Running Linux and Open Source in the Cloud: Tips and Tricks

Rod Cope, CTO & Founder
OpenLogic, Inc.

Agenda

- Introduction
- Background
- Problem
- Solution
- Tips & Tricks
- Final Thoughts
- Q & A

Introduction

- **Rod Cope**
  - CTO & Founder of OpenLogic
  - 25 years of software development experience
  - IBM Global Services, Anthem, General Electric
  - First Linux: Slackware 1.01 in 1993 (on 24 3.5” floppies!)
  - Currently writing “Cloud Computing in Action: Innovating with Open Source” for Manning
- **OpenLogic**
  - Open Source Provisioning, Support, and Governance Solutions
  - Certified library w/SLA support on 650+ Open Source packages
    - Including CentOS, Ubuntu
    - http://olex.openlogic.com
  - Over 250 Enterprise customers

Background: Linux & Open Source in the Cloud

- **In 2010, Amazon EC2 spun up an average of 70,000 instances per day**
- **Gartner expects "cloud services" to be a $150b market in 2 years**
- **All public clouds support Linux VM's**
  - Including Microsoft Azure (CentOS)
  - Amazon EC2: Ubuntu is most popular OS (by factor of 3)
- **Open Source is a great fit for the cloud**
  - Universal access, innovative, easier, better
  - No commercial licensing issues related to counting servers, auto scaling, part-time instances, etc.

OpenLogic's Problems to Solve

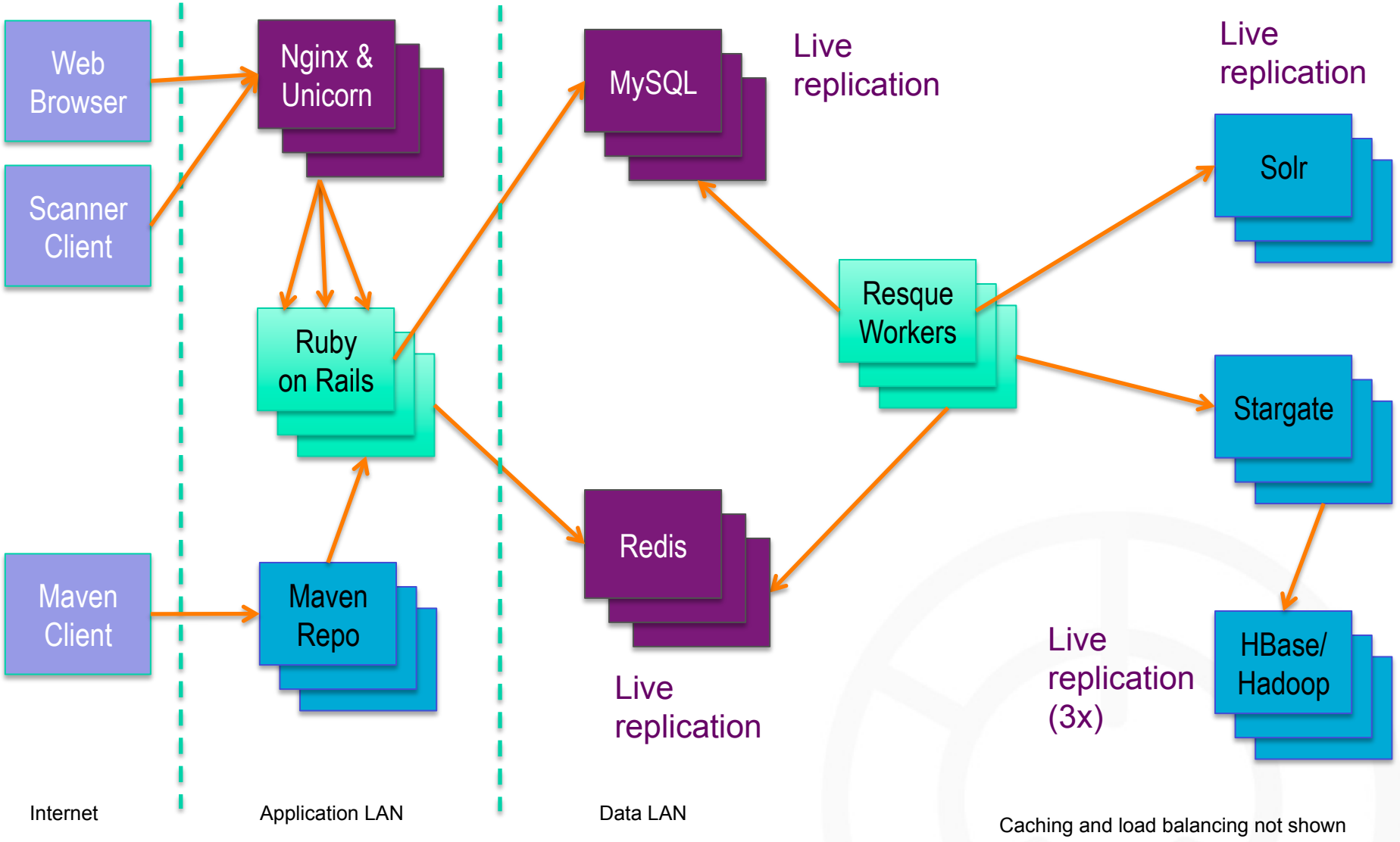
- **Source & binary scanner**
  - "Big Data"
    - All the world's Open Source Software
    - Metadata, code, indexes
  - Growing every day
  - Need real-time random access to all data
  - Long-running and complex analysis jobs

