CI583: Data Structures and Operating Systems
Distributed systems

Distributed operating systems

The topic of distributed systems is a broad one, and includes everything from the internet, to a small LAN, to render farms in which many (otherwise independent) computers collaborate to complete a computationally intensive task.

Distributed operating systems

From an OS point of view, we are interested in a particular type of distributed system – one in which the elements, or nodes, share one or more of the core OS responsibilities, such as process management.

This can include clusters and cloud computing, but also more tightly coupled distributed systems.

Distributed operating systems

Truly distributed OS

A truly distributed OS is one in which several physically separate machines each provide part of the functionality of a single OS and, taken together, the collection of machines provides the image of a single unit.

The physical location of each OS component (external storage, for example) is transparent to the user. Many nodes may be assigned to a particular function (e.g. the render farm example, though most render farms work at the application level and are not based on a distributed OS).

Distributed operating systems

The motivation for this architecture includes:

1. Load balancing.
2. Reliability, redundancy and fault tolerance. Several nodes can work on the same task, and the first one to complete.
3. Availability.

The danger is that nodes might spend most of their time communicating with other nodes...

Distributed operating systems

Within such a system, each machine runs a microkernel that manages that node's hardware and handles communication with other nodes.

Each node also contains one or more system management components, that provide that node's functionality.

