Nikolai Joukov, Ph.D.

Enterprise Storage Systems
Content

- Introduction
  - This chart, sign-up

- Enterprise Storage Systems Transformation
  - Real world
  - Storage Optimization Examples

- DevOps

- Reading list
Gain vs. Pain of Transformation

Pain: We want this to be the least!

Gain: We want this to be the most!
Abstract Architecture for Enterprise IT Transformation

- Customer and infrastructure data collection
- Analytics, Design, Planning
- Procurement and physical setup
- Migrate and modify
- Test and remediate
- Synchronize and cutover

Non-customer sources
Consolidated data repositories
Migration Analytics

Project workflow and status visualization
Enterprise Storage
Typical Server Example

- Stored on file systems or block devices directly
  - Server OS
  - software installations
  - software configurations
  - application data
  - temporary files
  - logs
Almost any component can interact directly with another component.

- e.g., a database can store data on a file system or directly on a disk partition.
Local Hard Disks

- Hot-swappable
- Typically used for OS/middleware
- Almost always SCSI
- Some servers have none (diskless)
Embedded or External RAID Arrays

- Redundant Array of Independent Disks (RAID)
- Software
- Hardware
- Internal
- External
- With and without NVRAM

**RAID5**
Local File Systems

- Disk-based
  - fat-like (ext2, fat32)
  - Journaled (ext3, ntfs, xfs)
  - Chunked (ext4)
  - Log-based (zfs, wafl)
- Integrated with lower block-level
  - zfs, vxfs
- Optimized for special purpose
  - Oracle ASM
- Memory-based
- Interfaces
  - /proc, /dev
- Stackable
  - unionfs
  - cryptfs
Network File Systems

- Client-Server
  - NFS
    - v3: stateless
    - v4: stateful
  - CIFS
- Distributed
  - Andrew
  - DCE/DFS
  - GPFS
- Typical uses
  - user directories
  - data sharing
  - shared software installation code
  - common data repositories such as software installations
Discovery: Software Models

- Each per-software sensor builds a specific model (e.g., for DB2 or JFS) based on:
  - configuration data
  - logs
  - available monitoring
- Models get connected together via “URLs”
Logical Volume Manager Example

File systems (local and network)

Logical devices

LVM

Local hard disks

Databases and other software not shown here

Could be SAN connections
Linux Server DB2-to-Storage Discovery Example

- DB2, two instances, databases
- LVM install, volume groups, volumes
- SCSI disk, partitions
- DB2 on another server that we did not scan
- Ext3 mounts
- NFS mounts
- NFSD on another server that we did not scan
- Unused, not partitioned IDE disk
- Another SCSI disk and partition
- NFSD on another server that we did not scan
- SCSI disk, partitions
- DB2 on another server that we did not scan
- Ext3 mounts
- NFS mounts
- Unused, not partitioned IDE disk
- Another SCSI disk and partition
- DB2 on another server that we did not scan
- Ext3 mounts
- NFS mounts
- Unused, not partitioned IDE disk
- Another SCSI disk and partition
- DB2 on another server that we did not scan
Storage Area Networks

- Block-level networks to transfer data between storage devices to servers
- Dedicated networks
- Almost always each server has two redundant links to the storage controllers
- Each server adapter and each storage controller port has a World Wide Port Name (WWPN)
  - Relinking servers and storage requires updating port mappings
- SANs are zoned for security
- Logical Unit Number (LUN) is the unit of data accessed by the servers
OS/hardware clusters

- E.g., AIX, Linux, Windows OS and Veritas Clusters
- They almost always assume shared storage clusters via Storage Area Networks (SANs)
- They can be integrated with middleware and monitor if, say, a database is servicing requests
- Typically have multiple heart-beat loops including a dedicated shared storage volume
Enterprise Storage Systems

- Fully redundant
- Each hard disk has two data ports
- Each such port is connected to a separate server inside of the storage controller
- Disks are RAIDed and have hot spares
VMware ESX Client VM (left) and Server (center)

Local/NAS/SAN

/vmfs/volumes/4bb0e06a-f41c3d56-ee24-000d60d49c13
Storage Optimization Examples
Enterprise Applications

- You will hear “Application” everywhere and during all lectures of this course
- Application is a high-level view on IT based on the performed business function
- Examples: travel expense reimbursement application, student grading application
- Google search is a large distributed application too
- One application can consist of one or hundreds of servers
- Some servers or even objects like databases are shared by multiple applications
- Applications have owners
- Applications have dramatically different criticality for enterprises
Example Use Case: Business Data Criticality vs. Storage Tier (30 production AIX servers)
**Example Use Case: Disk Consolidation (30 production AIX servers)**

- **x100 disk power reduction opportunities by virtualization**

<table>
<thead>
<tr>
<th>Size (GB)</th>
<th>Used (#)</th>
<th>Unused (#)</th>
<th>System (#)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>73</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>36</td>
<td>29</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>73</td>
<td>29</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>178</strong></td>
<td><strong>21</strong></td>
<td><strong>54</strong></td>
</tr>
</tbody>
</table>