- **Cloud product**
  - Deploy, manage, monitor, and support open source stacks on Amazon EC2, Rackspace, and other public and private clouds
  - Multiple operating systems (starting with Ubuntu)

Scanner Solution

- **Hadoop, HBase, and Solr**
  - Hadoop – distributed file system
  - HBase – "NoSQL" data store – column-oriented
  - Solr – search server based on Lucene
  - All are scalable, flexible, fast, well-supported, used in production environments
- **And a supporting cast of thousands…**
  - Ubuntu, CentOS, Stargate, MySQL, Rails, Redis, Resque, Nginx, Unicorn, HAProxy, Memcached, Ruby, JRuby, OpenSSL, …

Solution Architecture



Hadoop & HBase Implementation

- **Environment**
  - 100+ CPU cores
  - 100+ Terabytes of disk
  - Machines don't have identity
  - Add capacity by plugging in new machines
- **Why not EC2?**
  - Great for computational bursts
  - Expensive for long-term storage of Big Data
  - Not yet consistent enough for mission-critical usage of HBase

Configuration & Tuning

Base Operating System Configuration

- **CentOS 5.4 and 5.5**
- **Typical security measures**
  - Patches
  - Turn off unused services
  - Limit physical access
  - Limit remote access to authorized users on certain networks
  - Audit logging
  - Intrusion detection
  - And so on…

BIOSes and Drivers

- **Upgrade your machine BIOS, network card BIOS, and all hardware drivers**
  - Example: issues with certain default configurations of Dell boxes on CentOS/RHEL 5.x and Broadcom NIC's
    - Will drop packets & cause other problems under high load
  - Disable power saver (C-states) in machine BIOS
    - BIOS setup -> Processor -> C-states (disable)
    - Uses significantly more power during idle time

Driver issues

- **Buggy Broadcom driver (bnx2) before 2.6.33**
  - “under certain heavy traffic conditions in MSI-X mode, the driver can lose an MSI-X vector causing all packets in the associated rx/tx ring pair to be dropped. The problem is caused by the chip dropping the write to unmask the MSI-X vector by the kernel (when migrating the IRQ for example).”
  - If on CentOS/RHEL 5.4, can upgrade kernel and driver, then turn off MSI

Disable MSI (Message Signaled Interrupts)

- **/etc/modprobe.conf**
  - alias eth0 bnx2
    alias eth1 bnx2
    alias eth2 bnx2
    alias eth3 bnx2
    options bnx2 disable_msi=1