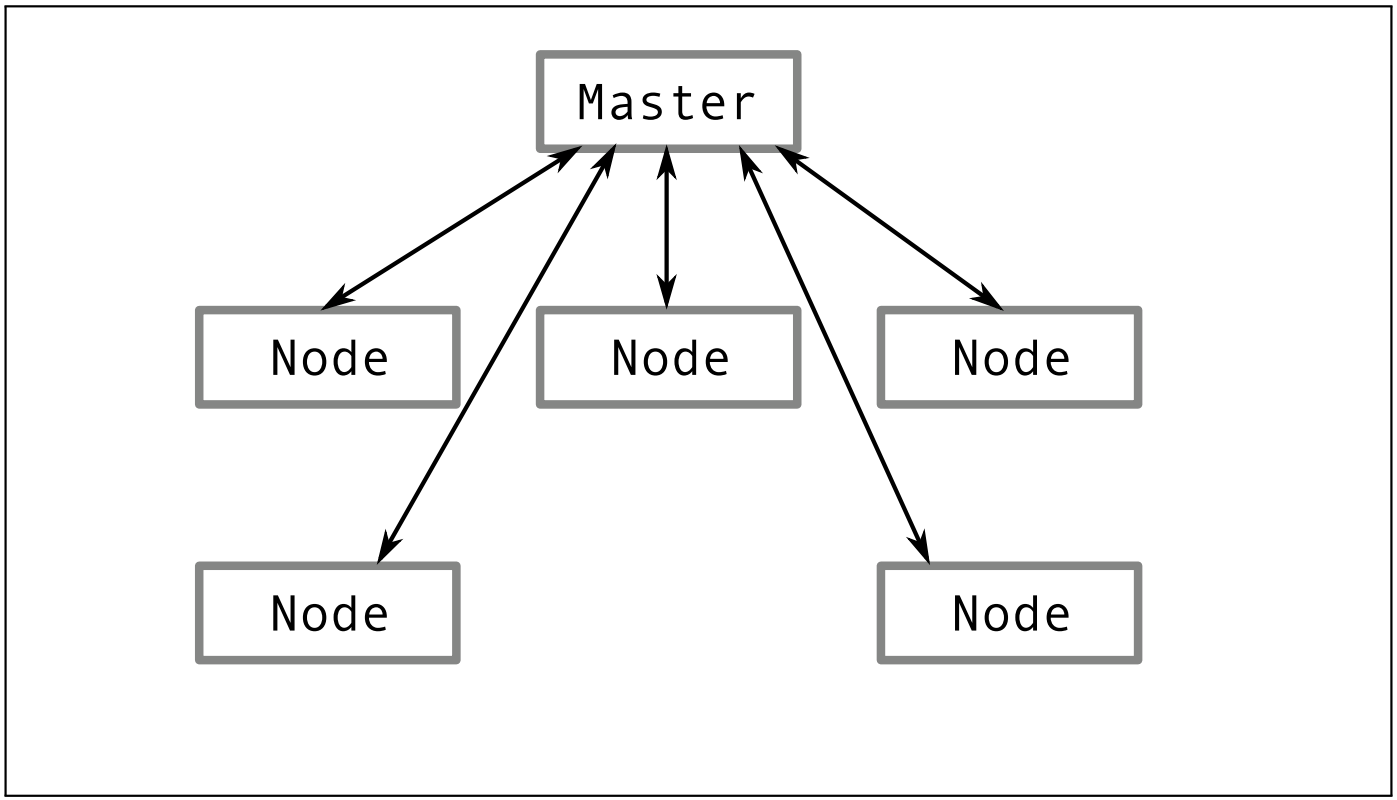
Distributed operating systems

A key distinction between such systems is how the distribution is managed:

1. centralised: all activity is managed by a single master node,
2. decentralised: a tree-like structure where nodes are *branches*, and manage the activity of nodes beneath them, or *leaves*,
3. distributed: each node has one or more links to other nodes and there is no master. There may be no way for any node to "see" the entire system.

