Solr on Amazon's EC2

Erick Erickson, Lucid Imagination
erick.erickson@lucidimagination.com, October, 2011

Presented by

lucid IMAGINATION

Lucene

Apache Solr

Who am I

- Erick Erickson, Lucid Imagination Principle Engineer
- Solr/Lucene Committer
- Solr performance, query tuning and scaling
  - **Editorial system, Chicago Tribune**
  - **Started with Lucene for genealogy**
  - **Sailing, scuba diving and gardening**

Topics

- What made the cloud possible
- Scaling solr
- Solr on Amazon's EC2 cloud
- Using EC2 to scale Solr
- Taking advantage of EC2 possibilities
- Computing on the Cloud, "gotchas" and possibilities

The Challenge

- Understand how Solr works in the cloud (AWS/ EC2)
  - **"Rent stuff"**
    - *Machine images (AMIs) of various "sizes"*
    - *Other services and resources (e.g. data transfer)*
- What gremlins lurk?
- What possibilities exist for dynamic scaling (autoscaling)?
- Understand the tools and limitations.

Background

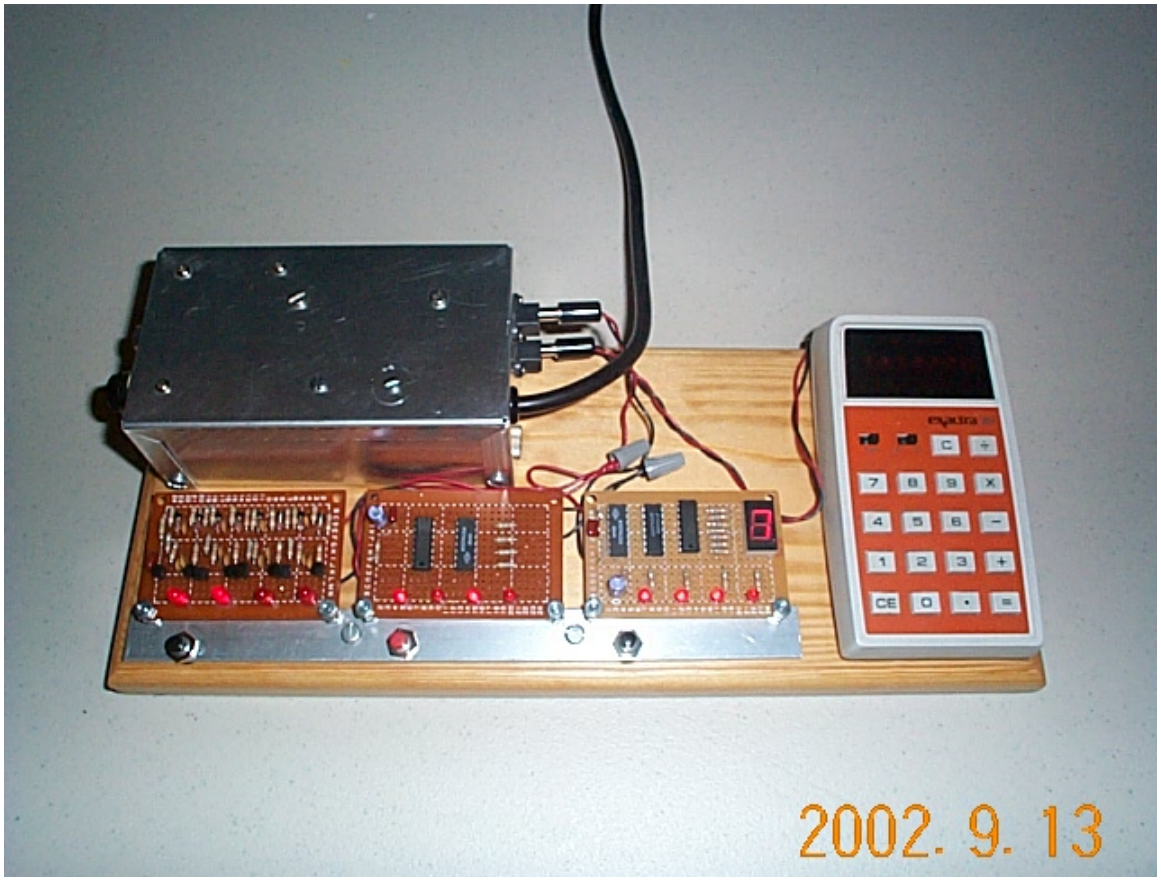
- A bit of history
  - **Ancient history (1940's) courtesy of professors Scott and Jarvis**
- "Data center in a box"
  - **Portable "drop in" data centers**
- Modern (mega) server farms
  - **One way of using the "data center in a box"**