App-specific OS Configuration: Hadoop

- **Hadoop abuses your network – hard**
- **Any little problem will be exposed**
- **Based on Java, which brings additional networking issues**
- **Picky about name resolution, refuses to use only IP's**
  - Make sure /etc/hosts is really up to date on all machines
    - Don't report 127.0.0.1 as a machine's address to other machines
  - Make sure /etc/nsswitch.conf has a "hosts" entry that starts with "files"

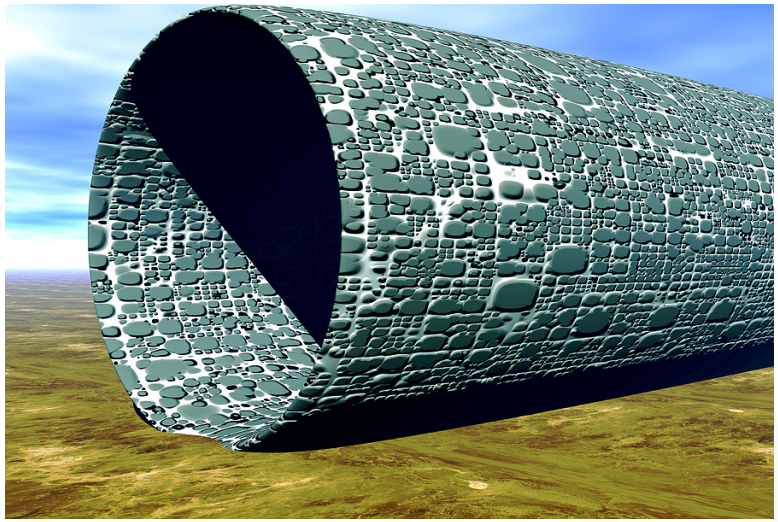
Comment found in Hadoop networking code

```
Ideally we should wait after transferTo returns 0. But
because of a bug in JRE on Linux
(http://bugs.sun.com/view_bug.do?bug_id=5103988),
which throws an exception instead of returning 0, we wait
for the channel to be writable before writing to it. If
you ever see IOException with message "Resource
temporarily unavailable" thrown here, please let us know.
Once we move to JAVA SE 7, wait should be moved to correct
place.
```

- Hadoop stresses every bit of networking code in Java and tends to expose all the cracks
- This bug was fixed in JDK 1.6.0_18 (after 6 years)
- Stick to explicitly recommended JDK minor versions
  - Avoid JDK 7!

Hadoop is a bandwidth hog

- **Can easily saturate a 1Gb network**
- **Good improvements reported on 10Gbps networks**
- **Enable bonding for faster networking**
  - /etc/modprobe.conf
    - alias bond0 bonding
    - options bond0 miimon=80 mode=5
  - /etc/sysconfig/network-scripts/
    - ifcfg-eth*
    - ifcfg-bond*
    - route-eth*
    - route-bond*



Hadoop is an everything hog

- **It's not just bandwidth**
  - It will use every disk you have, all the memory you can afford, …
- **Edit /etc/security/limits.conf**
  - Increase open file limits
    - hadoop – nofile 32768
  - Increase number of concurrent processes
    - hadoop – nproc 32768

Hadoop especially loves/hates your network

- **Meet your new best friend: /etc/sysctl.conf**
- **Increase kernel packet buffer**
  - net.core.netdev_max_backlog = 4000
  - Value is per CPU and defaults to 1000
- **Increase socket connection waiting queue**
  - net.core.somaxconn = 4000
  - Defaults to 128
- **Turn off selective acks – not needed for low latency networks**
  - net.ipv4.tcp_sack = 0
  - net.ipv4.tcp_dsack = 0

More networking tweaks

- **Make keepalives useful**
  - net.ipv4.tcp_keepalive_time = 600
  - net.ipv4.tcp_keepalive_probes = 5
  - net.ipv4.tcp_keepalive_intvl = 15
  - Defaults: wait 2 hours, send 9 probes 75 seconds apart. If probes aren't ack'd, consider other side dead and close socket.
- **Don't wait so long for socket close ack**
  - net.ipv4.tcp_fin_timeout = 30
  - Default: 180 seconds
  - Don't go too small because it's cheaper to reconnect before the expiration

