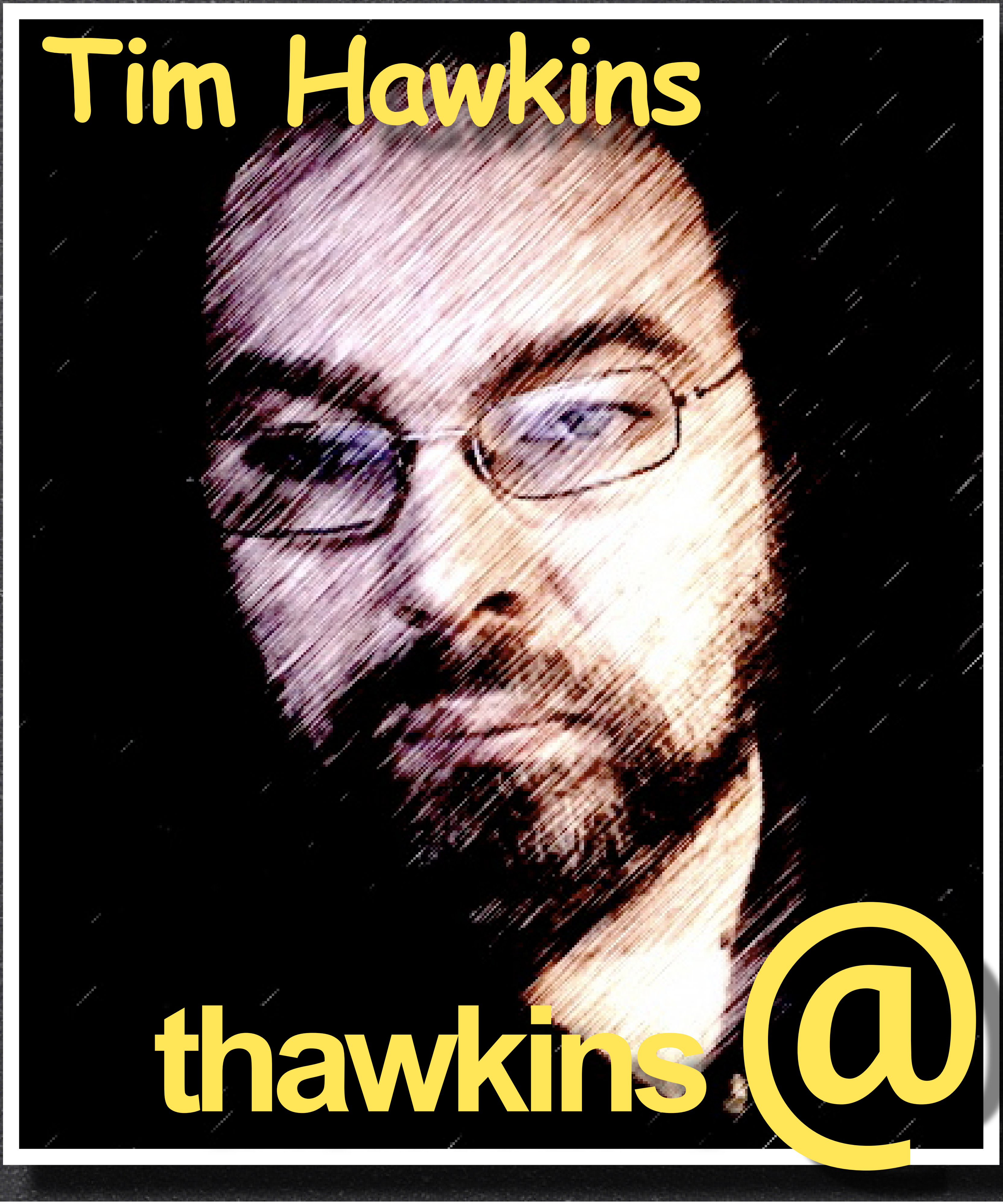


Tim Hawkins

Beefing it Up

Or "How to survive the digg or slashdot effect"

BarCampLondon2
BT Center
London
Feb 2007

thawkins@yahoo-inc.com

myWeb20app.com

You get this way cool new idea....

You slave away day and night to make it happen ...

You bring it online
on your single machine Hosted server .....

You are so proud, you tell ALL your mates ....
And they are seriously
impressed!, and rave about it on all their
blogs.....

And somebody
posts it to
slashdot,
technorati, and
digg .....

You make the home-page on
all 3 !, it was a way
cool idea after all!

And.......

Profit … ?

I don’t think so!

And there is more!

- You have missed out on all that Adsense revenue!, no birthday present for Gran this year...
- Your Hoster is pissed, because you took out 453 other sites hosted on that box, they suspend your account.
- You ran through your entire monthly bandwidth allocation in 23 mins, and to rub salt into the wounds, that was serving “exceeded maximum number of database connections” error messages from PHP.
- Your mates now think your site was called “Error 500” and thats been done before.

luzr!!!!

Maybe Not!

There are some strategies you can use to harden your site against burst traffic.

Accelerators
Caching
Session strategies.
Site Architecture

There are new options for hosting, that give you big boy type capabilities for very little cost.

If you are dealing with News, you will have to deal with this no matter what you do. News is by its very nature bursty.

