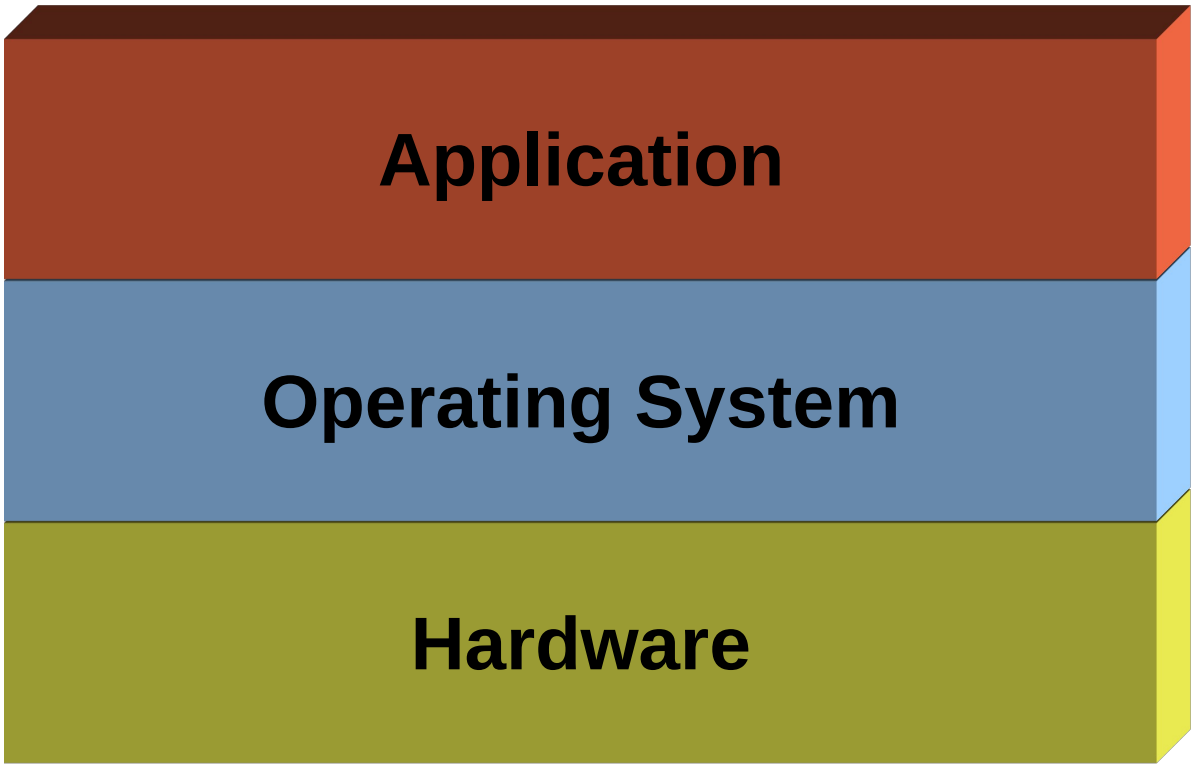
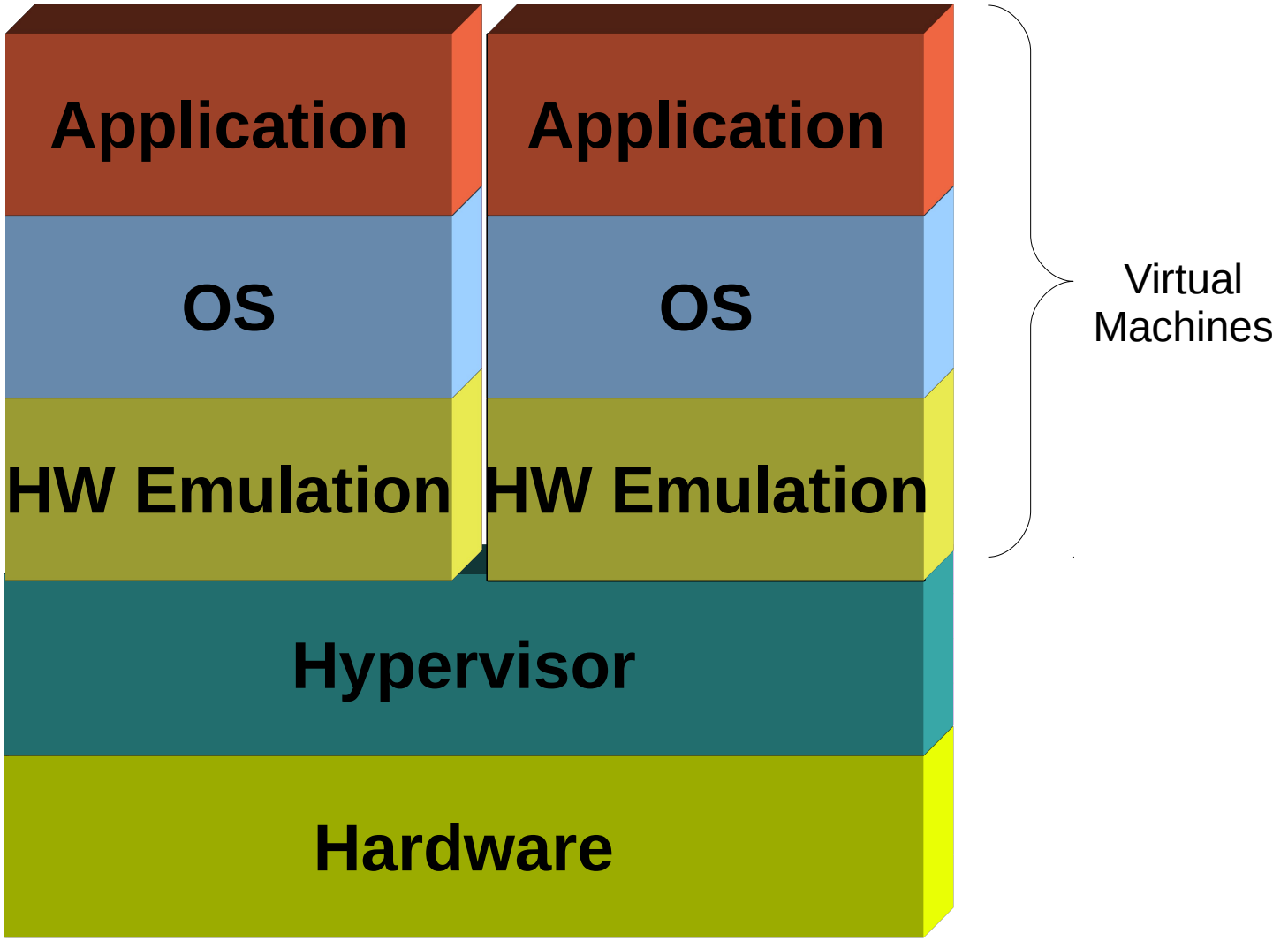
Options in Open Source Virtualization and Cloud Computing