Yet more networking

- **You're going to need bigger buffers**
  - Hadoop moves 64MB blocks (by default)
  - net.ipv4.tcp_rmem = 32768 436600 4194304
  - net.ipv4.tcp_wmem = 32768 436600 4194304
  - Values: min default max
  - Defaults
    - Read – 4096 87380 4194304
    - Write – 4096 16384 4194304

Yes, more network stuff

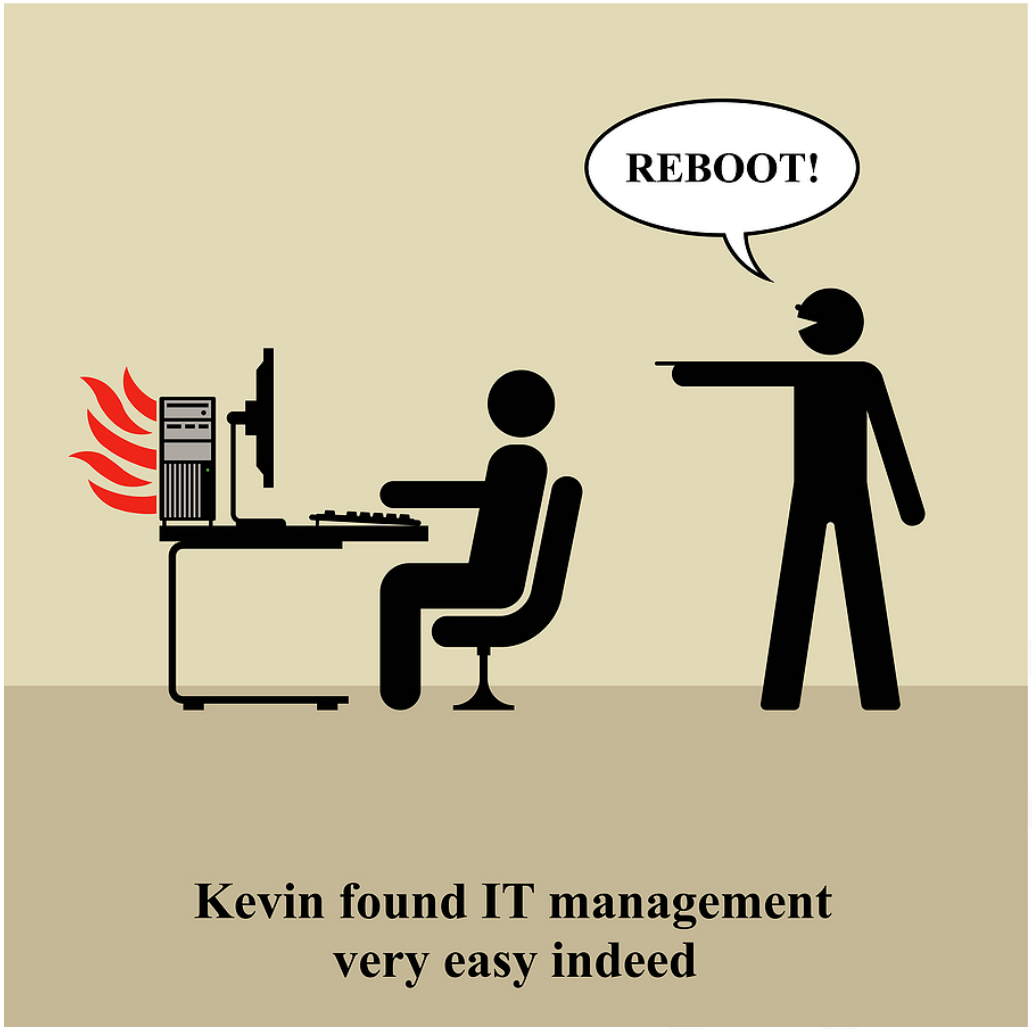
- **Don't try so hard to resend a packet**
  - net.ipv4.tcp_retries2 = 10
  - Default: 15
  - Note: there's an exponential back-off so this is still several minutes of trying
- **Don't try so hard to resend socket close acks**
  - net.ipv4.tcp_synack_retries = 3
  - Default: 5