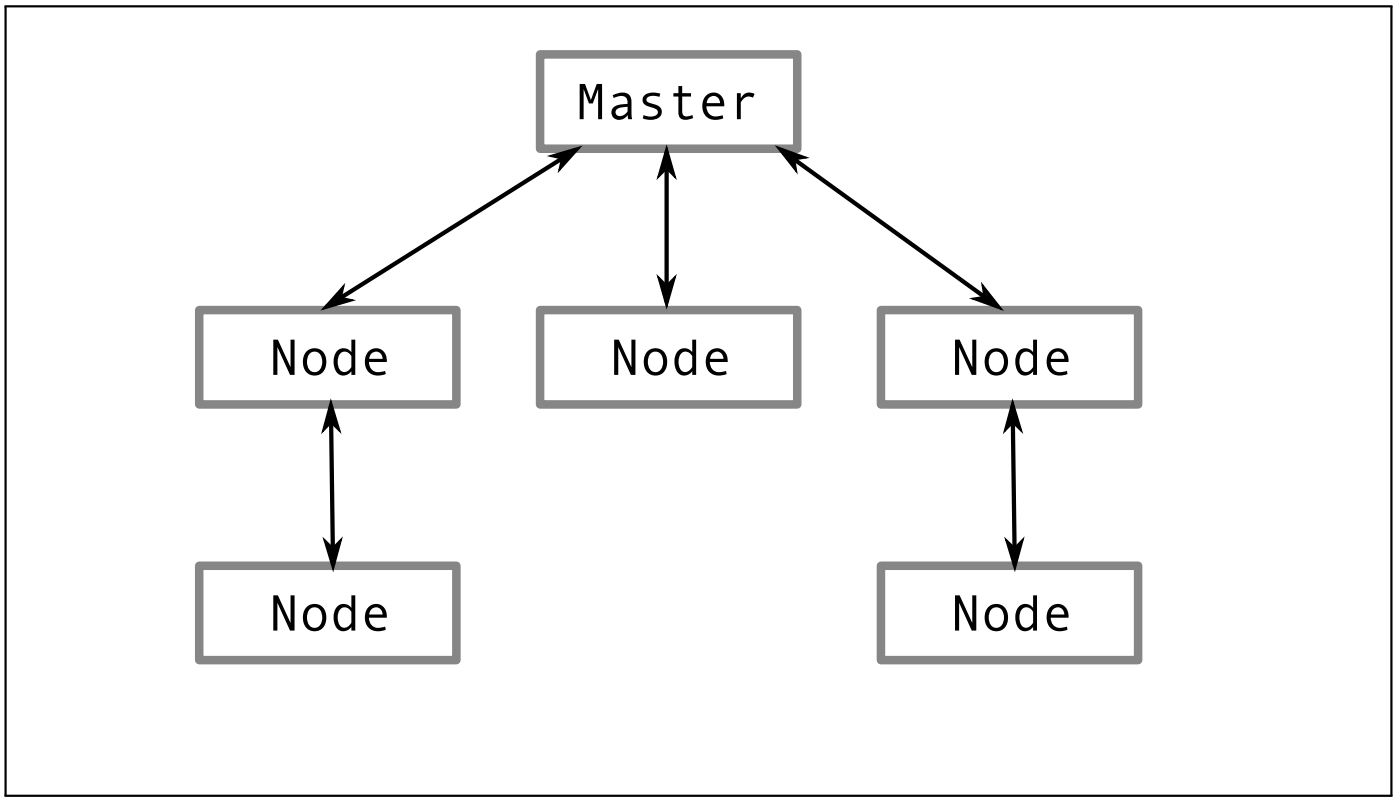
Distributed operating systems

Distributed architecture: centralised



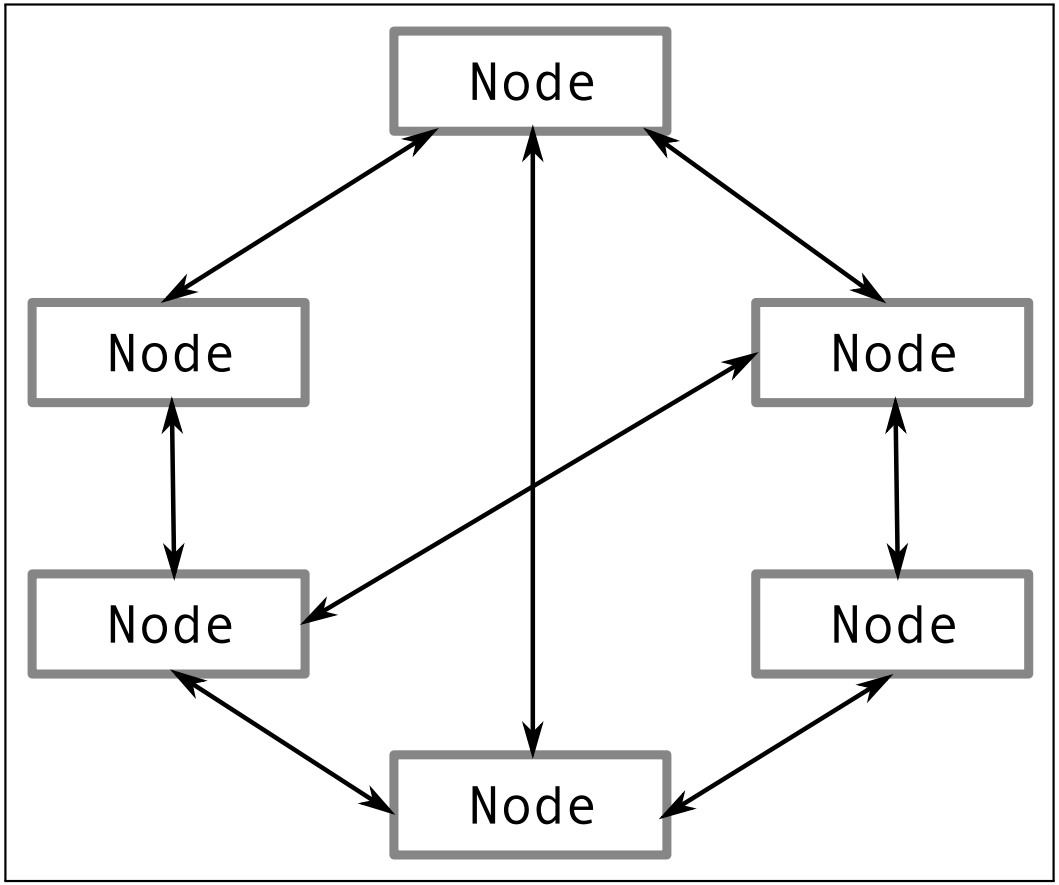
Distributed operating systems

Distributed architecture: decentralised



Distributed operating systems

Distributed architecture: clustered



Distributed operating systems

These tightly-coupled system were the focus of research for decades from the 1970s onward, but no system ever solved the problem of efficient communication between nodes.

The OS usually communicates with IO devices like network cards via a high speed bus, for example. Doing the same thing over a LAN is far slower.

Since the early 1990s, attention has turned to more loosely-coupled systems such as clusters – collections of independent machines each with their own OS.

Clusters

A cluster is a collection of independent machines, each with their own traditional OS installed, but which collaborate on some task in such a tightly-connected way that they can be viewed from the outside world as a single system.

Clusters of quite cheap machines have been assembled that rival the world's fastest and most expensive individual computers.

Using commodity machines means that when a node fails, it is just thrown out and replaced (though it might just be left in place for a while and collected with other dead nodes in one sweep).

Clusters

A cluster provides large-scale parallelism (small-scale being taking advantage of multiple cores), so that jobs can be broken up into tasks that are worked on simultaneously.

Most cluster architectures still rely on a master node whose failure will wipe out the system.

Clusters

A Google server room containing a single cluster called the Compute Engine – world's third fastest supercomputer. You can hire it for $2m per day. The system it runs is based on GFS and the MapReduce architecture.

http://research.google.com/archive/mapreduce.html

Clusters

The Compute Engine has 96,250 physical machines, with access to 770,000 cores. Each core has access to 3.75GB of RAM.

Figures from `http://www.extremetech.com/`.

Clusters

`Hadoop` is a FOSS system based on a reverse-engineering of MapReduce. It is used by Facebook, Yahoo!, Amazon and others.