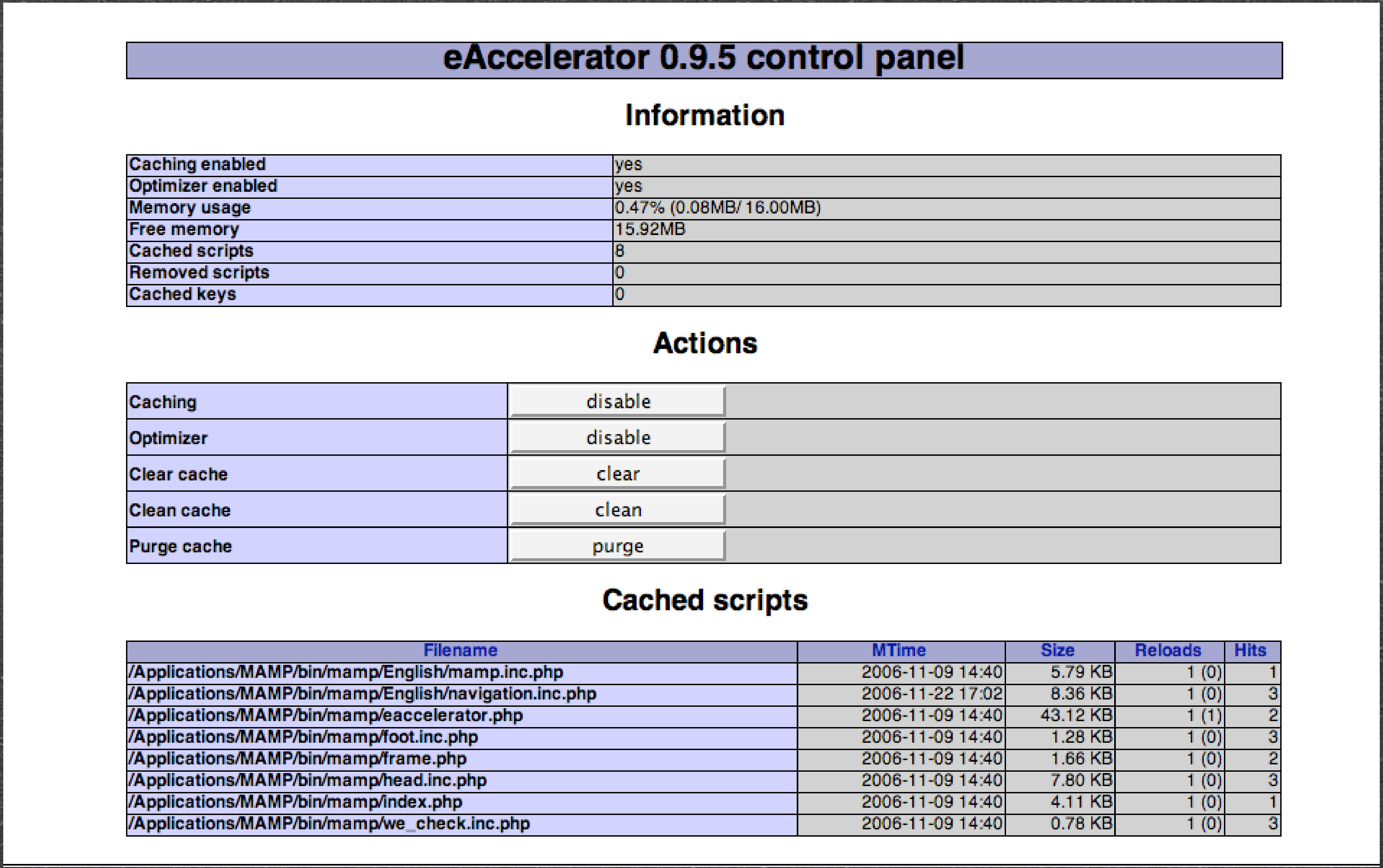
Agenda

- Overview of software performance enhancements.
- Overview of database performance
- Architectures for high performance database systems.
- High performance server farm, coloc farm, configurations and components.
- Big Metal systems
- Back down to the real world, some possible low cost solutions.

PHP with racing stripes

- Use a PHP accelerator
  - Zend Optimizer ($$$)
  - eAccelerator (free)
  - APC (free)
- Can make 30%-1000% difference in performance.
- Watch out with frameworks, some frameworks do not play well with accelerators
- Make sure that PHP gets plenty of memory.

eAccelerator



eAccelerator 0.9.5 control panel

Information

Caching enabled	yes
Optimizer enabled	yes
Memory usage	0.47% (0.08MB/ 16.00MB)
Free memory	15.92MB
Cached scripts	8
Removed scripts	0
Cached keys	0

Actions

Caching	disable
Optimizer	disable
Clear cache	clear
Clean cache	clean
Purge cache	purge

Cached scripts

Filename	MTime	Size	Reloads	Hits
/Applications/MAMP/bin/mamp/English/mamp.inc.php	2006-11-09 14:40	5.79 KB	1 (0)	1
/Applications/MAMP/bin/mamp/English/navigation.inc.php	2006-11-22 17:02	8.36 KB	1 (0)	3
/Applications/MAMP/bin/mamp/eaccelerator.php	2006-11-09 14:40	43.12 KB	1 (1)	2
/Applications/MAMP/bin/mamp/foot.inc.php	2006-11-09 14:40	1.28 KB	1 (0)	3
/Applications/MAMP/bin/mamp/frame.php	2006-11-09 14:40	1.66 KB	1 (0)	2
/Applications/MAMP/bin/mamp/head.inc.php	2006-11-09 14:40	7.80 KB	1 (0)	3
/Applications/MAMP/bin/mamp/index.php	2006-11-09 14:40	4.11 KB	1 (0)	1
/Applications/MAMP/bin/mamp/we_check.inc.php	2006-11-09 14:40	0.78 KB	1 (0)	3

Cache, Cache, Cache

- Two main ways of caching, in the app, or in the hardware.
- Many frameworks have built in caches, use them they make a big difference.
- Alternatively if you are building out your site, consider putting an inline cache/proxy on the front to cache outbound html. See web accelerators later.
- Cache in as many layers as you can, each one reduces hits to your expensive data back-end.
- Watch out for update latency in caching.

Mysql for speed freaks

- Make sure that you give your database server plenty of RAM, and tune the mysql caches to match.
- Make sure you have thought hard about your indices, and make sure all your queries are covered.
- Use the longrunningquries log file to show you which queries your system is struggling with.
- Avoid the use of esoteric sql functions and constructs, they often cause issues with replication. Keep it simple, also helps with maintence

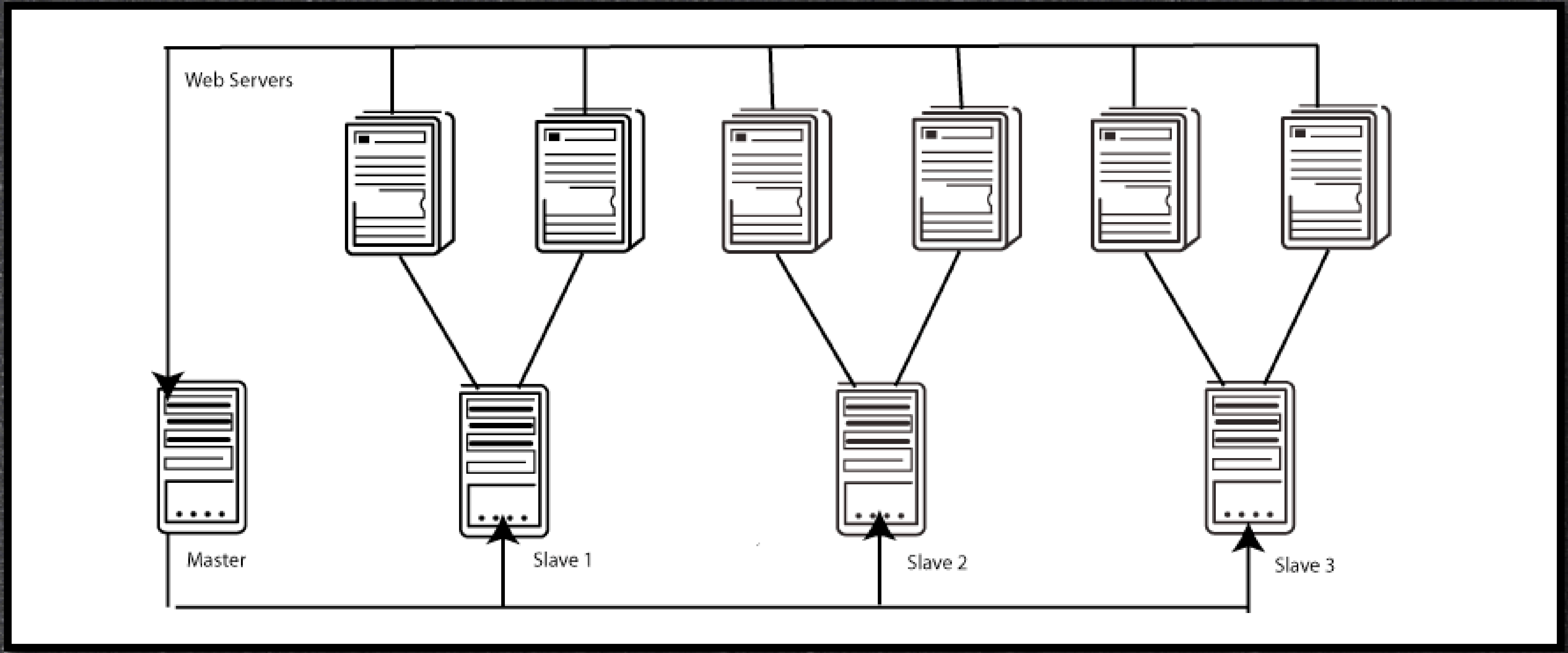
External Indexers

- Use an external indexer for search functions
- Nutch (Java)
- Lucene (Java)
- cLucene (“C”, PHP Ruby etc)
- htDig (“C”, PHP)
- mnogoSearch (“C”, PHP)
- Much higher performance on Free Text queries, do not use MySQL for search functions.

