fashion - to support scalable configurations and overcome failures
  - Useful for high volume transaction processing
    - Example: a compilation farm - assign the next compilation to least loaded node

Dynamic workload distribution

- A set of algorithms that match between required and available resources
  - Geared to maximize the performance
  - Initial allocation of processes to the best available nodes in the user's local cluster
    - Not to nodes outside the local cluster
  - Multi-cluster-wide process migration
    - Automatic load-balancing
    - Automatic migration from slower to faster nodes
    - Authorized processes move to idle nodes in other clusters
- Outcome: users need not know the current state of the cluster and the multi-cluster resources

Core technologies

- **Process migration – move the process context to a remote node**
- **The MOSIX virtualization layer allow migrated processes to run in remote nodes, away from their creation (home) nodes**



The MOSIX virtualization layer

- **Provides the necessary support for migrated processes**
  - **By intercepting and forwarding most system-calls to the home node**
- **Result: migrated processes seem to be running in their respective home nodes**
  - **The user's home-node environment is preserved**
  - **No need to change applications, copy files or login to remote nodes or to link applications with any library**
  - **Migrated processes run in a sandbox**

Outcome: users get the illusion of running on one node

- **Drawback: increased communication and virtualization overheads**
  - **Average overhead ~1% (over GEthernet) -reasonable vs. added cluster/multi-cluster services**

Main multi-cluster features

- Administrating a multi-cluster
- Priorities among different clusters
- Monitoring
- Supports checkpoint and recovery
- Supports disruptive configurations

Administrating a multi-cluster

A collection of clusters, servers and workstations whose owners wish to cooperate from time to time

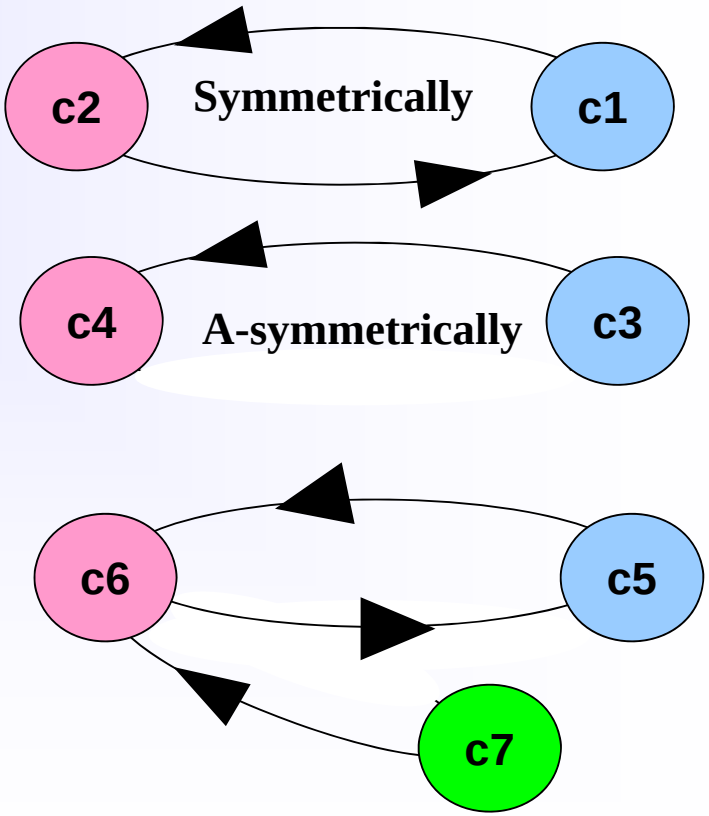
- **Collectively administrated**
  - **Each owner maintains its private cluster**
    - **Determine the priorities vs. other clusters**
  - **Clusters can join or leave the multi-cluster at any time**
  - **Dynamic partition of nodes to private virtual clusters**
  - **Users of a group access the multi-cluster via their private clusters and workstations**
- **Process migration among different cluster**

Outcome: each cluster and the multi-cluster performs like a single computer with multiple processors

The priority scheme

- **Cluster owners can assign priorities to processes from other clusters**
  - **Local and higher priority processes force out lower priority processes**
- **Pairs of clusters could be shared, symmetrically(C1-C2) or asymmetrically(C3-C4)**
- **A cluster could be shared (C6) among other clusters (C5, C7) or blocked for migration from other clusters (C7)**
- **Dynamic partitions of nodes to private virtual clusters**

Outcome: flexible use of nodes in shared clusters



When priorities are needed

- **Scenario 1: one cluster, some users run many jobs, depriving other users from their fair share**
- **Solution: partition the cluster to several sub-clusters and allow each user to login to only one sub-cluster**
  - **Users in each sub-cluster can still benefit from idle nodes in the other sub-clusters**
  - **Processes of local users (in each sub-cluster) has higher priority over guest processes from other sub-clusters**
- **Scenario 2: some users run long jobs while other user need to run (from time to time) short jobs**
- **Scenario 3: several groups using a shared cluster**
  - **Sysadmin can assign different priorities to each group**