Andrew Hadinyoto
Republic Polytechnic

No Virtualization



Virtualization (general)



Benefits?

- Maximizing computer hardware utilization
- Reducing compute hardware procurement
- Learning, Teaching, Experimenting
- Development and staging
- Much more

Options in Linux

- VMWare
- VirtualBox
- Xen
- KVM
- QEMU
- (others)

But first, the CPU (Intel, AMD)

- Does it have VT-x or AMD-V?
  - CPU direct support for virtualization
- No VT-x or AMD-V
  - Para virtualization (modified OS)
  - Full virtualization ala VMWare or VirtualBox
- With VT-x or AMD-V
  - Approximately - CPU sold from mid 2006 onward should have it – but, check the spec and BIOS
  - Hardware assisted virtualization (HVM)
  - Full virtualization with HVM

VMWare

- Not Open Source
- Good performance
- Solid and stable
- Great for Desktop and Server
  - VMWare Server, ESX, ESXi
- Supports HVM
- Problem with Linux and VMWare Server
  - Patching is needed
  - Time skew – disable ACPI and APM on the host

VirtualBox

- Two type of Licenses
  - Open Source Edition – GPL
  - Proprietary
    - PUEL - for evaluation and personal use
- As of version 3.0 onward – GREAT!
- Supports HVM
- Runs on Linux, Windows (includes 7), Mac OS X (Snow Leopard)
- Command line support
- Recommended for desktop (we'll talk more about this)

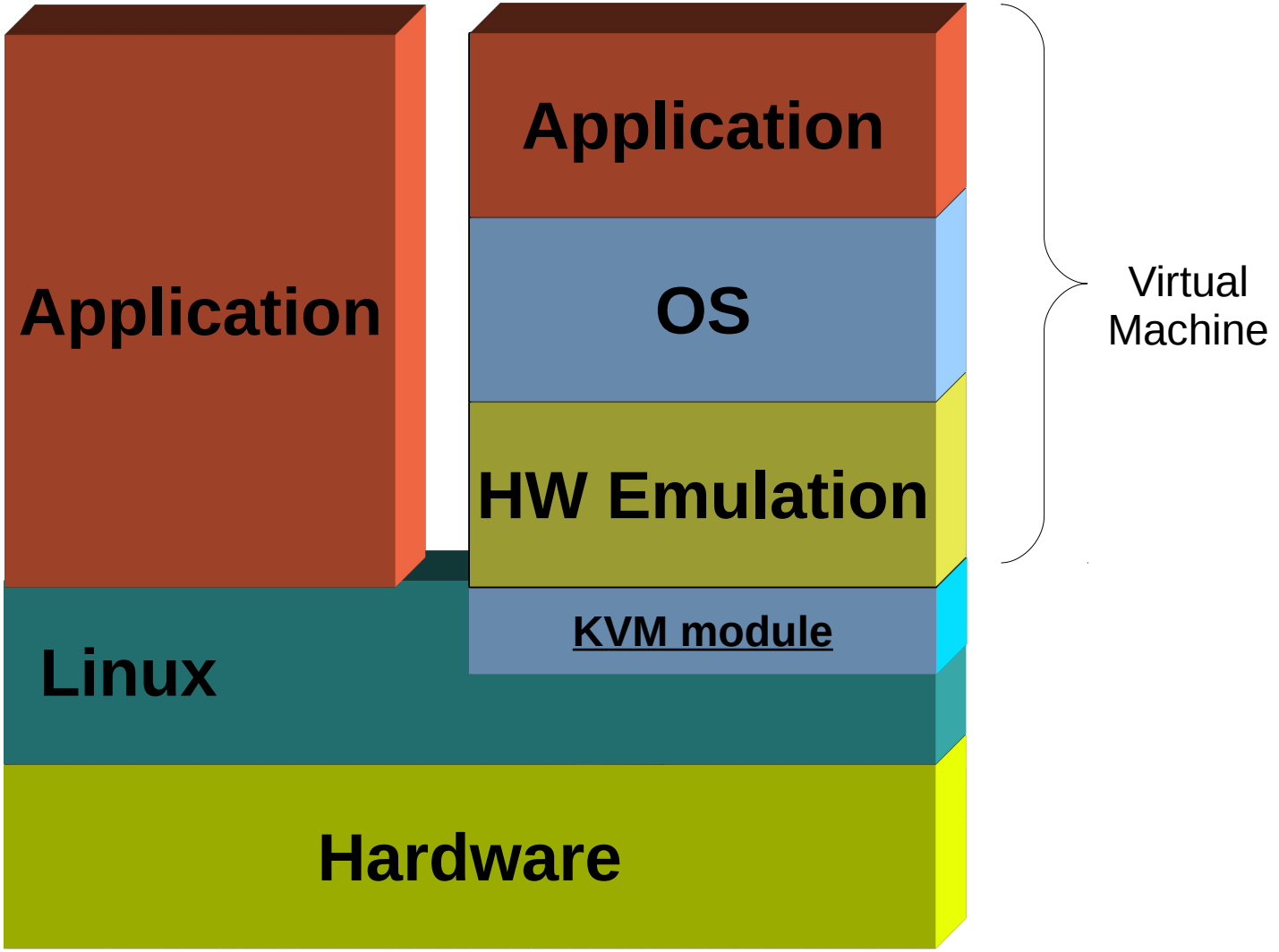
Xen

- Open Source – GPL
- Excellent performance – near native speed
  - Paravirtualization – best out of the box
  - Hardware assisted virtualization (HVM)
- Excellent for server
- Issues
  - Intrusive – fork of the kernel
  - RHEL/CentOS 5.x – uses 2.6.18 kernel
  - HVM is extremely slow – requires configuration tweaking