Flip-Flop, 1940s

14 watts, 200 DC volts

Courtesy of Professor John Jarvis

IC-based counters, circa 1960



Courtesy of Professor John Jarvis

Too small to see

- Professor Scott, circa 1983 – “now there are 64,000 transistors on a 3x5mm chip”.
- 2006, Wikipedia – “Up to 1 million transistors per square millimeter”

Data Center In a Box

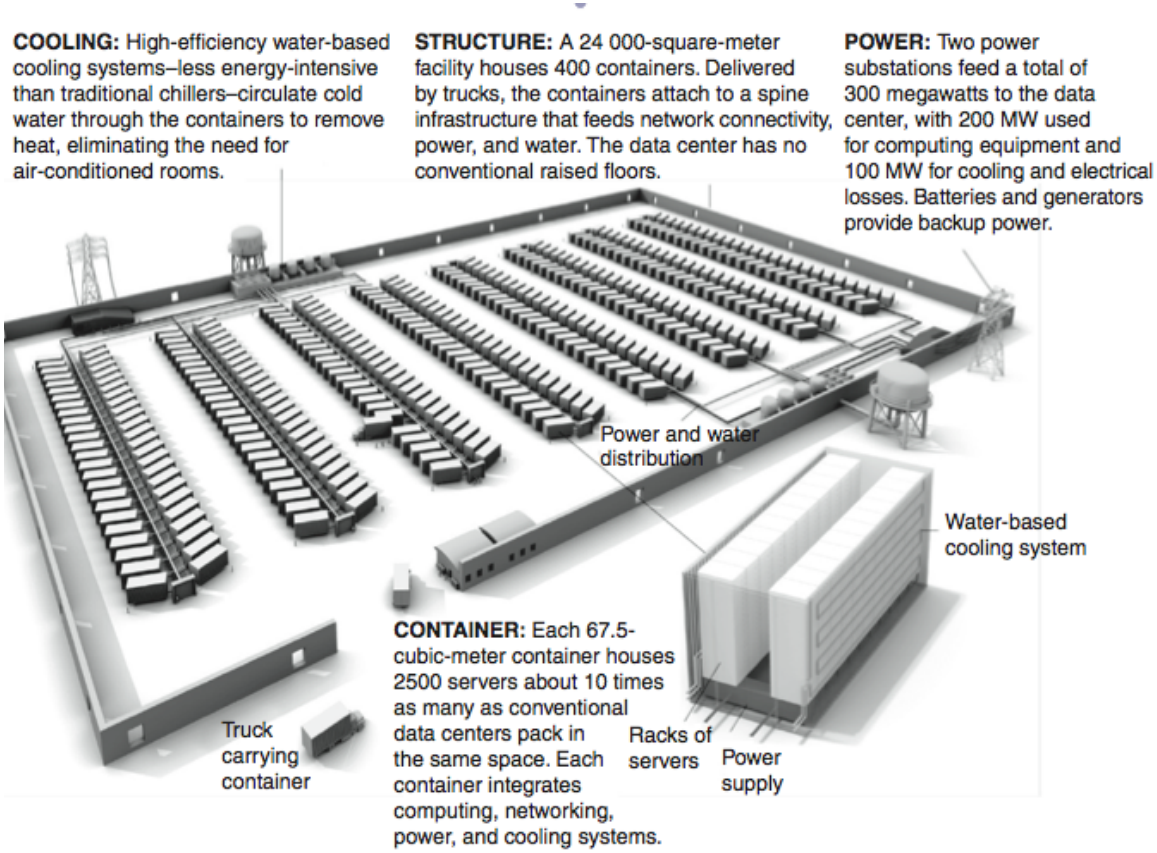
- Developed by Sun, 2006 - 2008.
  - **8’ x 8’ x 20’ (2.4m x 2.4m x 6.0m) cargo container**
  - **Contained 250 SunFire servers**
  - **Water cooled**
- Modern versions contain over 2,500 servers
  - **8’ x 8’ x 40’ (2.4m x 2.4m x 12m)**
  - **Up to 29.8 Petabytes of storage**