- **spinning but unused disks – recommend SAs to power down**
Example Use Case: Database Storage Space Reorganization
(270 AIX, 21 HP-UX, 2 Windows production servers)

- DB2, Oracle, Sybase, PostgreSQL, MySQL, Microsoft SQL DBs
- EMC shared storage
- >200 file systems with tablespaces 100% full – unoperational databases

<table>
<thead>
<tr>
<th>Databases (#)</th>
<th>1,076</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size (TB)</td>
<td>151.7</td>
</tr>
<tr>
<td>Size Old (TB)</td>
<td>0.4</td>
</tr>
<tr>
<td>Unused (TB)</td>
<td>50.3</td>
</tr>
</tbody>
</table>

Tablespaces not used for 2 months or more
Tablespace space allocated but not used
Example Use Case: Network File Systems Usage (30 production AIX servers)

<table>
<thead>
<tr>
<th>Usage Type</th>
<th>Clients</th>
<th>Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homes</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Application Data</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Bulk Data</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

only a few servers depend on NFS performance
Data Migration for Server Virtualization

- Databases and all other data needs to be transferred to the new images even if all software is reinstalled and reconfigured
- As always: discover, plan, move, test
- Block-level tools
  - move block devices
- File system-level tools
  - move file systems
  - the only way to move data between different file system types
    e.g., when migrating from Solaris to AIX
- Some tools can capture the data on-the-fly without bringing the servers down or even stopping the services
- Bandwidth is almost always an issue
  - move data in bulk (sometimes physically)
  - resynchronize latest changes
DevOps
IT REVOLUTION is happening now

- **Yesterday:**
  - One security expert / sysadmin / application owner manually manages 10s of servers

- **Tomorrow: 1,000s!**
  - Do not manage a server: learn to automate the process of managing servers

- **Example:**
  - Nathanael Burton, NSA CTO presented at OpenStack summit:
    Team of only 12-15 people manages all NSA OpenStack Cloud infrastructure
DevOps

- A portmanteau of "development" and "operations"
- Term coined in 2008, got popular in 2009
Infrastructure-as-code

- Basic principles:
  - Do no system and software configurations manually on each system
  - Templates describing how to install/configure systems/devices/software/users
e.g., deploy your application as rpm on a set of hosts, describe how to configure it

- How does it affect migrations?
  - Automated discovery of as-is state
  - Automated provisioning
  - Automated verification that configurations do not change (diverge) from standard later
Infrastructure-as-code today

- Highlights of the main players:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Puppet</th>
<th>Chief</th>
<th>Salt</th>
<th>Ansible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Ruby</td>
<td>Ruby</td>
<td>Python</td>
<td>Python</td>
</tr>
<tr>
<td>First release</td>
<td>2005</td>
<td>2009</td>
<td>2011</td>
<td>2012</td>
</tr>
<tr>
<td>Out-of-order execution</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Push/Pull</td>
<td>Pull (Agents)</td>
<td>Pull (Agents)</td>
<td>Both</td>
<td>Push (Agent-less)</td>
</tr>
<tr>
<td>Notes</td>
<td>More OSs supported, Largest user base</td>
<td>Easier to learn</td>
<td>Infrastructure-as-data</td>
<td>Very simple</td>
</tr>
</tbody>
</table>
Simple Template Examples

- Puppet

```ruby
class httpd {
  package { 'httpd-devel':
    ensure => installed,
  }
  service { 'httpd':
    ensure => running,
    enable => true,
    subscribe => Package['httpd-devel'],
  }
}
```

- Salt

```yaml
#/srv/salt/mysql.sls
mysql:
  pkg.installed:
    - name: mysql-server
  service.running:
    - enable: True
    - require:
      - pkg: mysql-server
```

- Ansible

```
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum:
        pkg=httpd state=latest
    - name: write the apache config file
      template:
        src=/srv/httpd.j2 dest=/etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at boot)
      service:
        name=httpd state=started enabled=yes
      handlers:
        - name: restart apache
          service:
            name=httpd state=restarted
```
Docker

- Docker is relevant for this lecture because it is:
  - 1) Union file system based
  - 2) DevOps-based
Docker: layers

- Dockerfile

```
FROM debian:wheezy

WORKDIR /tmp

RUN wget -nv &&
tar -xvf someutility-v1.0.0.tar.gz &&
mv /tmp/someutility-v1.0.0/someutil /usr/bin/someutil &&
rm -rf /tmp/someutility-v1.0.0 &&
rm /tmp/someutility-v1.0.0.tar.gz
```

- DevOps-configured layers
Reading List (storage)

- Nikolai Joukov, B. Pfitzmann, H. V. Ramasamy, and M. V. Devarakonda, Application-Storage Discovery, The 3rd Annual Haifa Experimental Systems Conference (Systor'10), Haifa, Israel, May 2010


- Mohammad Hajjat, Xin Sun, Yu-Wei Sung, Dave Maltz, Sanjay Rao, Kunwadee Sripanidkulchai and Mohit Tawarmalani, "Cloudward Bound: Planning for Benefical Migration of Enterprise Applications to the Cloud", Proceedings of ACM SIGCOMM, New Delhi, India, August 2010

Questions?