Break it up

- You will have probably started off with Apache and MySQL co-hosted on the same machine.
- If possible use separate web and database servers so you do not have I/O bottlenecking, and you can tune each environment to match.

Write Master, Read Slaves



- Single master write server, multiple read slaves.
- Suitable for systems with high read/write ratio.
- Suitable for up to 6-7 slaves, beyond that replications traffic gets flakey.
- Can use twin masters for redundancy.

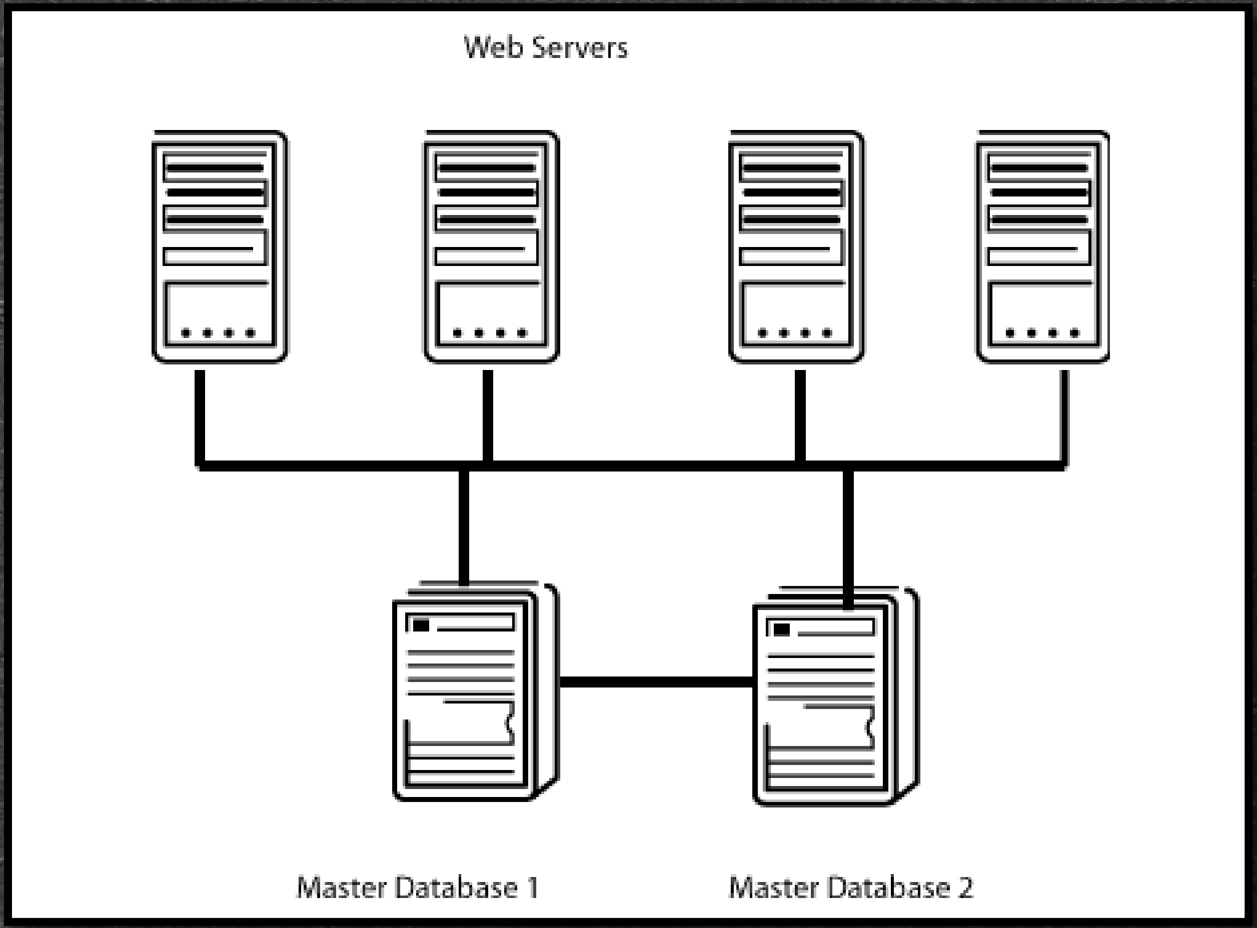
Master-Master Clustering

- Master Master - clusters support replication between two machines that can both accept writes and replicate to each other.

- However there are restrictions, all keys must be unique across both systems not just the one. Auto-increment values are problematic.

- Not practical to operate more than two masters, problems mount quickly after that.

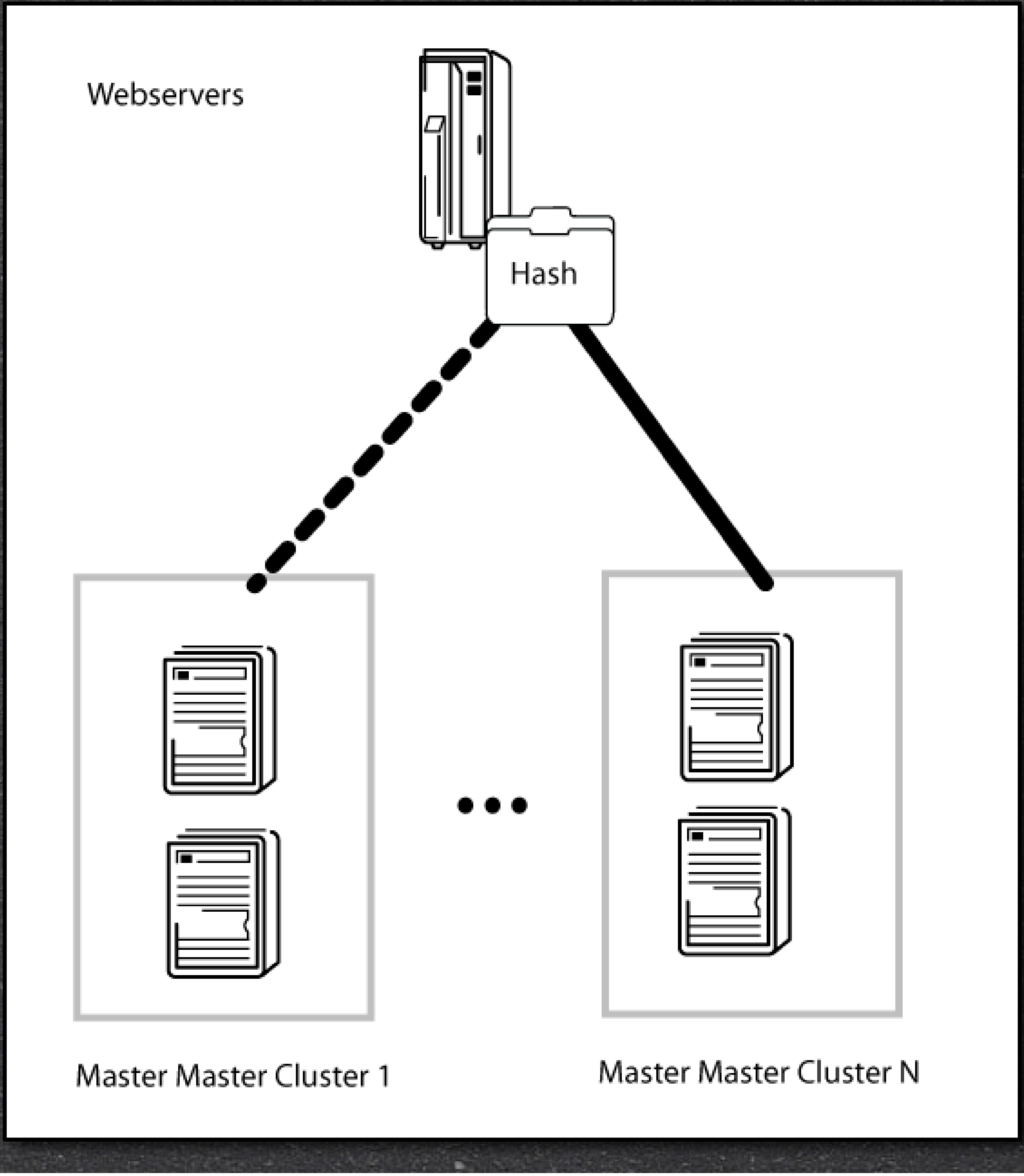
- Not all SQL syntax is available in this mode.