"We expected skepticism, and we got it," he said. "This is a radical concept in the data-center world. What we didn't expect, however, was to give a presentation and have people come up afterward saying, 'I need 10 of these tomorrow.' " – Greg Papadopoulos, Sun's chief technology officer, 2007.

Google's Dalles, OR center

Source: Melanie Conner, *New York Times*

Schematic, "Mega" data center



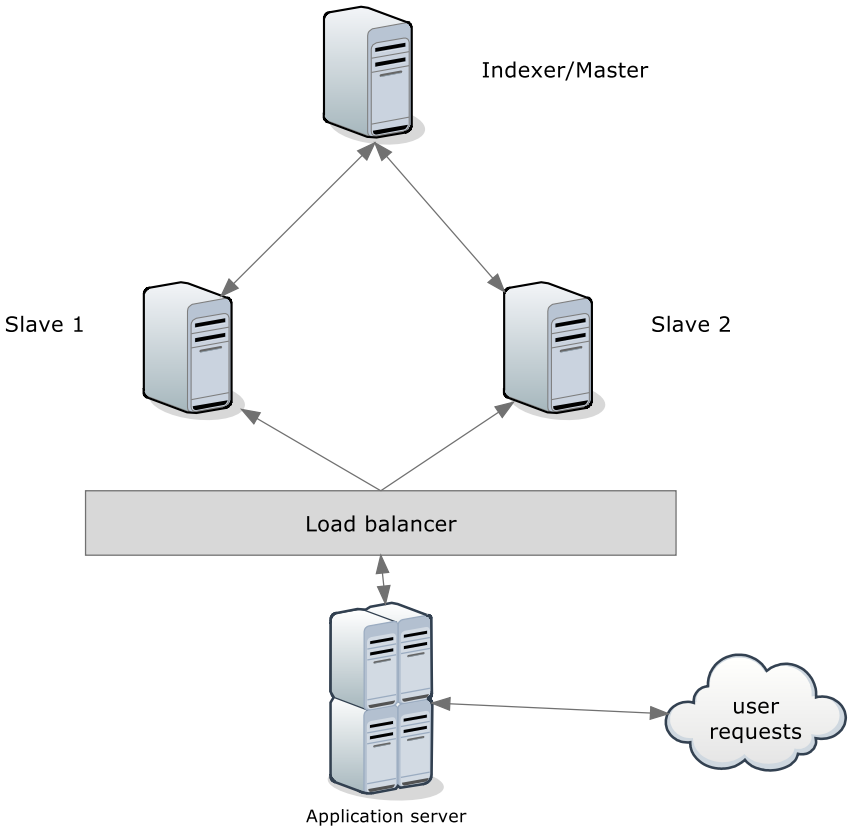
Source: *IEEE Spectrum magazine*

Solr at scale

Scaling techniques for Solr: background for moving to the cloud.