Three problems to solve:

1. Message passing.
2. Task scheduling.
3. Node failure.

Clusters

Message passing

Obviously, the performance of the bus or network that connects nodes in the cluster is critical. Most clusters use a specialised protocol written on top of `TCP/IP` called `MPI`. This protocol has extended strategies for dealing with messages that never arrive, etc.

Especially in large clusters, it is important to minimise the distance of each node from the master. `Hadoop` is "rack-aware", so it knows the physical location of each node.

Clusters

Task scheduling

Deciding which tasks should be given to which nodes is an open problem.

`MapReduce` has a process on the master node called the `JobTracker`, which breaks work down into individual tasks.

Clusters

Task scheduling

The `JobTracker` passes these tasks to nodes, trying to minimise network traffic by choosing nodes on the same rack and as close as possible to the source of the job. Once processing starts there is no preemption.

`MapReduce` uses a strategy of high redundancy – several nodes might be working on the same item of work and so each item is guaranteed to be done *at least* once, rather *exactly* once.

Clusters

Node failure

When a node fails or appears to be malfunctioning, that task is re-executed (unless a high-redundancy strategy is already in place, meaning that another node is already working on the task).

That node then needs to be isolated from its neighbours – this can be done by powering off the node or simply making sure that it has no access to shared resources, such as external storage that other nodes also make use of.

This is called resource fencing.

Clusters

Like virtualisation, clustered computing is an idea that has really taken off.

Rather than running their own server room, a small-to-medium sized business would now be much better off outsourcing this problem to a company like Google or Amazon S3.

Clusters

The company's computing needs are then served by a cluster sitting in the cloud.

The nodes in the cluster are probably running VMs.

This takes care of load-balancing, reliability, backup, expansion and contraction of the network, load on individual nodes and bandwidth usage.