KVM

- License – GPL, LGPL
- Acquired by Red Hat
- The future? Very likely
  - Red Hat current and future strategy
  - Ubuntu Cloud/Eucalyptus is using KVM too
- Only supports HVM – must have VT-x / AMD-V
- KVM is a Linux module – export /dev/kvm
- Relies on QEMU for hardware emulation
  - I/O (block and eth)
  - Guest execute directly
  - Virtio – paravirtualization *driver* – better performance

More on KVM



http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf

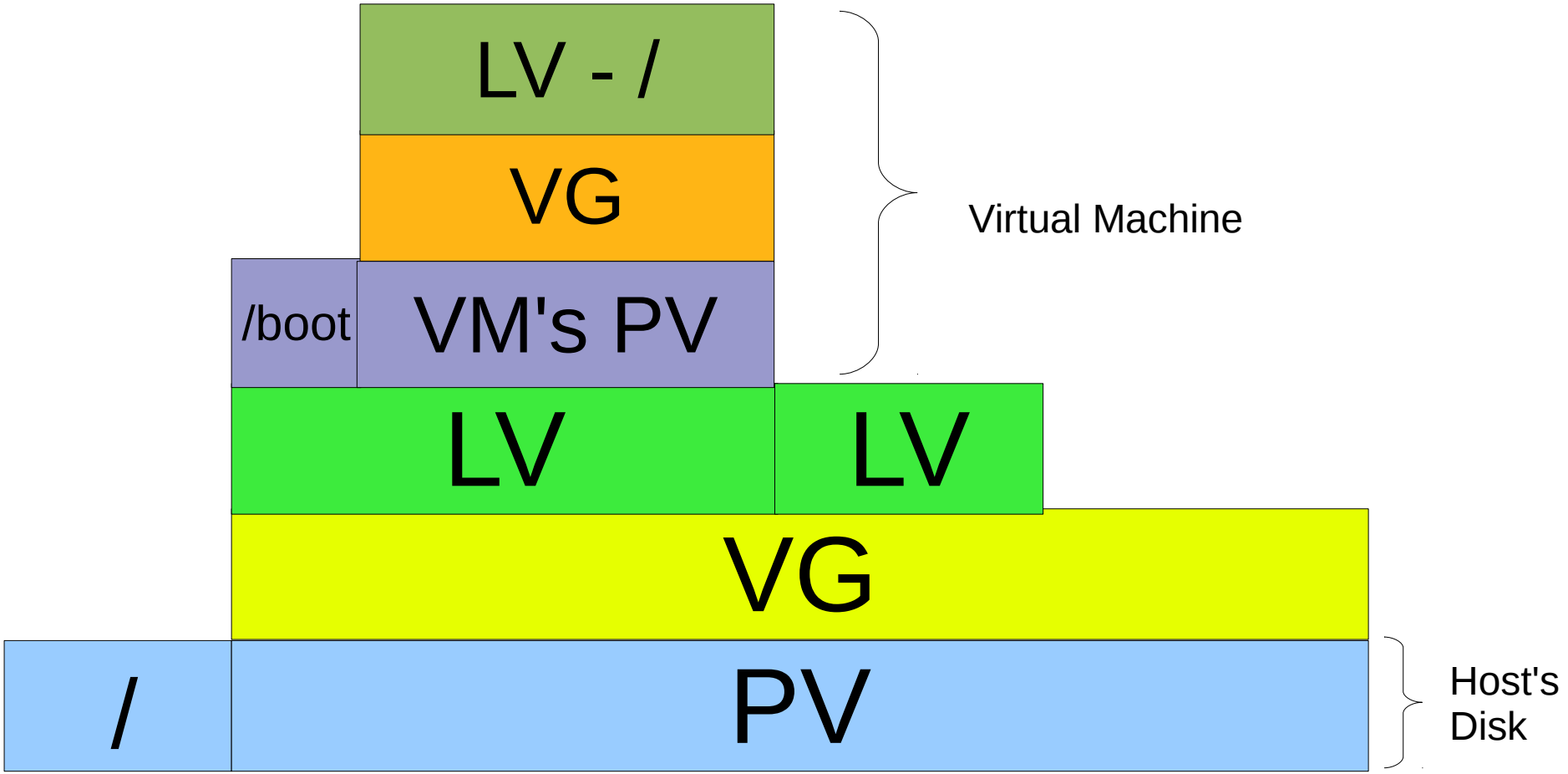
Working with KVM

- libvirtd
  - virsh, virt-install, virt-clone, virt-manager, virt-viewer
- Install new VM with
  - virt-manager (GUI)
  - virt-install (CLI)
- virtio – enabled manually
  - http://wiki.libvirt.org/page/Virtio
  - /etc/libvirt/qemu/yourvm.xml
- Managing
  - virt-manager (GUI)
  - virsh

Storage and KVM

- Image (file) - qcow2 format
- Regular disk partition
  - Best performance
- Logical Volume Manager (LVM)
  - Most flexible – resizable, expandable
  - Better than image-based but some overhead
  - A little complicated

Storage with LVM



Networking and KVM

- libvirtd provides *virbr0* interface by default
  - Provides DHCP interface to the VMs
  - NAT
- Bridge
  - Best if VMs are used servers
  - Appears as if it is a separate network interface
  - Manual setup
    - CentOS/RHEL - /etc/sysconfig/network-scripts/ifcfg-br0

Cloning

- Install OS on one VM
- Clone it with *virt-clone*
  - *virt-clone --force --original vmtoclone --name newvm --file /dev/VG/newvm*

You need small VM size?

- Ubuntu JeOS - Just Enough OS
- Build using *vmbuilder*
  - Produce *qcow2* image for KVM
  - Produce *vmdk* image for VMWare
- VM with LAMP ~ 420MB
- https://help.ubuntu.com/community/JeOSVMBuilder

Random Notes VirtualBox

- Multiple VMs to form an internal network
  - *intnet* interfaces
  - Learning/Teaching networking, System Administration
- Ubuntu JeOS and vmbuilder