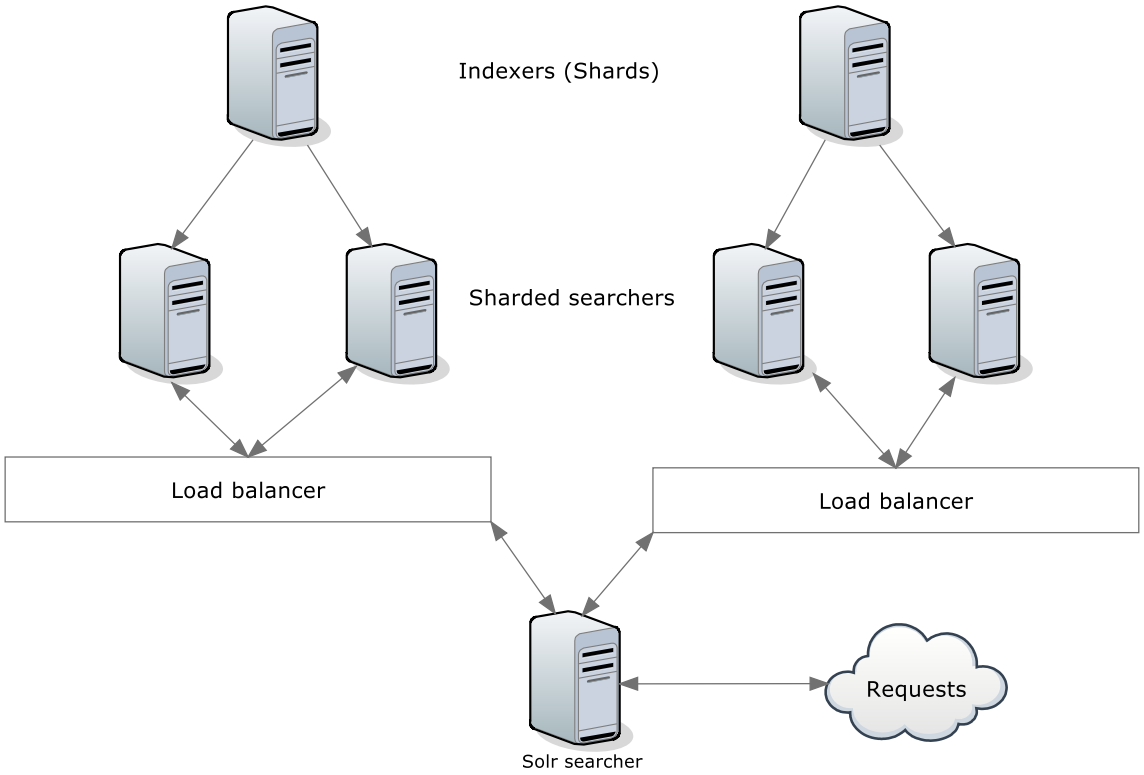
- Replication – Used when satisfactory search response time available from a single server
  - **Interesting question re: EC2. Is the best strategy to move to bigger machine image?**
- Sharding
  - **When the index is too large for a single machine to handle**

Basic Solr Installation



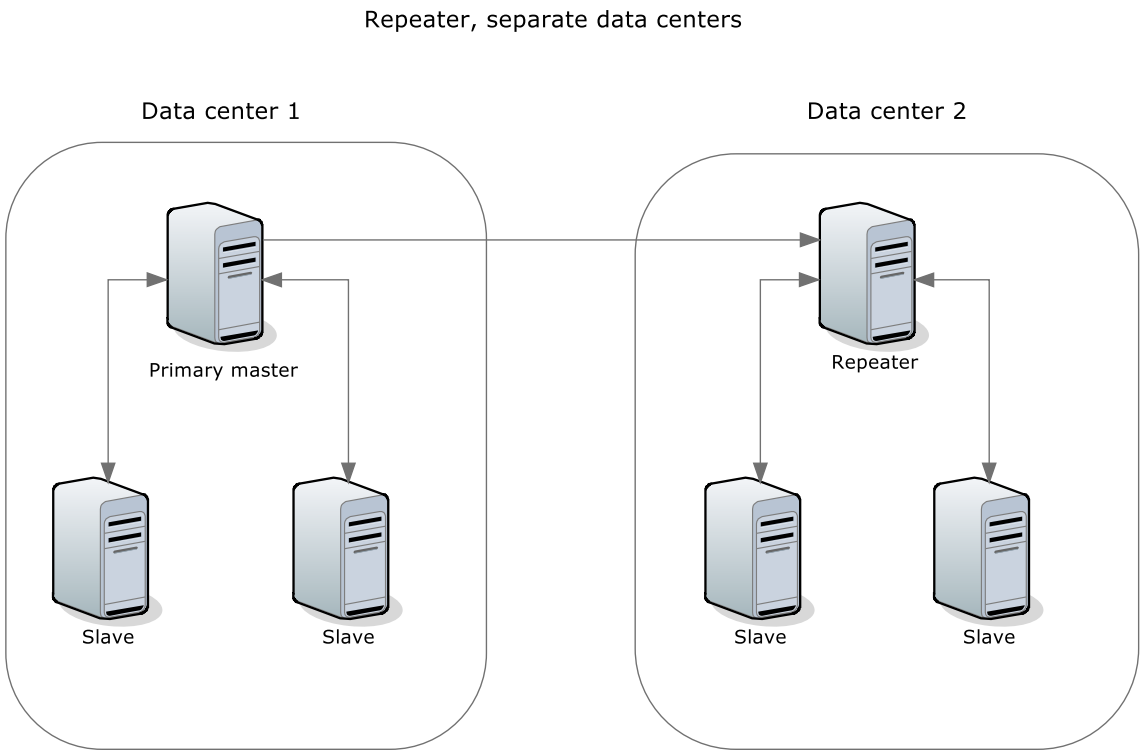
Used when the response time for queries is acceptable but more queries-per-second are required. Can be scaled to as many slaves as necessary to satisfy throughput requirements.

