Advanced Unix System Administration

Lecture 25
May 2, 2007

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Clustering

- Why clusters?
  - Large simulations/heavily-trafficked sites/etc. require a lot of processing power
    - This level of performance would be impractical to build into a single machine
    - Even where it is practical, it might be too expensive
  - Single machines fail
    - Multiple machines in a cluster can provide more reliability

Clustering

- Types of clusters
  - High availability
    - Multiple, (usually) identically configured machines
    - Usually provides a failover mechanism
    - Can also be used in load-balancing configuration
  - Load-balancing
    - Multiple, identically configured machines providing the same service
    - Traffic is directed to each of the cluster machines in a random manner (perhaps weighted for performance of the individual nodes)

Clustering

- Types of clusters con't
  - High performance
    - Designed to provide maximum performance for applications which benefit from parallel processing
    - Applications (usually) need to be designed for the particular solution
    - One can consider modern distributed computing efforts to be an extension of the HPC cluster

Clustering

- Load balancing solutions
  - DNS round robin
    - Post multiple A records; resolvers should choose one at random
    - No front end required; caching/keepalive may be a problem
  - NAT magic
    - Front-end redirects traffic at network layer to machines in the cluster
    - Sessions/state may be an issue
  - Application-specific proxy

Clustering

- High availability solutions
  - Heartbeat solutions
    - A monitoring system watches over the servers in the cluster and detects failures
    - In load-balancing clusters, failure just results in the removal of the server from the cluster
    - Can also have systems where services running on failed nodes are restarted elsewhere
    - Can have systems which use hot spares on node failure

Clustering

- High performance computing
  - MPI: Message Passing Interface
    - Provides an API for processes in an HPC cluster to coordinate between different machines
    - No attempt is made to present a usual API, so applications need to be written for the cluster
  - Single system image (OpenMosix, etc.)
    - Attempts to present illusion of single computer with lots of processors
    - More overhead, but less adaptation needed for applications

Clustering

- Common clustering issues
  - Management
    - As your cluster scales, centralized management solutions are essential, whether they're scripts to run things on lots of machines at once, monitoring daemons, etc.
    - Similar needs as managing large groups of workstations
  - Reliability
    - Lots of nodes = more node failures
    - Need way of dealing with failed nodes gracefully

Clustering

- Clustering issues con't
  - Storage
    - The clusters need some way of accessing the same data, preferably one which scales well with parallel access
    - Can be a network file system or a SAN (storage cluster)
  - Networking
    - Needs to be as fast and reliable as possible
    - Topology is important
      - Want to minimize the number of connections
      - Nodes may need direct connections to other nodes