  - VirtualBox understands VMWare vmdk image format
  - Produce JeOS vmdk image, run it on VirtualBox

Cloud Computing

Cloud Computing Types

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Utility Computing

Software as a Service

- Software solution that runs on the provider's infrastructure
- Usually web-based application
- Example:
  - Google Docs, Google Map
  - Salesforce (sales management application)
- May be integrated with other apps (mashups)

Platform as a Service

- Provides a platform for developing application
- Google App Engine (GAE)
  - Allows developer to write web-app code, and
  - Deploy it over at GAE's servers
  - The app will then be accessible from GAE's server
  - Python-based framework (Java is coming)
- Heroku
  - Ruby on Rails apps
  - Using “git” to push from local development to the cloud

Google App Engine (GAE)

- Running web application on Google App Engine platform
- Optimized for Web application
- Using GAE web framework
  - Currently, Python-based framework
  - Other programming languages coming soon
- Data storage provided by GAE
  - Database, but not RDBMS like MySQL

Utility Computing

- Computing resources as public utility
  - Public utility such as electricity, water, gas
- Mostly using virtualization to maximize hardware utilization
- Computing resources are for general purpose use
- Example:
  - Amazon Elastic Compute Cloud (EC2)

Elastic Compute Cloud (EC2)

- “Renting” of computing Resources
  - Small, Large and Extra Large computing power
- Xen Virtual Machines
- Currently three zones: US-East, US-West, EU (Asia likely will be in Singapore)
- As of now, less than US$0.10 per virtual machine for small VM (US-East)
  - Windows OS costs more
- Additional charges apply for the usage of
  - Bandwidth usage
  - Monitoring
  - Permanent IP address (Elastic IP)
- Details: http://aws.amazon.com/ec2/#pricing

Simple Storage Service (S3)