Other services

- **Checkpoint & recovery** - time basis, manually or by the program
- **Built-in on-line monitor** for the local cluster resources
- **On-line web monitor** of the multi-cluster and each cluster
  - http://www.mosix.org/webmon

Disruptive configurations

When a cluster is disconnected:

- **All guest processes move out**
  - **To available remote nodes or to the home cluster**
- **All migrated processes from that cluster move back**
  - **Returning processes are frozen (image stored) on disks**
  - **Frozen processes are reactivated gradually**

Outcome:

- **Long running processes are not killed**
- **No overloading of nodes**

Our multi-cluster campus Cloud (HUGI)

- **11 MOSIX clusters ~250 nodes, ~1000 cores**
  - **In Life-sciences, Med-school, Chemistry and Computer Science**
- **Sample applications that our users are running:**
  - **Nano-technology**
  - **Molecular dynamics**
  - **Protein folding, Genomics (BLAT, SW)**
  - **Weather forecast**ing
  - **Navier-Stokes equations and turbulence (CFD)**
  - **CPU simulator of new hardware design (SimpleScalar)**

Web monitor: www.MOSIX.org/webmon

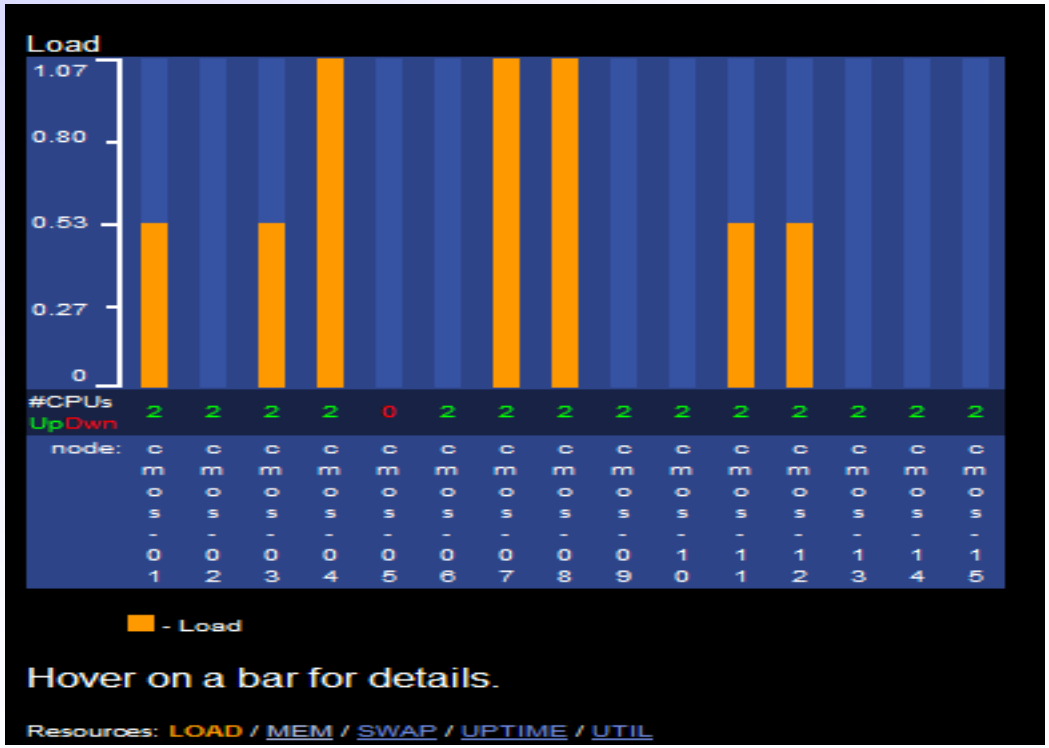
Display:

- **Total number of nodes/CPUs**
- **Number of nodes in each cluster**
- **Average load**

Zooming on each cluster

Display:

- **Load**
- **Free/used memory**
- **Swap space**
- **Uptime**
- **Users**



Conclusions

MOSIX is a comprehensive set of tools for automatic management of Linux clusters and multi-clusters

- **Self-management algorithms for dynamic allocation of system-wide resources**
  - **Cross clusters performance nearly identical to a cluster**
- **Many supporting tools for ease of use**
  - **Includes an installation script and manuals**
- **Can run in native mode or in a virtual machine**

How to obtain a copy of MOSIX

- **A copy is provided at**

http://www.MOSIX.org