Join the federation

STAR TREK

- Create multiple clusters of multi-master replication pairs.
- Use a hashing function on primary key to bucket reads/writes to a particular cluster.
- No limit to the number of clusters, depends on the hashing function.
- Known as a federated array, hash function is the “federation function”.
- Federation is implemented in the data access layer of web app



Load balancing

- Multiple techniques can be used to distribute requests amongst a group or “farm” of web servers.
- Important tool for scaling systems beyond the performance of a single server.
- Can operate across different co-locs to manage farms of co-locs, each containing farms of web servers.
- Costs vary depending on approach.
- Software or hardware based systems are available.

Lets look at some methods.

Simple DNS load balancing

Assign multiple ‘A’ Names to a single dns name.

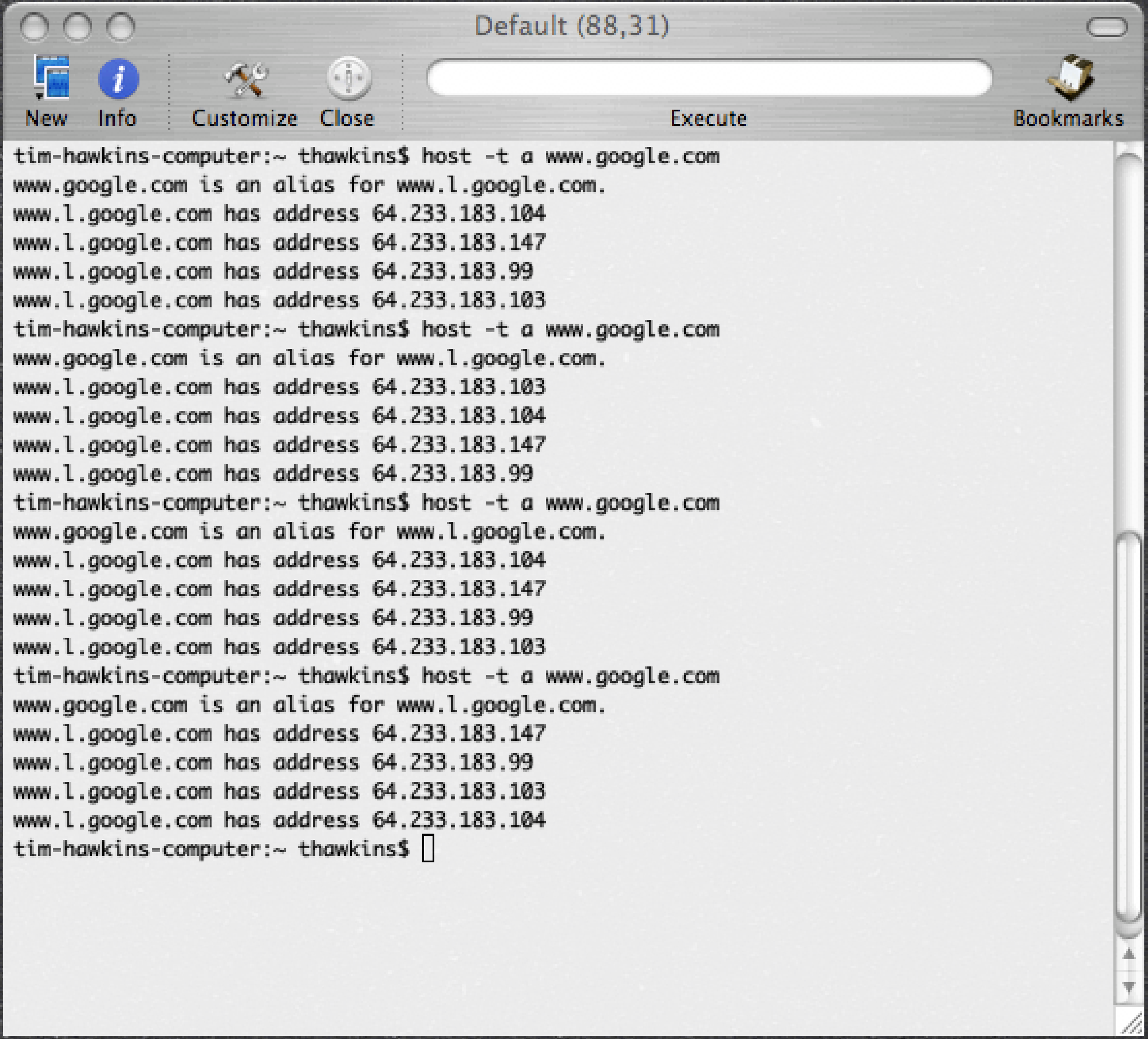
DNS randomizes the order of the returns on each call

Simple load distribution.

Caution: will direct traffic to dead servers

Bad: DNS changes to add/remove servers can take a long time to propagate.

Good: Dirt Cheap.



```
tim-hawkins-computer:~ thawkins$ host -t a www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 64.233.183.104
www.l.google.com has address 64.233.183.147
www.l.google.com has address 64.233.183.99
www.l.google.com has address 64.233.183.103
tim-hawkins-computer:~ thawkins$ host -t a www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 64.233.183.103
www.l.google.com has address 64.233.183.104
www.l.google.com has address 64.233.183.147
www.l.google.com has address 64.233.183.99
tim-hawkins-computer:~ thawkins$ host -t a www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 64.233.183.104
www.l.google.com has address 64.233.183.147
www.l.google.com has address 64.233.183.99
www.l.google.com has address 64.233.183.103
tim-hawkins-computer:~ thawkins$ host -t a www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 64.233.183.147
www.l.google.com has address 64.233.183.99
www.l.google.com has address 64.233.183.103
www.l.google.com has address 64.233.183.104
tim-hawkins-computer:~ thawkins$ 
```

Redirection Load Balancing

Uses a “controller” to redirect requests from site root domain to individual named servers.