Kernel version-specific tuning

- epoll limits required as of 2.6.27, then no longer required in newer kernels such as 2.6.33+
- http://pero.blogs.aprilmayjune.org/2009/01/22/hadoop-and-linux-kernel-2627-epoll-limits/
- /etc/sysctl.conf
  - fs.epoll.max_user_instances = 2048

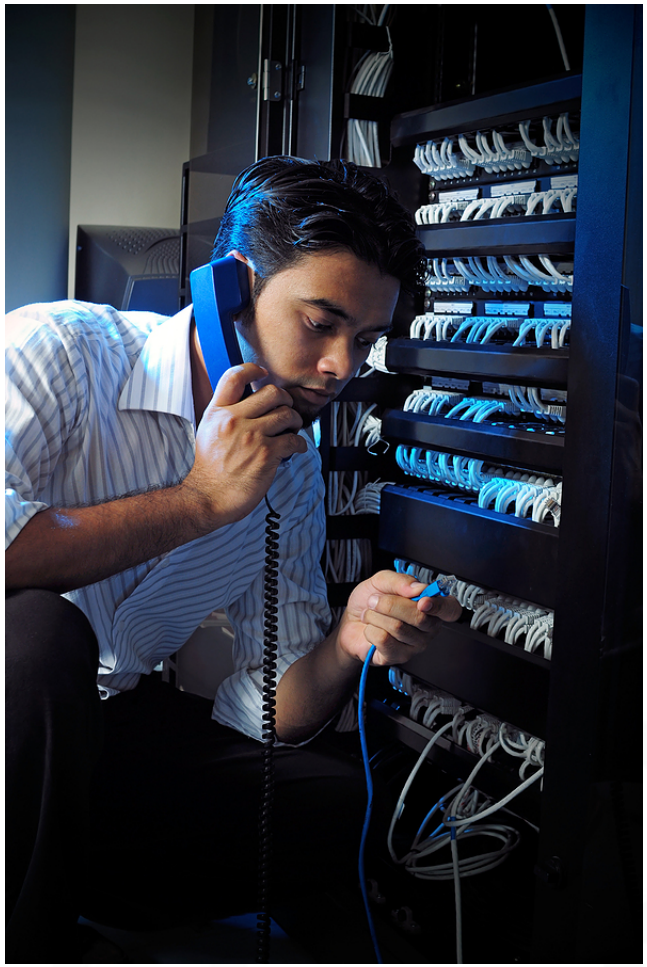
Hadoop & HBase Configuration

- **Follow all HBase configuration advice here:**
  - http://wiki.apache.org/hadoop/Hbase/Troubleshooting
  - Yes, that's a whole lot of configuration
  - Skip steps at your own peril!
- **If you hit datanode timeouts while writing to sockets:**
  - dfs.datanode.socket.write.timeout = 0
  - Even though it should be ignored…


REBOOT!
Kevin found IT management
very easy indeed

Hadoop Troubleshooting

- **Turn off IPTables**
  - chkconfig --del iptables
  - chkconfig --del ip6tables
- **Disable SELinux**
  - /etc/selinux/config
    - SELINUX=disabled

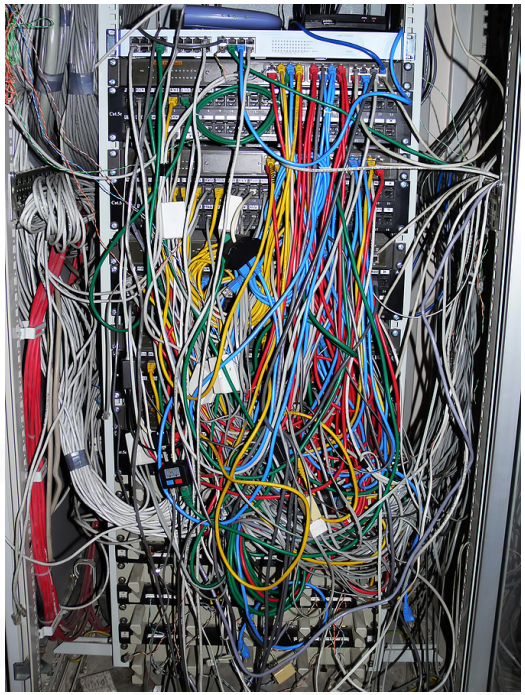


Hadoop Troubleshooting (cont.)

