Inside Picnik: How we built Picnik and what we learned along the way

Mike Harrington
Justin Huff

Web 2 Expo, April 3rd 2009
Thanks! We couldn’t have done this without help!
Our happy problem
Fun with Flex
We ❤ Flash

- Browser and OS independence
- Secure and Trusted
- Huge installed base
- Graphics and animation in your browser
We ❤ Flash

- Decent tool set
- Decent image support
- ActionScript
Flash isn’t Perfect

- Not Windows, Not Visual Studio
- ActionScript is not C++
- SWF Size/Modularity
- Hardware isn’t accessible
- Adobe is our Frienemy
- Missing support for jpg Compression
Flash is a control freak
Forecast: Cloudy with a chance of CPU
Your users are your free cloud!

- Free CPU & Storage!
- Cuts down on server round trips
- More responsive (usually)
- 3rd Party APIs Directly from your client
- Flash 10 enables direct load
You get what you pay for.

- Your user is a crappy sysadmin
- Hard to debug problems you can’t see
- Heterogeneous Hardware
- Some OS Differences
Client Side Challenges

• Hard to debug problems you can’t see
• Every 3\textsuperscript{rd} party API has downtime
• You get blamed for everything
• Caching madness
Dealing with nasty problems

• Logging
• Ask customers directly
• Find local customers
• Test, test, test
Would we use Flash again?

Absolutely.
Apache

Check.

(static files)
MySQL

Yeah, that too.
P

Is for Python
Python

90% of Picnik is API.
CherryPy

- We chose CherryPy
- RESTish API
- Cheetah for a small amount of templating
Ok, that was boring.
Fun Stuff
Virtualization

- XEN
- Dual Quad core machines
- 32GB RAM
- We virtualize everything
  - Except DB servers
Virtualization Pros

- Partition functions
- Matching CPU bound and IO bound VMs
- Hardware standardization (in theory)
Virtualization Cons

• More complexity
• Uses more RAM
• IO/CPU inefficiencies
Storage

- We started with one server
  - Files stored locally
  - Not scalable
Storage

- Switched to MogileFS
  - Working great
  - No dedicated storage machines!
Storage

- Added S3 into the mix
Storage – S3

- S3 is dead simple to implement.
- For our usage patterns, it's expensive
- Picnik generates lots of temp files
- Problems keeping up with deletes
- Made a decision to focus on other things before fixing deletes
Load Balancers

- Started using Perlbal
- Bought two BigIPs
  - Expensive, but good.
- Outgrew the BigIPs
- Went with A10 Networks
  - A little bumpy
  - They've had great support.
CDNs Rock

- We went with Level3
- Cut outbound traffic in half:

```
<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Current</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound</td>
<td>303.04 M</td>
<td>75.18 M</td>
<td>362.54 M</td>
</tr>
<tr>
<td>Outbound</td>
<td>251.54 M</td>
<td>86.64 M</td>
<td>319.62 M</td>
</tr>
</tbody>
</table>
```
Rendering

• We render images when a user saves
• Render jobs go into a queue
• Workers pull from the queue
• Workers in both the DC and EC2
Rendering

- Manager process maintain enough idle workers
- Workers in DC are at ~100% utilization
EC2

- Our processing is elastic between EC2 and our own servers
- We buy servers in batches
If a rack falls in the DC, does it make any sound?
Monitoring

• Nagios for Up/Down
  – Integrated with automatic tools
  – ~120 hosts
  – ~1100 service checks
• Cacti for general graphing/trending
  – ~3700 data sources
  – ~2800 Graphs
• Smokeping
  – Network latency
Redundancy

- I like sleeping at night
- Makes maintenance tasks easier
- It takes some work to build
- We built early (thanks Flickr!)
War Stories
ISP Issues

• We're happy with both of our providers.
• But...
  – Denial of service attacks
  – Router problems
  – Power
• Have several providers
• Monitor & trend!
ISP Issues

- High latency on a transit link:

- Cause: High CPU usage on their aggregation router
- Solution: Clear and reconfig our interface
Amazon Web Services

• AWS is pretty solid
• But, it's not 100%
• When AWS breaks, we're down
  – The worst type of outage because you can't do anything.
• Watch out for issues that neither party controls
  – The Internet isn't as reliable point to point
EC2 Safety Net

- Bug caused render times to increase

Call Time - render_processing_time

- Time: Last: 4.24  Average: 4.08  Max: 14.14
EC2 Safety net

- Thats OK!

**EC2 VMs**

<table>
<thead>
<tr>
<th></th>
<th>m1.small</th>
<th>Current</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Thu</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fri</td>
<td></td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Sat</td>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Mon</td>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Tue</td>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>c1.medium</th>
<th>Current</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wed</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Thu</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fri</td>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Sat</td>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Mon</td>
<td></td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Tue</td>
<td></td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>
Flickr Launch

- Very slow Flickr API calls post-launch
- Spent an entire day on phone/IM with Flickr Ops
- Finally discovered an issue with NAT and TCP timestamps
- Lessons:
  - Have tools to be able to dive deep
  - Granular monitoring
Firewalls

• We had a pair of Watchguard x1250e firewalls.
  – Specs: “1.5Gbps of firewall throughput”
  – Actual at 100Mbps:
authorize.net

- Connectivity issues
- Support was useless
- Eventually got to their NOC
- We rerouted via my home DSL.
- Lessons:
  - Access to technical people
  - Handle failure of services gracefully
The end.
Thanks!

mike@picnik.com
justin@picnik.com
Credits/Resources

Talks that influenced us:
http://tinyurl.com/scalingwebsites

Images:
http://www.flickr.com/photos/kimberlyfaye/2785383504/
http://www.flickr.com/photos/piez/574804017/sizes/o/
http://www.flickr.com/photos/unclejojo/3224365412/sizes/o/
http://www.flickr.com/photos/phoenixdailyphoto/1467681879/sizes/l/
http://abductionlamp.com/
http://www.flickr.com/photos/msig/2716038191/
http://www.flickr.com/photos/touringcyclist/303577344/
http://www.flickr.com/photos/davidt2006/1407837008/