- Permanent storage in the cloud
- You pay by the GB
  - Currently US$0.15 per GB per month for less than 50TB
  - Can handle Petabytes (PB = 1,000TB) of data
- Separate charges for storing and retrieving of data
- Data transfer will apply soon
- Details: http://aws.amazon.com/ec2/#pricing

Elastic Block Storage (EBS)

- EC2 can have permanent storage via EBS
- EBS is using S3
- EBS must be in the same zone as the EC2 instances
- Mountable from within the instance
- Currently, US$0.10 per GB per month
- Currently, US$0.10 per million I/O (Input Output)
- Details: http://aws.amazon.com/ebs/

Hypothetical Calculation

- 25 instances, small, US-East
- 8 hours * 4 days = 32 hours
- 25 GB of storage
- 10 GB I/O between the machine and storage
- 10 GB Network I/O between EC2 instances to my office
- Total: US$69.85
- Calculator: http://calculator.s3.amazonaws.com/calc5.html

EC2 - Good and Bad

- Advantages
  - On demand thousands of computers in a matter of minutes (but, default is 20)
  - On demand large storage (GB, TB, PB) in a matter of minutes
  - No setting up of your own infrastructure
  - Can run all sorts of applications (EC2 provides the machine, NOT the application)
- Disadvantages
  - Not permanent – once you shutdown your instance, all data will be cleared
  - Latency issue – I/O to storage takes long because it's not in the same machine
  - Not suitable for some applications

Applications

- It's GOOD for
  - Batch processing
    - Converting large number of videos or pictures
    - Digital rendering
  - Data warehousing
    - Processing large data for business analysis
  - Running MapReduce jobs
    - Large scale data processing
- It's BAD for (when EBS is used)
  - Highly interactive app
  - Web application
    - Database data must be stored in EBS
    - Yet, EBS has poor latency