- **Disable IPV6**
  - /etc/sysctl.conf
    - net.ipv6.conf.all.disable_ipv6 = 1
    - net.ipv6.conf.default.disable_ipv6 = 1
    - net.ipv6.conf.lo.disable_ipv6 = 1
  - /etc/sysconfig/network
    - NETWORKING_IPV6=no
  - echo "install ipv6 /bin/true" >> /etc/modprobe.d/disable-ipv6

Configuration Debugging Tips

- **Many problems only show up under severe load**
  - Sustained, massive data loads running for 2-24 hours
- **Change only one parameter at a time**
  - Yes, this can be excruciating
- **Ask the mailing list or your support provider**
  - They've seen a lot, likely including your problem…but not always
  - Don't be afraid to dig in and read some code

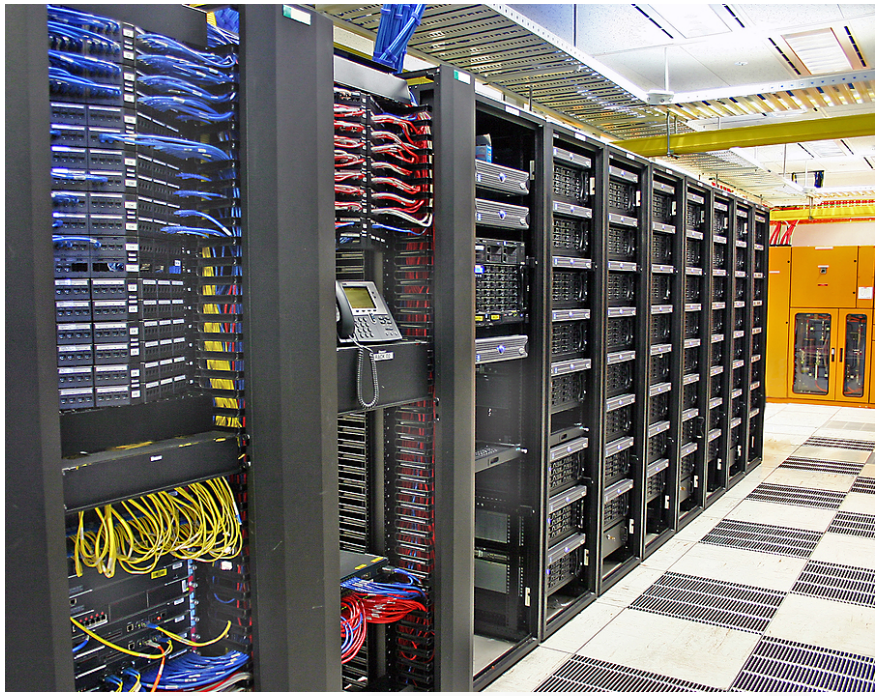


Commodity Hardware

- **“Commodity hardware” != 3 year old desktop**
- **Dual quad-core, 32GB RAM, 4+ disks**
- **Don’t bother with RAID on Hadoop data disks**
  - Be wary of non-enterprise drives
- **Expect ugly hardware issues at some point**

OpenLogic's Hadoop Deployment

- **Cloudera's Distribution for Hadoop (CDH)**
  - OpenLogic's CDH-based scanner used on CDH code
- **Dual quad-core and dual hex-core Dell boxes**
- **32-64GB RAM**
  - ECC (highly recommended by Google)
- **6 x 2TB enterprise hard drives**
- **RAID 1 on two of the drives**
  - OS, Hadoop, HBase, Solr, NFS mounts (be careful!), job code, etc.
  - Key "source" data backups
- **Hadoop datanode gets remaining drives**
- **Redundant enterprise switches**
- **Dual- and quad-gigabit NIC's**