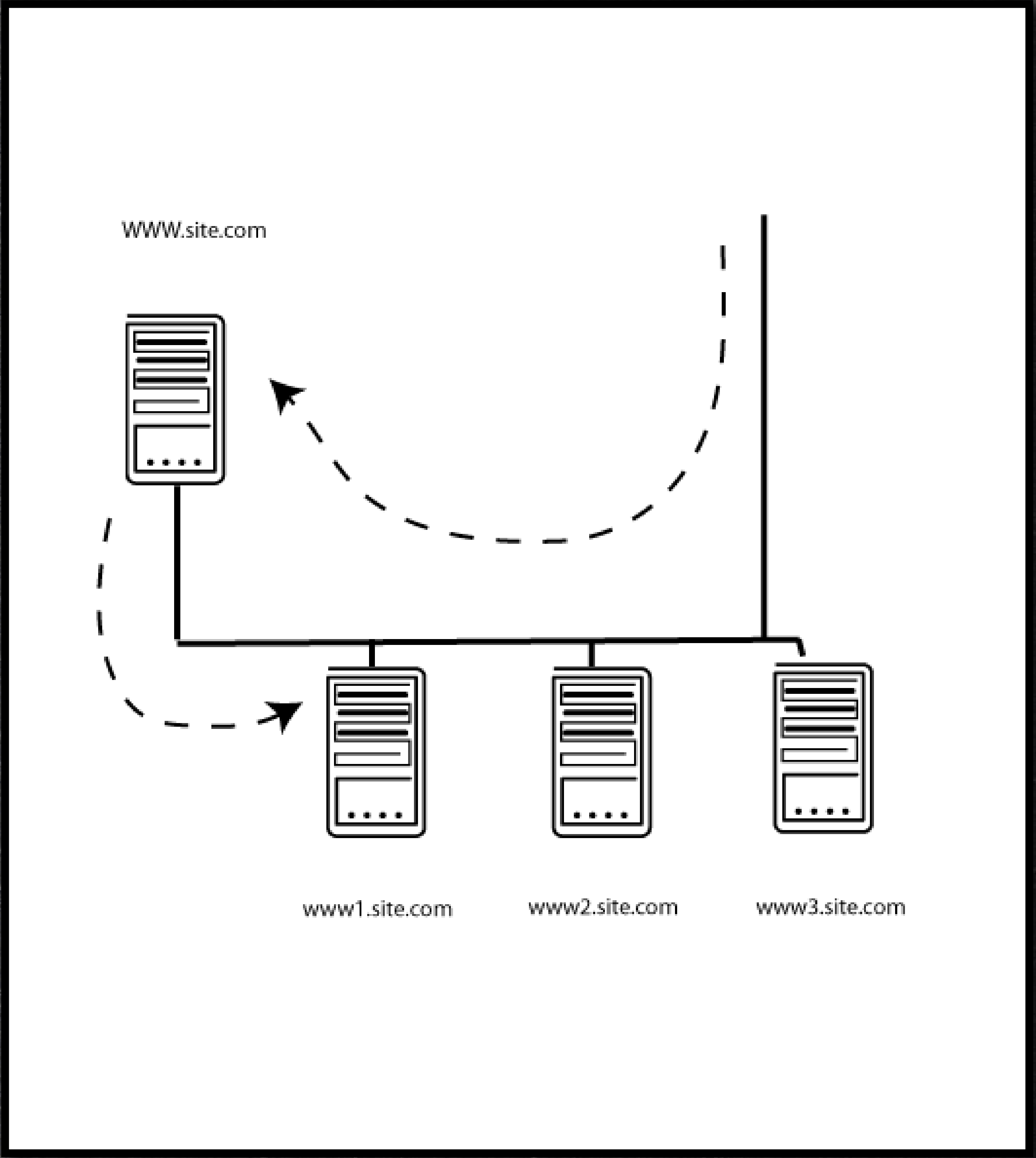
Fairly simple to implement.

Can be a single point of failure in the controller.

Naturally forces all requests for a user session to one server.

Gets used a lot on Java App servers.

Can be cheap as all web servers can also serve as controllers.



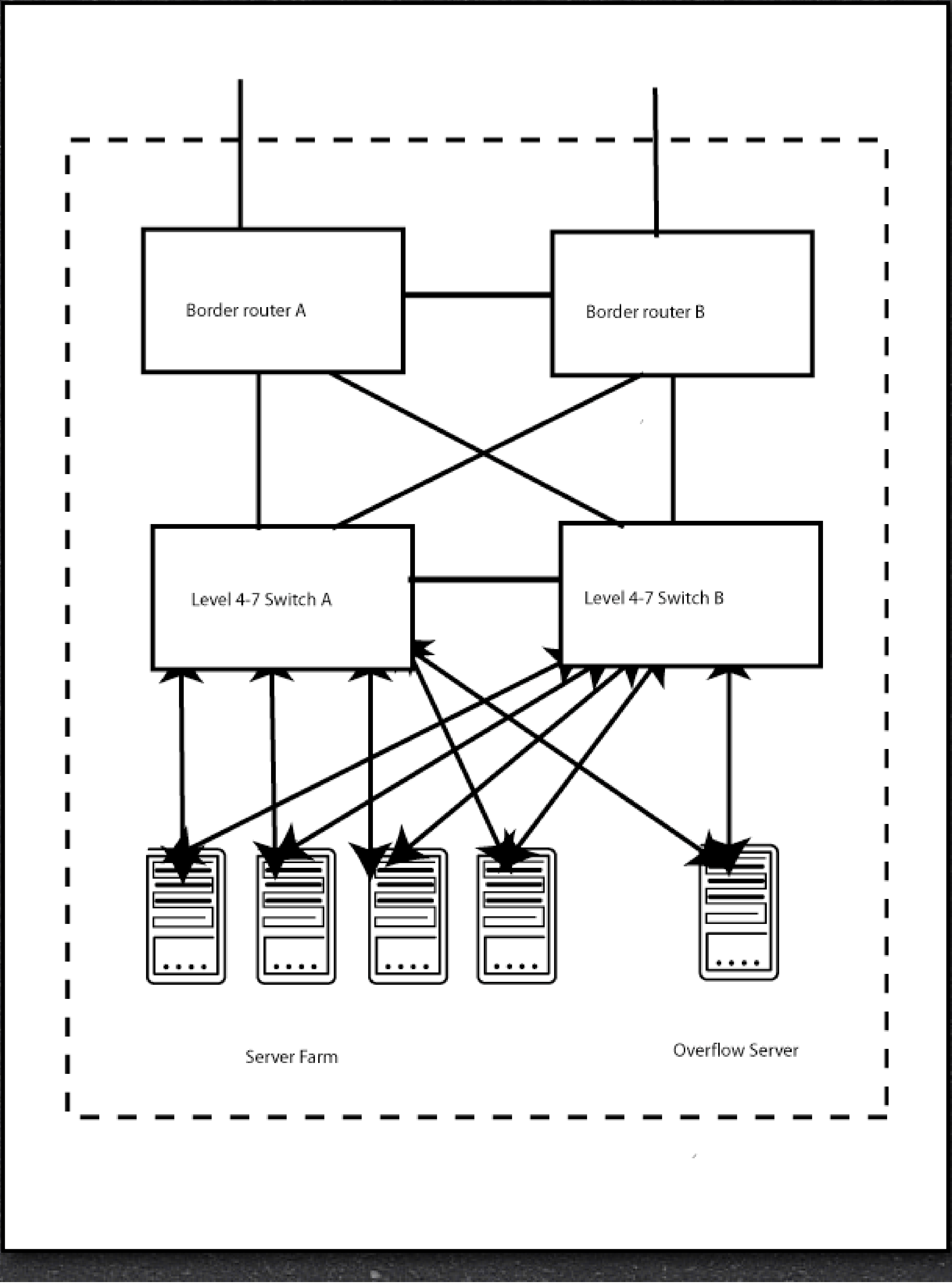
Layer 4-7 Switch balancers

Distributes requests to multiple web servers.