Sharded Solr Installation



When the index becomes too large, the index can be “sharded” – each “shard” is a separate part of a single logical index. Shards can then be replicated as necessary.

Separate Data Centers



This configuration allows physically separate data centers to efficiently share a common index.

Now, what you've been waiting for…

Solr on the cloud (AWS/EC2)

What is “EC2” anyway?

- Why was it developed?
- What is it, really
- AMI’s “Amazon Machine Image”
- ELB and S3 (storage)
- Solr in this environment
- Tools
- Gotchas

Why was it developed?

- Genesis of EC2 was Amazon's explosive growth.
- Developed to serve Amazon's internal needs for fast, flexible growth
- 18 months after launch as a commercial product, customers' usage surpassed Amazon's usage

What is it, really?

- Virtual machines (think VMWare) that can be instantiated and released at will
- Backed by more-or-less permanent storage
  - **EBS is more limited, but faster than S3**
- Infrastructure to manage many, many images
- Simple conceptually, but the key is *you don't have to develop it all yourself*

Start with just three concepts

- An AMI (Amazon Machine Image)
- Some storage (S3 and EBS)
- Scripting
  - **"Anything you can do manually, or through the GUI, you can script"**
  - **Scripting (command-line interface) has capabilities the GUI console doesn't**

AMI's

- Amazon Machine Images, often called an "instance"
- Various sizes of machines (more later) that you can rent by the hour
  - **Additional costs for storage and transfer**
  - **Costs constantly changing.**
- You can make your own AMI or start with one of the "stock" offerings

AMI's continued

- Various operating systems allowed:
  - **Windows, Linux, SUSE, Open Solaris, etc.**
- Various software
  - **Databases, web servers, etc.**
- Can start with a base configuration, add as per your own requirements, and save image
  - **Doubtful that putting your index on the image a good idea**

AMI instance types (Range)

- Small Instance (Default)
  - 1.7 GB of memory
  - 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)
  - 160 GB of local instance storage
  - 32-bit platform
  - $0.085/hr (USD, Linux)
- High-Memory Quadruple Extra Large Instance
  - 68.4 GB of memory
  - 26 EC2 Compute Units (8 virtual cores with 3.25 EC2 Compute Units each)
  - 1690 GB of local instance storage
  - 64-bit platform
  - $1.60/hr (USD, Linux only)

Which AMI is best for Solr?

- It depends ™.
  - **Interesting writeup at: http://www.artirix.com/media/artirix-blog/scaling-solr-on-amazon-ec2-go-big/**
  - **The exciting bit is that you can test easily and inexpensively.**

S3 and EBS (storage)

- S3 is permanent, secure, backed up
  - **May want to put periodic snapshots of index on S3**
  - **Use as a basis for EBS volume (below)**
- EBS (Elastic Block Storage) is only accessible by a single instance
  - **Consider taking a snapshot of master index periodically (to S3) and using that as basis for EBS volume for new instance**
  - **EBS data may or may not be deleted when instance goes away**

Storage (cont)

- "Ephemeral"
  - Goes away when instance dies
  - "Closer to metal"
  - May be a good place for slave indexes