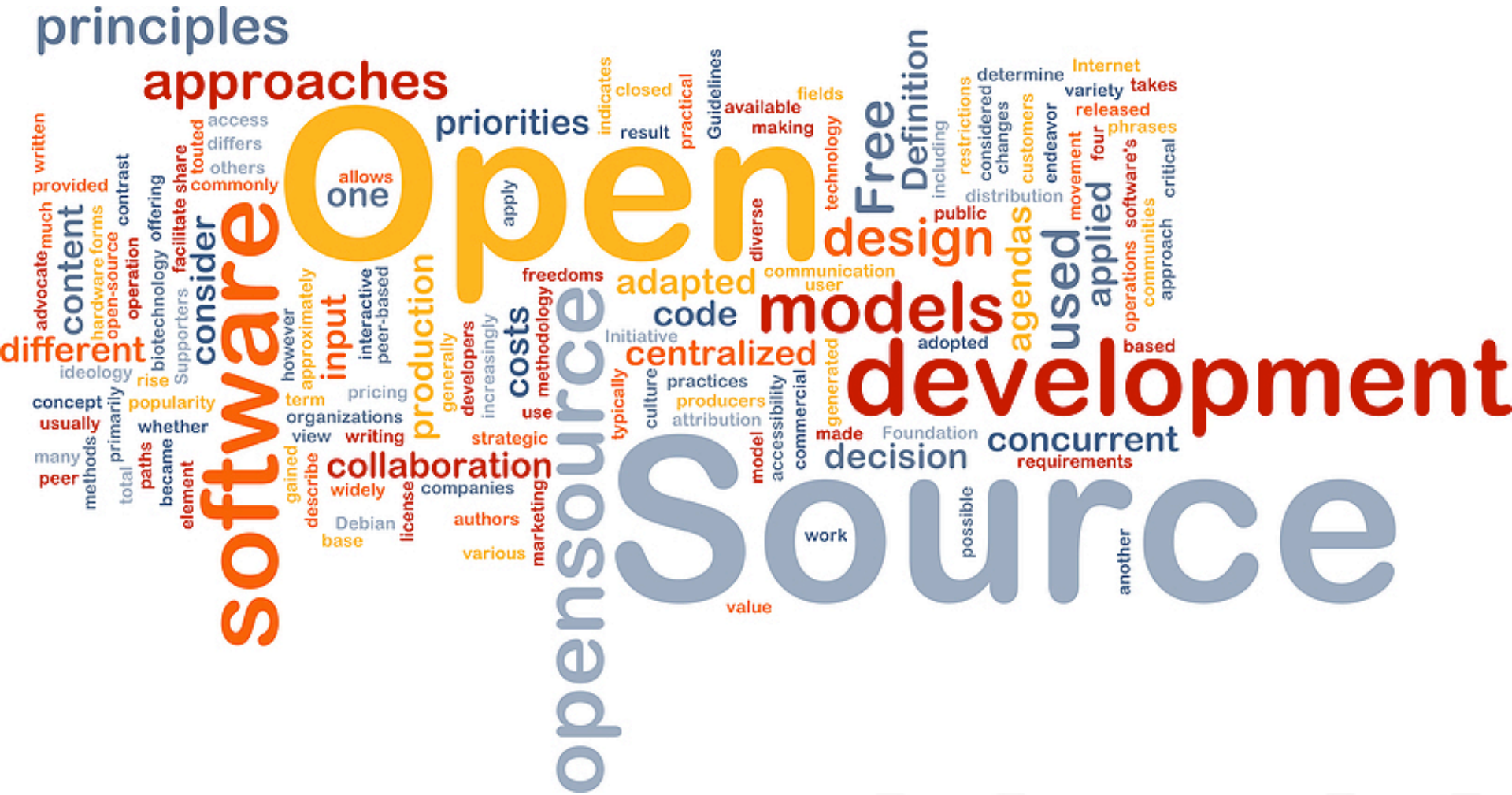
OpenLogic's Cloud Product Solution

- **Homegrown product**
  - On Amazon EC2 today w/multi-region & Rackspace in the works
  - Ruby on Rails, OpenSSL library for AES security
  - CouchDB (hosted by Cloudant)
- **Relies heavily on CloudInit**
  - Ubuntu package (also on EC2 in Amazon's CentOS-based Amazon Linux AMI)
  - Makes it easy to spin up new cloud-based servers
  - "user-data" provides lots of flexibility
    - Update the OS, install packages, run arbitrary scripts
    - Download code, phone home
    - Optional: multi-part, gzipped