Monitors health of servers and drops out dead servers automatically.

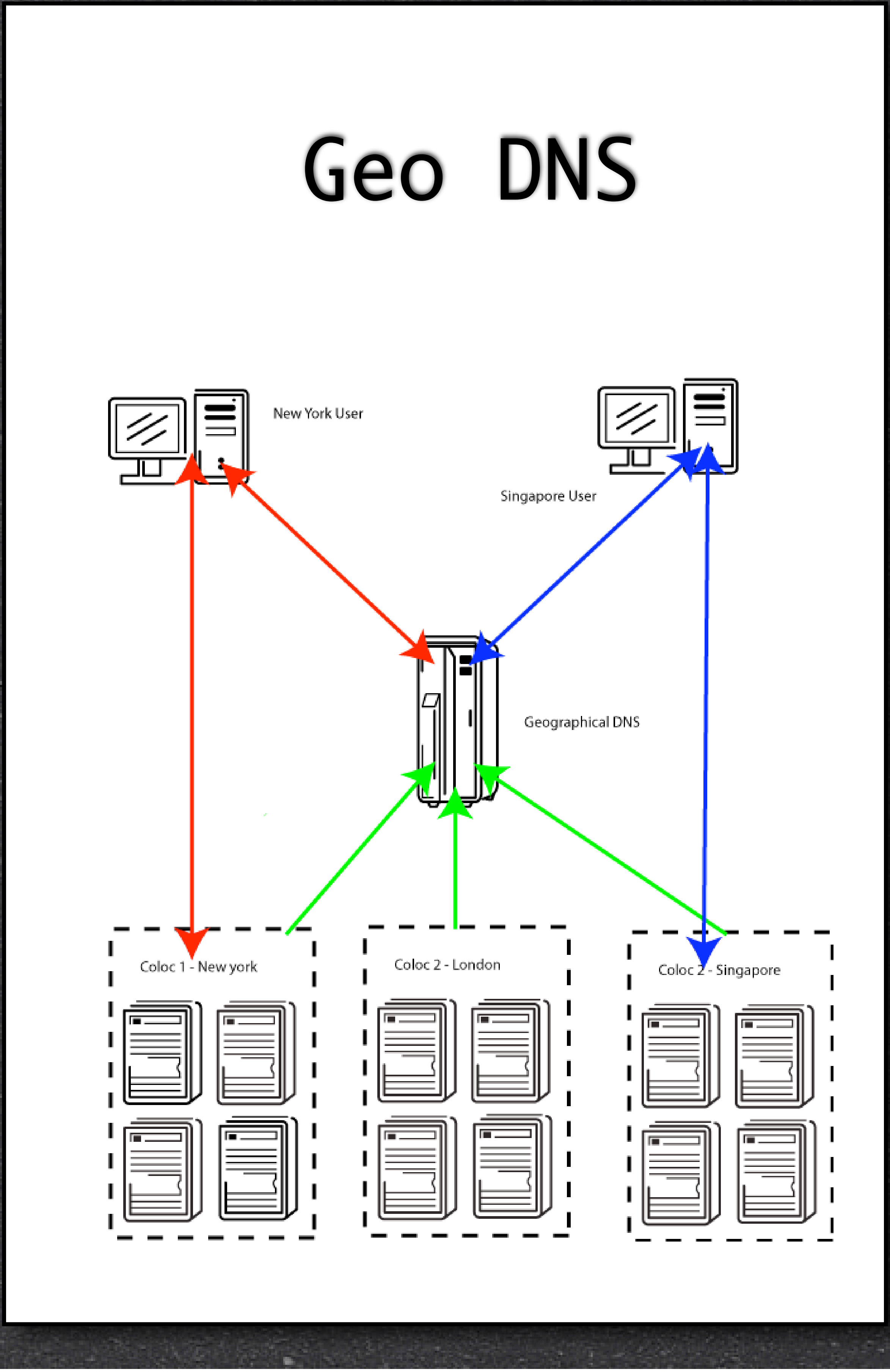
Can distribute excess connections to static overflow server.

Supports multiple distributions algos, (round robin, least connections, sticky session).



Geographical Load Balancing

- Used to balance traffic between multiple colocs.
- Routes user to closest network presence based on latency time between client and server.
- Can be used to allow automatic fail-over between different colocs.



Akamai load balancing

- Akamai provides two main services
  - ➡ Geographical DNS routing, aka geographical load balancing.
  - ➡ Asset caching and distribution (Edge network).
- Expensive solution.
- Can support redundant colocs.
- Monitors status of servers and routes to nearest/fastest.

Web I/O Accelerators

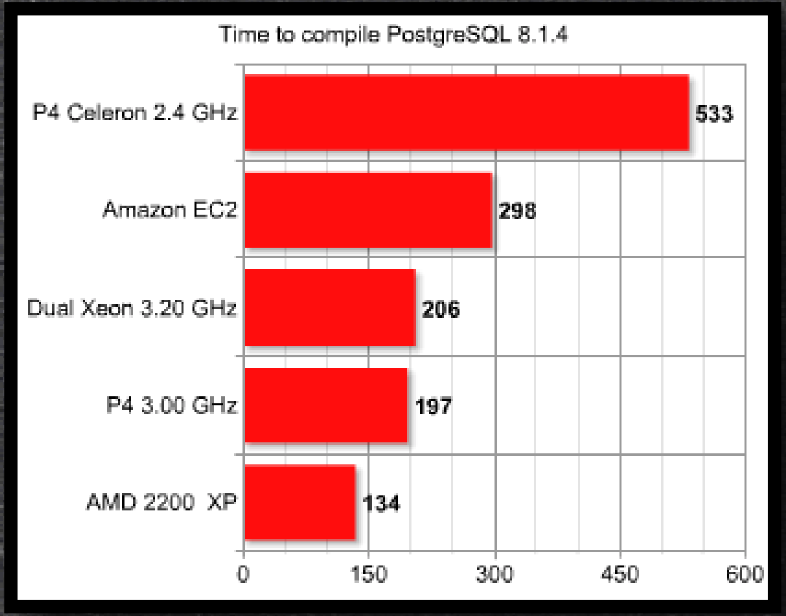
- Web I/O accelerators are a hybrid system consisting of a Layer 4-7 switch balancer, page I/O cache, and encryption/compression processor.
- Offsets the responsibility for result page conditioning and transmission to an external appliance.
- Sits in-front of a farm and handles all I/O.
- Pipelined fast I/O to web-servers.
- Can dramatical increase I/O performance.

Go Stateless

- Try where possible to make your APP stateless.
- State, or sessions cause headaches for the load balancer.
- Some environments (like java sessions) require the whole user session to be directed to the same web-server. Locks users into machines.
- State-full systems often result in slow back-end session storage requests, or large front-end memory overhead.
- In general “State is the enemy of performance”.

Powering up the grid - EC2

- Amazon EC2, provides grid computing capabilities at relatively low cost.
- Upload your own linux images for use as instances.
- Create large farms without using physical hardware.
- You can add or remove nodes to scale up or down.
- You can even script it to activate/ deactivate node dependent on load, true virtual hardware, node management API available.
- Get big site capabilities for relatively little cost.



