HY559
Infrastructure Technologies for Large-Scale Service-Oriented Systems

Kostas Magoutis
magoutis@ics.forth.gr
http://www.ics.forth.gr/~magoutis
Background

- Microsoft expanding its web-scale online services
  - Windows Live Search
  - Windows Live Mail

- Goals of Autopilot
  - Keep total cost of data center, including operational and capital expenses, as low as possible
  - Use more intelligent software to replace repetitive work typically handled by operations staff
  - Maintain as few staff as possible on 24-hour call
  - Support many thousands of servers per member of operations staff, run on 8x5 rather than 24x7 schedule
Scope

- Focus on basic services
  - Provisioning and deployment
  - Monitoring
  - Hardware lifecycle, including repair and replacement

- Focus on mechanism rather than policy

- No need to support legacy applications
Design principles

• Fault tolerance
  – Replicate all vital state
  – Any necessary fail-over must be completely automatic
  – Assume non-Byzantine failures
  – Crash (accidental or forced) is only exit mechanism

• Simplicity
Hardware configuration

- Buy and install computers in multiples of a rack
  - Each rack has top-of-rack switch, connected to broader switch hierarchy

- Each computer conforms to one of a small number of hardware specifications
  - “Application”: Multi-core CPU, few direct-attached HDDs
  - “Storage”: Higher number of HDDs per CPU

- Typical “application” rack contains
  - 20 identical multi-core computers, each with 4 HDDs
  - Each computer has a management interface, either built-in to server or accessed via rack-mounted serial concentrator
Autopilot system and applications
Low level services

- **Autopilot services**
  - Filesync: ensure correct files are present on server disk
  - Application manager: ensure correct processes are running
  - OS images pre-configured with DNS names of Autopilot components

- **Independently managed services**
  - DNS (active directory)
  - DHCP
  - Network boot
OS provisioning

• Provisioning service replicated for availability
  – Replicas elect a leader that carries out appropriate actions

• Scans network looking for new machines
  – When found, query Device Manager about OS image to run

• Use server’s management interface to install and boot OS image
  – Machine name determined by position in network hierarchy

• If successful, notify Device Manager
Application deployment

- Deployment service is a set of replicas, each of which contains a set of manifest directories
  - Populated by a build system

- Each server has set of configuration files and application binaries required for its *machine type*
  - Machine type examples: web crawler, front-end web server
  - Each set of configuration files, application binaries described by a manifest

- Filesync service ensures each server always has appropriate manifests
Deploying new code

• When new version of an application is ready to deploy
  – New manifest is stored on deployment service
  – Device mgr instructed to roll out on specified machine type
  – Kicks each server of that type of fetch it
  – When enough servers have fetched it, ready to deploy

• Scale unit
  – Allows staged roll-outs over multiple machine types
  – Example: application component uses 1000 machines partitioned over 10 x 100-machine scale units
  – A given machine type spread evenly across scale units
  – Roll out at most 3 scale units (33% of a type) at a time
Deploying new code (2)

• When ready to roll forward set of computers in a scale unit with particular machine type
  – Update database to make new version manifest active
  – Kick machines to update configuration

• Device manager monitors machines to determine success
  – When enough machines have successfully rolled forward, scale-unit roll-forward considered successful
  – Move to next scale unit

• If timeout reached before sufficient machines have rolled forward, roll back everyone to old version
Automatic repair services

**Healthy**
- continuously error-free for long enough
- an error is reported
- a deployment action is performed
- in Probation too long without moving to Healthy

**Failure**
- a repair action is performed
- Reboot, reimage, replace

**Probation**