Scripting

- Amazon provides a set of APIs for carrying out various actions
  - **Amazon's web-based GUI useful for getting started, but will quickly want to use command-line APIs.**
  - **Download locally and set up environment**
  - **http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference/**
- These can be used in scripts on AWS.
- Download libraries for use from your desktop
- ELB and EC2 are in different downloads

Scripting examples

- ec2-run-instances -k dev-erick -t m1.large -b /dev/sdb=ephemeral0 ami-8e1fece7
- elb-create-lb erick-elb -z us-east-1d --listener "protocol=http, lb-port=80, instance-port=80”

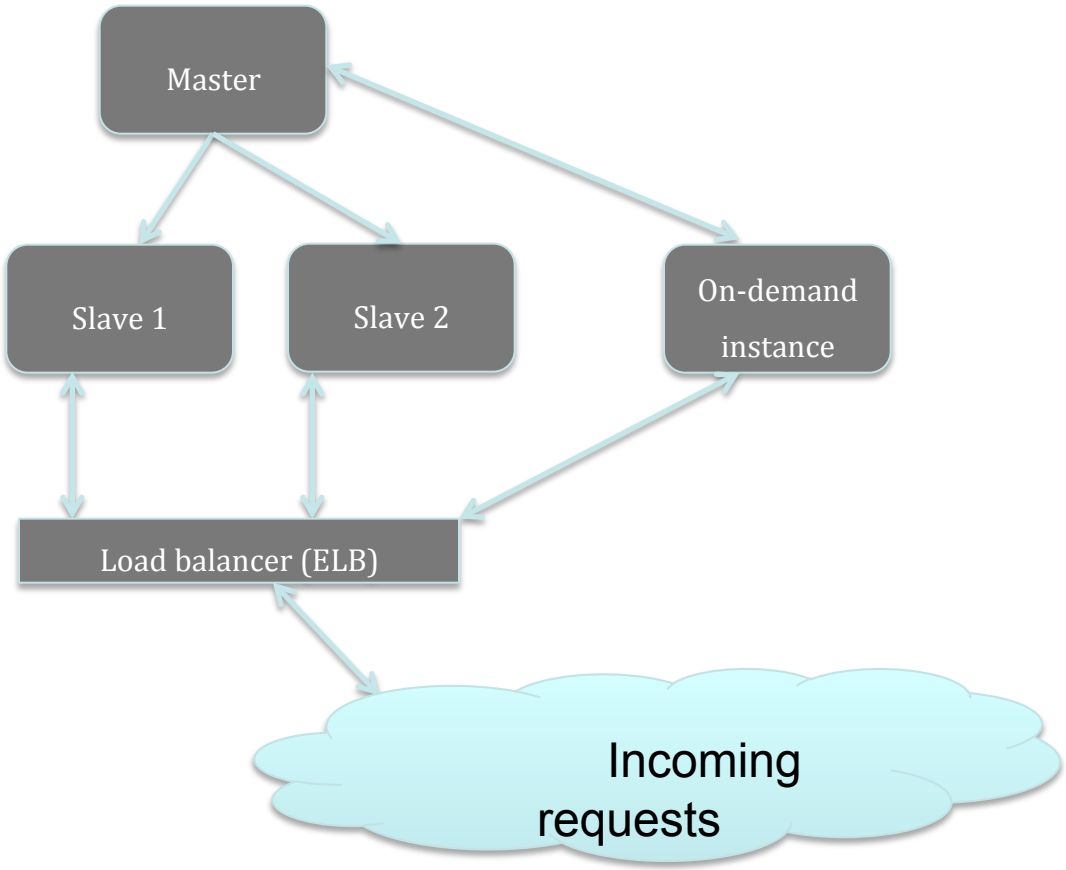
Solr on EC2

- Simple approach, "manually" host Solr on EC2
  - **Pick a machine image you're comfortable with**
  - **Manually create/configure instance (and save it)**
  - **Create scripts to handle load (optional)**
    - *Bring up instances on a pre-defined schedule that you manually tune to reflect your traffic pattern*
  - **Could even index locally and periodically transfer index up**
  - **It "just works"**
  - **Just treat EC2 as a convenient place to run Solr without having to buy hardware yourself**