Costs:

$0.10 per instance-hour consumed (or part of an hour consumed).
$0.20 per GB of data transferred into/out of Amazon (i.e., Internet traffic).
$0.15 per GB-Month of Amazon S3 storage used for your images (charged by Amazon S3).

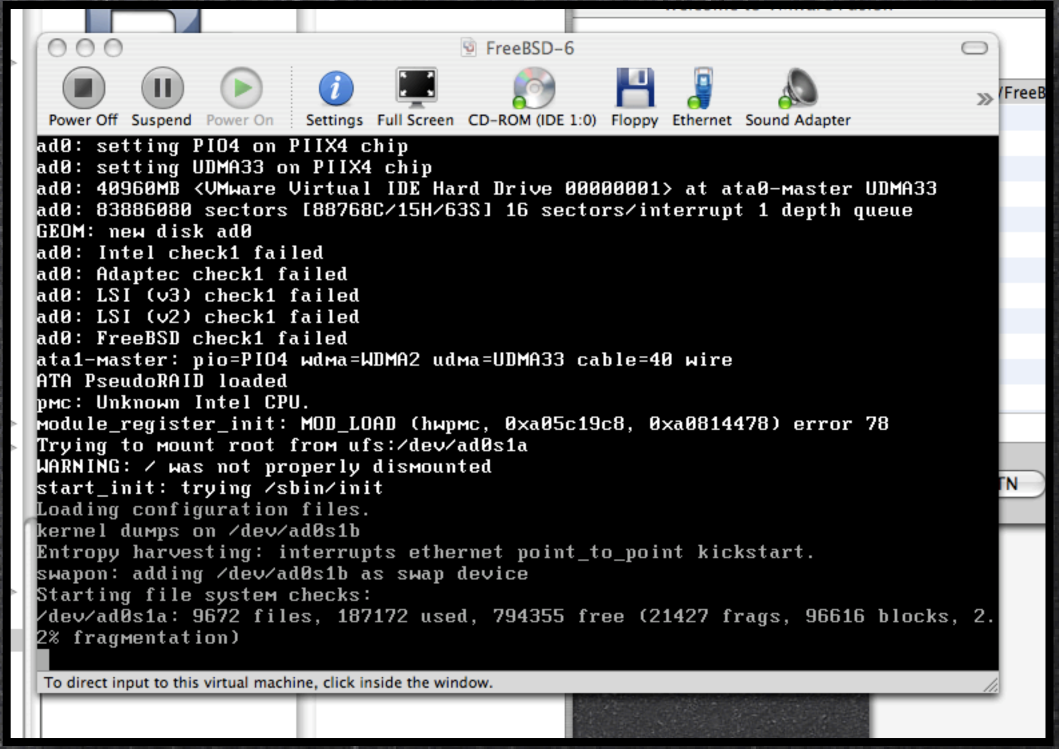
Prototype with VM’s

You do’nt even need hardware to prototype.

Modern VM systems like Q, Boches, Parallels and Vmware allow you to build complete systems inside a development machine.

Create a multi-node system, ready to upload to a Grid system like EC2.

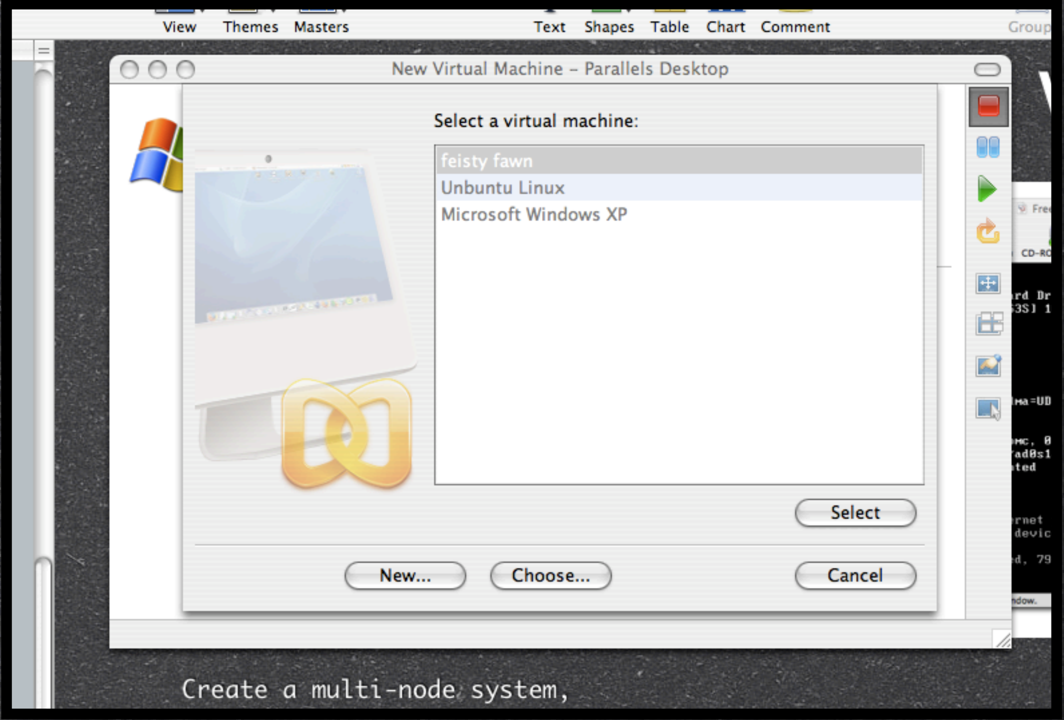
You can version control images with SVN or CVS, to manage roll-back etc.



FreeBSD-6

```
ad0: setting PIO4 on PIIX4 chip
ad0: setting UDMA33 on PIIX4 chip
ad0: 40960MB <VMware Virtual IDE Hard Drive 00000001> at ata0-master UDMA33
ad0: 83886080 sectors [88768C/15H/63S] 16 sectors/interrupt 1 depth queue
GEOM: new disk ad0
ad0: Intel check1 failed
ad0: Adaptec check1 failed
ad0: LSI (v3) check1 failed
ad0: LSI (v2) check1 failed
ad0: FreeBSD check1 failed
ata1-master: pio=PIO4 wdma=WDMA2 udma=UDMA33 cable=40 wire
ATA PseudoRAID loaded
pmc: Unknown Intel CPU.
module_register_init: MOD_LOAD (hwpmc, 0xa05c19c8, 0xa0814478) error 78
Trying to mount root from ufs:/dev/ad0s1a
WARNING: / was not properly dismounted
start_init: trying /sbin/init
Loading configuration files.
kernel dumps on /dev/ad0s1b
Entropy harvesting: interrupts ethernet point_to_point kickstart.
swapon: adding /dev/ad0s1b as swap device
Starting file system checks:
/dev/ad0s1a: 9672 files, 187172 used, 794355 free (21427 frags, 96616 blocks, 2.
2% fragmentation)
```

To direct input to this virtual machine, click inside the window.