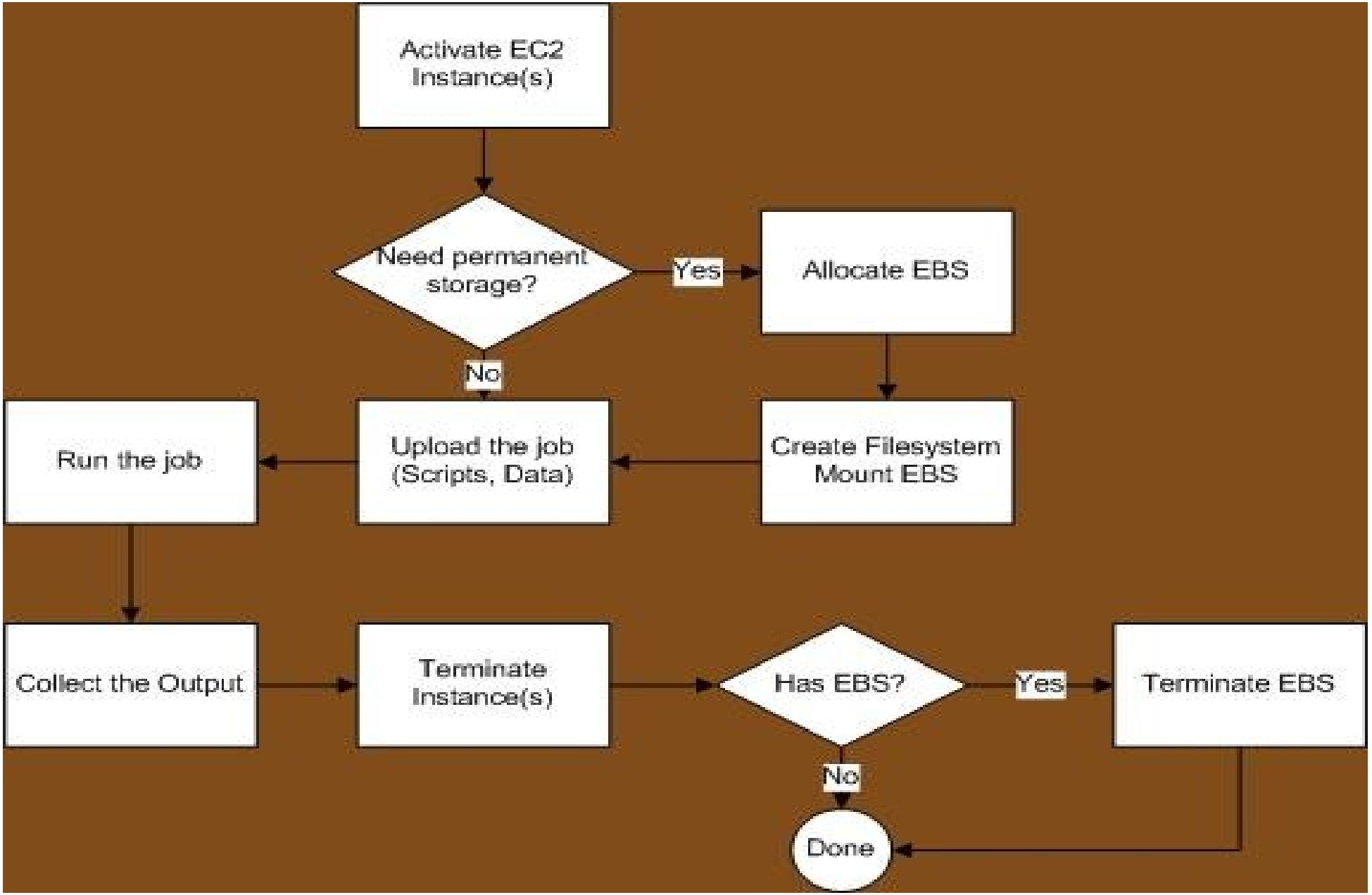
Working with EC2

- Web Interface – AWS Management Console
  - http://aws.amazon.com/console/
- EC2 API tools
  - Command line interface
  - Using Java – JDK 1.5 and above
  - Interacts via Web Services
    - Using SOAP web service protocol
    - In fact, AWS stands for Amazon Web Services
  - Runs on Linux, Windows, Mac OSX, Solaris
  - Scriptable – good for automation

Main Flow of Setting Up EC2 Tools

- JDK is Java Development Kit
- Some environment variables must be set to locate
  - Java Runtime
  - Private Key
  - Certificate

Main Flow Working With EC2



Security

- Login credentials
  - Email and password login
  - For AWS Management Console login
- Private Key and Certificate
  - To make call with EC2 API
  - *cert-UVUIMO.....pem* (X.509 Certificate)
  - *pk-UVUIMO....pem* (Private Key)
- Key Pair
  - SSH key to access your instance
- Security Group
  - Firewall rule for accessing your instance

Ref: https://aws-portal.amazon.com/gp/aws/developer/account/index.html

Working with EC2

- Connection is mostly via SSH and can be slow at times
- Issuing remote SSH command maybe a better option
  - *ssh-agent* and then *ssh-add* can be used for passwordless access
  - For our case today, it helps to set our default identity (no "*-i id_rsa-c330YourName*")

Q and A?

Offline: andrew_hadinyot@rp.sg

EC2 Demo

Activities – KVM

- Enable - VT-x / AMD-V in BIOS
- Install KVM
  - Ubuntu/Debian - apt-get install kvm
  - CentOS - yum install kvm
- Install virt-manager
  - Ubuntu/Debian - apt-get install virt-manager
- Get the Ubuntu Server
  - http://foss.sit.rp.sg/ubuntu/

Activities - KVM - JeOS Ubuntu/Debian only

- apt-get install ubuntu-vm-builder
- Install virt-manager if you haven't
- Get the build script
  - http://foss.sit.rp.sg/ubuntu/build-kvm.sh
  - Run as root
- Run the VM from virt-manager once finished
- Setup the bridge
  - http://foss.sit.rp.sg/ubuntu/interfaces (copy to /etc/network/interfaces
  - /etc/init.d/networking/restart

- Setup bridge
  - Edit /etc/libvirt/qemu/ubuntu.xml
  - Modify from:

    [...]

    <interface type='network'>

    <source network='default'/>

    </interface>

    [...]
  - To:

    [...]

    <interface type='bridge'>

    <source bridge='br0'/>

    </interface>

    [...]