Software Testing and Deployment Using Virtualization and Cloud

Presented by: Omer Khalid

Contributors: Arsalaan Shaikh, Brice Copy
Outline

- Part I - Background
  - Use cases, Infrastructure

- Part II - Concepts
  - Virtualization, Cloud Computing, Tools and Deployment Models

- Part III - Implementation
  - Private cloud, Contextualization, Image Management

- Part IV - Results
  - Deployment Times, Wait Times
Part I

Introduction
Use Cases

• OPC Server
  • Different vendors: CAEN, Wiener, ISEG
  • Load testing to examine OPC server behaviour – high traffic, duration of execution etc.

• PLC Security
  • Process monitoring, Communication analysis
  • Security attack deployments and evaluations

• PVSS Testing and Patching
  • Developing new patches, bug fixing, pre-production deployment

• Infrastructure Development
  • Multi-Platform Deployment, Hot-Swap of production services
Common Requirement

Setup and Configure

New Machines

with pre-configured software

within 30 minutes
Physical Architecture

16 HP Proliant G4 Servers: 4 TB Storage, 10 GB RAM each
Additional Constraints

- **Multiple Platforms**
  - Windows XP, Scientific Linux 5/6 (32 and 64 bit)

- **Software Dependencies**
  - OPC Server (multiple versions, multiple vendors)
  - PVSS (multiple versions/patch level)
  - PLC Environments (Step7 and Unity – multiple versions)

- **Time Constraints**
  - Each user needs a machine urgently for testing
  - Each machine must be re-installed after every test \( \approx 2 \text{ hrs} \)
Additional Constraints

Multiple-Dimension Knapsack Problem

Limited Supply – More Demand
(resources vs users vs time)

How to solve?
Virtualization And Cloud Computing
Virtualization, in computing, is the creation of a virtual version of something such as hardware platform, operating system, storage device or network resources.
Cloud Computing?

Delivery of computing (CPU, Storage, Memory) as utility-service over a network

Cloud Computing?

Delivery of computing (CPU, Storage, Memory) as utility-service over a network

Part III

Implementation
What did we do?

- **Infrastructure as a Service (IaaS):**
  - Virtualization and Cloud Platform
    - Private/Shared: Users tests running in virtual machines + Library of images
  - Mash-up using:
    - Commercial, Open Source, and custom developed tools
    - Evaluated tools:
      - VMWare vCenter/Lab Manager, Citrix XenServer/Lab Manager
      - OpenStack (Compute/Glance), Eucalyptus (with Xen Source), OpenNebula (open-source/commercial)
      - Deployed: VMWare ESXi + OpenNebula + OpenStack (Glance)
  - Performance Testing:
    - Different deployment models – incremental requests vs. burst requests (10 VM’s)
    - Different storage models – shared storage (NFS) vs. distributed storage (each server)

Objective: Minimize VM Deployment Time
What did we do?

10 Server, 3 TB of Storage, 100 GB Ram = 90 Virtual Machines
Image Service and Configuration

• Image Management
  • Python service – runs on every server, standalone component
  • Updates and download images every hour from the OpenStack image service
  • Modified version of Open Nebula – deploys VM’s using locally cached image
Image Service and Configuration

• Contextualization (Windows specific)
  • Registering MAC address in the CERN network database
  • Regenerating system security ID using System Preparation (sysprep) Tool
    • Uses a special configuration for each organization wide settings
  • Adding the machine in the CERN Domain -> Reboot
Part IV

Results
Aggregated Results

Aggregated deployment times for all configurations

- **arch-1**: shared storage without front end
- **arch-2**: shared storage with front end
- **arch-3**: distributed storage, remote copy
- **arch-4**: distributed storage with local cache

Network delay to copy each VM image

Central storage based deployment

Optimization due to local caching of images

![Bar chart showing deployment times for different architectures and intervals.]
Wait Times – Burst mode

Single-Burst: 10 VM’s on one server

- arch-1: shared storage without front end
- arch-3: distributed storage, remote copy
- arch-2: shared storage with front end
- arch-4: distributed storage with local cache

remote copy, all VM’s booted at the same time

Shared central storage – difference due to memory caching

Optimization: all VM’s within 10 mins
Wait Times – Burst mode

Multi-Burst: 10 VM’s on all servers

- arch-1: shared storage without front end
- arch-3: distributed storage, remote copy
- arch-2: shared storage with front end
- arch-4: distributed storage with local cache

Multiple servers distribute the load – convergence

Optimization: all VM’s within 10 mins
Wait Times – Interval mode

Single-Interval: 10 VM’s on one server every 3 mins

- arch-1: shared storage without front end
- arch-3: distributed storage, remote copy
- arch-2: shared storage with front end
- arch-4: distributed storage with local cache

Remote copy: one-by-one, less parallelism

Multiple servers distributes the load – convergence

Optimization: all VM’s in 30 mins
Wait Times – Interval mode

Multi-Interval: 10 VM’s on all servers every 3 mins

- arch-1: shared storage without front end
- arch-3: distributed storage, remote copy
- arch-2: shared storage with front end
- arch-4: distributed storage with local cache

- Multiple servers distributes the load – convergence
- Optimization: all VM’s in 30 mins
Conclusion

• Cloud/Virtualization
  • Useful technologies for better utilization of physical infrastructure
    • Not necessarily applicable to all scenarios, but true for some
    • Lots of tools available: Commercial and Open Source
  • Private “Local” Clouds
    • Possible to deploy with commodity hardware – reasonable cost
    • Requires some expertise to deploy/maintain

• Achievement
  • Added flexibility to the physical infrastructure: multi-tenant
  • Library of images for our users – different OS with software
  • Users can get their VM (Windows or Linux) within 30 mins
    • Run the test, shutdown the VM’s after testing.
Useful Pointers

- Benjamin Farnham: WEPMS006, MOPMS025
  - “Automated testing of OPC Servers”
  - “Migration from OPC-DA to OPC-UA”

- Brice Copy: WEPKS001, WEAAULT02
  - “Agile Development and Dependency Management for Industrial Control Systems”
  - “Model Oriented Application Generation for Industrial Control Systems”

- Filippo Tilaro: WEPMU029
  - “Industrial Devices Robustness Assessment and Testing against Cyber Security Attacks”

- Paul C. Burkimsher: THBHMUST01
  - “Multi-platform SCADA GUI Regression Testing at CERN”
Questions

WHERE THE HECK
IS MY DATA?

ITS THERE, UP
IN THE CLOUDS.

Contact: Omer.Khalid@cern.ch