Scaling Wide

- But what about scaling?
  - **Scaling wide (replicating entire index) simple, just bring up a new instance.**
  - **Front it with ELB (Elastic Load Balancer)**
  - **Use cloudwatch to trigger alerts that require manual intervention.**

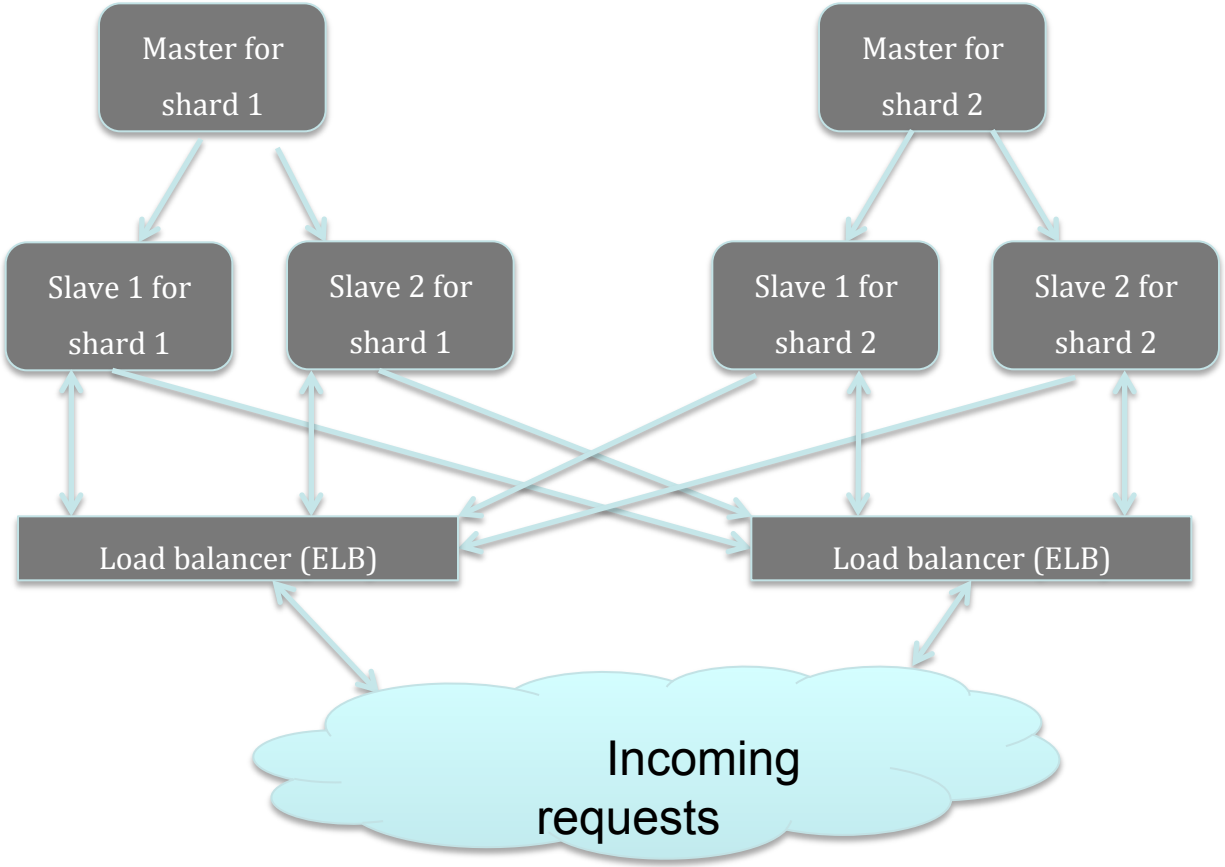
Scaling Wide on EC2



Scaling Deep

- Remember that sharding is useful when the QPS drops on your target hardware such that "scaling wide" is impractical
  - **EC2 makes this question harder to answer, since:**
    - *You can change the hardware size just by using a different AMI*
    - *Even if you're using the largest AMI available, by the time you look again there may be even larger instances*