CloudInit tips

- **Ensure VM is in a good known state**
  - Key packages installed
  - Correct versions
- **Install your "base layer"**
  - Things you depend on
  - Examples: Ruby, Chef, Java, Nginx, etc.
- **Do all this first and conditionally**
  - Look for an "init" file
  - If not there, do your set up then touch it
  - Example: .ol-init

Not Possible Without Open Source



Not Possible Without Open Source…

- Hadoop, HBase, Solr
- Apache, Tomcat, ZooKeeper, HAProxy
- Stargate, JRuby, Lucene, Jetty, HSQLDB, Geronimo
- Apache Commons, JUnit
- Ubuntu, CentOS
- Dozens more

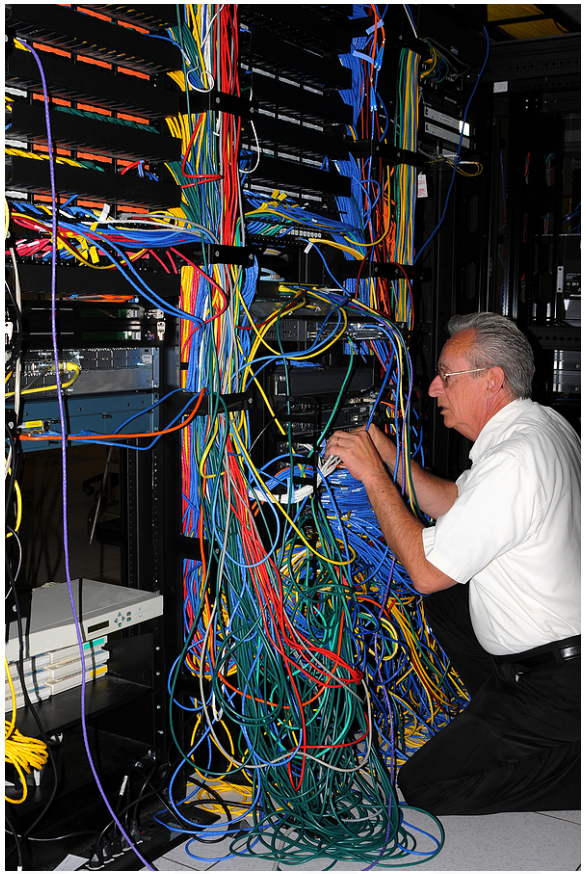
- Too expensive to build or buy everything

…But Still Cutting Edge

- **Hadoop**
  - SPOF around Namenode, append functionality
- **HBase**
  - Backup, replication, and indexing solutions in flux
- **CentOS**
  - Network driver & MSI issues, dropped packets
- **Ubuntu CloudInit**
  - Doesn't always run correctly on EC2
  - Not pre-installed on all public cloud images (e.g., Rackspace)

Expect Things to Fail – A Lot

- **Virtual Machines**
  - Won't start, hang for no reason, slow down and speed up occasionally
- **Operating System**
  - Kernel panics, zombie processes, dropped packets
- **Hadoop and Friends**
  - Failed: Hadoop datanodes, HBase regionservers
- **Your Code and Big Data**
  - Stray Hadoop jobs, strange corner cases in your data leading to program failures



Final Thoughts

- **Details matter**
  - Pay close attention to minor versions, version compatibility, and configuration details
  - Don't assume latest = best
- **You will invest in training and support**
  - Internal or external is your choice, but there's no free lunch
- **Architect for failure**
  - Don't be surprised when something breaks
  - Don't cross your fingers and hope for the best
  - Do set up automated monitoring and management

Q & A

Any questions for Rod?

rod.cope@openlogic.com

Slides: http://www.openlogic.com/downloads/presentations.php