Scaling Deep



Tools

- AWS provides multiple tools for managing your installation. Among the important ones for Solr are:
  - **Customizable machine images**
  - **Elastic Load Balancing**
  - **Cloudwatch/Autoscaling**

Cloudwatch

- The “trigger” for spinning up another instance
- Script new instances
  - **Probably need to wait for replication, although new EC2 capabilities allow you to register the instance with ELB at your leisure**
- Timed or HTTP interface for warmed searchers before registering with ELB
  - **Timed: “I know it takes X seconds (plus) to replicate”, register after that time has elapsed**
  - **HTTP: Use HTTP commands to answer the question “is replication complete” on the new instance before registering**

Autoscaling

- EC2/Cloudwatch allows new instances to be instantiated based on metrics
  - **QPS, CPU load, latency etc.**
  - **Custom alerts**
- Allows instances to be removed automatically based on other metrics
  - **Reduce to minimize costs**

Elastic Load Balancing

- Dynamic load balancing across instances as they come and go
  - **Requires registration**
- Script registration
  - **Allows registration *after* replication for new Solr instances**
  - **Want to include warmup times**

Security

- VPC (Virtual Private Cloud)
- Encrypting disks?
  - **"If encryption is important to you, we recommend that you run an encrypted file system on top of your Amazon EBS volume."**
- Can encrypt data, but searching can be tricky
- "Just trust Amazon"
- HTTPS/SSL
- Amazon security
  - **"Security Groups" are often used**

Gotchas

- Requests to the wrong host:
  - http://www.greenhills.co.uk/2011/06/22/elb-traffic-for-the-wrong-host.html
  - **Implies that IP's *will change* in the "normal" case**
- Latency
  - **Spinning up a new instance, shouldn't register with ELB before index replication is done.**
  - **Script this process.**
  - **Recent changes to EC2 facilitate this.**

Gotcha's (cont)

- Costs (always changing)
  - **Data transfer costs vary**
    - *Free within same availability zone*
    - *Free to upload*
    - *Charges apply between availability zones, across "regions", and out of EC2*
- Machine instances
  - **Costs vary by machine capability. Must test to find optimum.**
  - **Writeup at: http://www.artirix.com/media/artirix-blog/scaling-solr-on-amazon-ec2-go-big**

Gotcha's (cont)

- Response time less predictable (VMs)
- Instances can "just go away" (Chaos Monkey)
- Indexing speed still limited by network between you and EC2
- Amazon could pull the plug
- Security setup takes some getting used to
- At some threshold EC2 can be more expensive
- Allocate time to create in-house expertise

Cloud computing is exciting!

- You can be up and running tomorrow
- You can experiment with different “hardware equivalents” easily
- You could use EC2 to index huge amounts of data and bring the indexes down to your local data center
- You can scale wide and/or deep quickly
- Your business is probably *not* running a data center!

Excitement continued

- The cloud opens up new prototyping opportunities
- One test is worth a thousand opinions
- And if you're a startup…..

Parting whimper

- Just scratched the surface of EC2/Solr
- I pine for the days of yore
  - **Machines didn't change all that fast, or at least it took a year to requisition one!**
  - **Programming language in one slim 220 page volume (Kernighan & Ritchie 1941-2011)**
- Now, you may need a full-time staff member to just understand what AWS/EC2 has added *lately!*

Wrap Up

- Cloud computing presents exciting new possibilities made possible by the phenomenal advances in technology
- EC2 has its own tools, technologies and gotchas
- Serious move to the cloud will require in-house specialists, allocate-them

Thank you

- Lucid Imagination: http://www.lucidimagination.com/
- Recent improved control of when instances are visible to ELB: http://aws.typepad.com/aws/2011/07/auto-scaling-notifications-recurrence-and-more-control.html
- Professor Jarvis' website: http://www.knology.net/~johnfjarvis/HistCompNotes.html

Links (cont)

- Amazon AWS home page: http://aws.amazon.com/
- EC2 APIs: http://docs.amazonwebservices.com/AWSEC2/latest/CommandLineReference/
- I/O benchmarks: http://blog.cloudharmony.com/2010/06/disk-io-benchmarking-in-cloud.html

Contact

Erick Erickson
Erick.erickson@lucidimagination.com