New Virtual Machine – Parallels Desktop

Select a virtual machine:

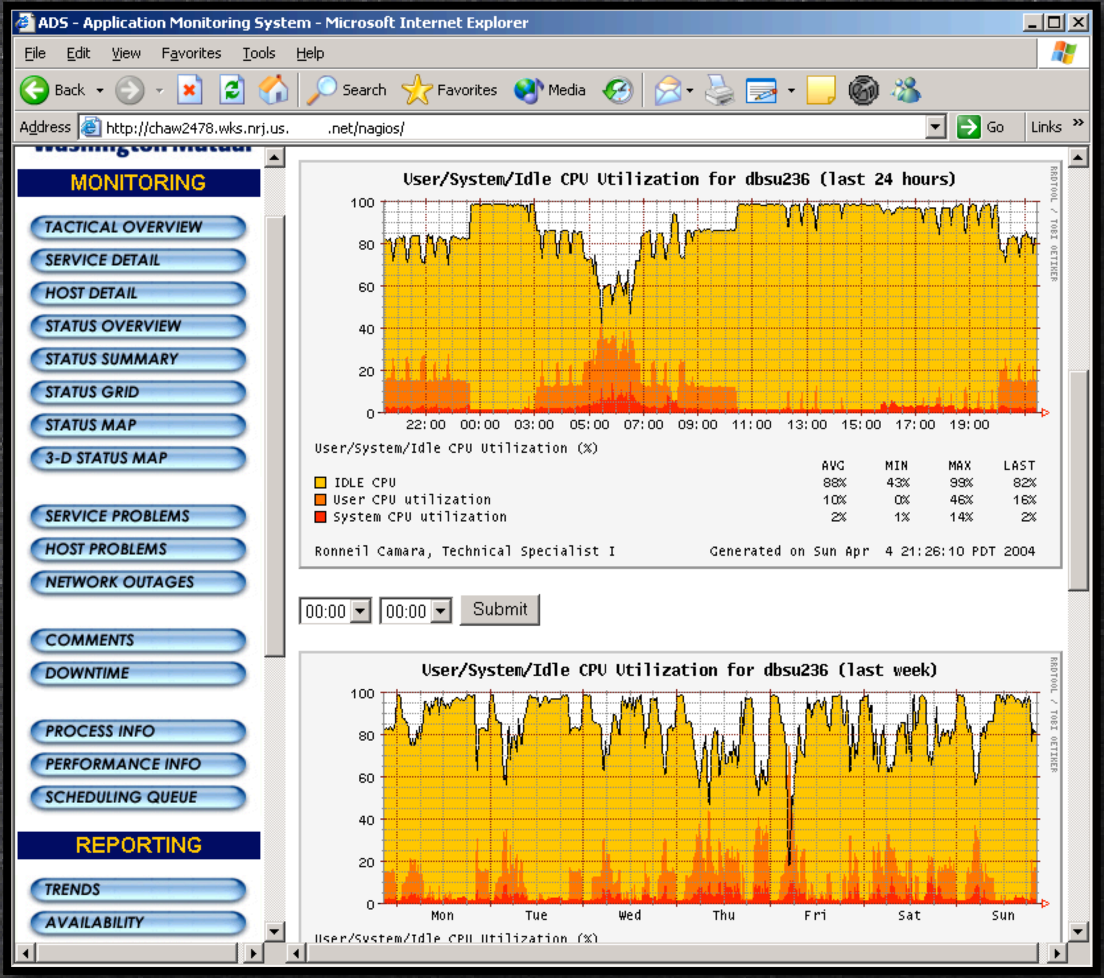
feisty fawn
Unbuntu Linux
Microsoft Windows XP

Monitoring Systems

- Basically two types of monitoring systems.
    - ➡ Availability monitoring.
    - ➡ Performance monitoring.
- Some systems can do both.
- Systems can have programed trip limits, and either alert, or execute resolution scripts.
- Try to get a system that supports persistent loging and baselining.

Nagios

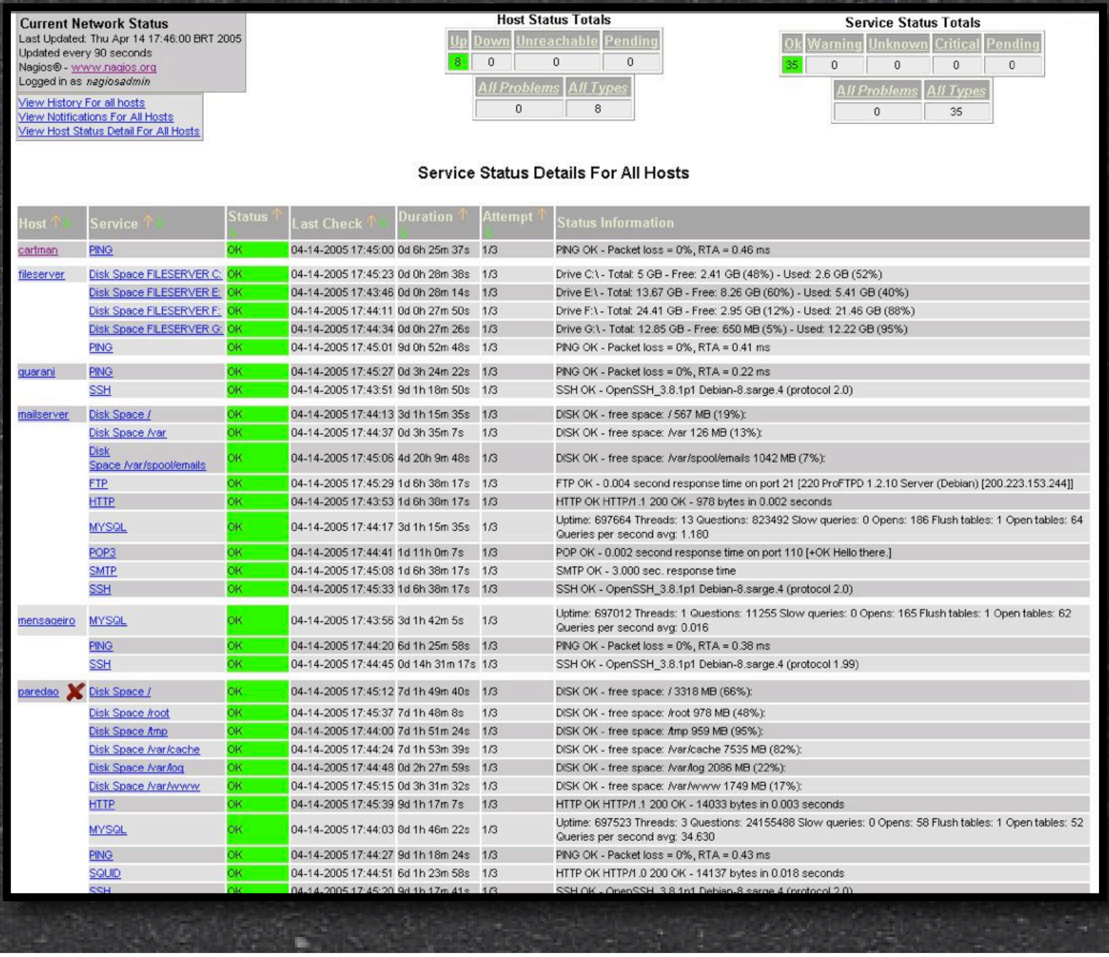
- Open source monitoring system.
- Supports custom monitoring scripts.
- Supports automatic fault resolution scripts.
- Has logging and baseline facilities.
- Performs status alerting via email or scripts.



User/System/Idle CPU Utilization (%)

	AVG	MIN	MAX	LAST
🟨 IDLE CPU	88%	43%	99%	82%
🟧 User CPU utilization	10%	0%	46%	16%
🟥 System CPU utilization	2%	1%	14%	2%

Ronneil Camara, Technical Specialist I   Generated on Sun Apr 4 21:26:10 PDT 2004


Current Network Status
Last Updated: Thu Apr 14 17:46:00 BRT 2005
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
8	0	0	0

All Problems	All Types
0	8

Service Status Totals

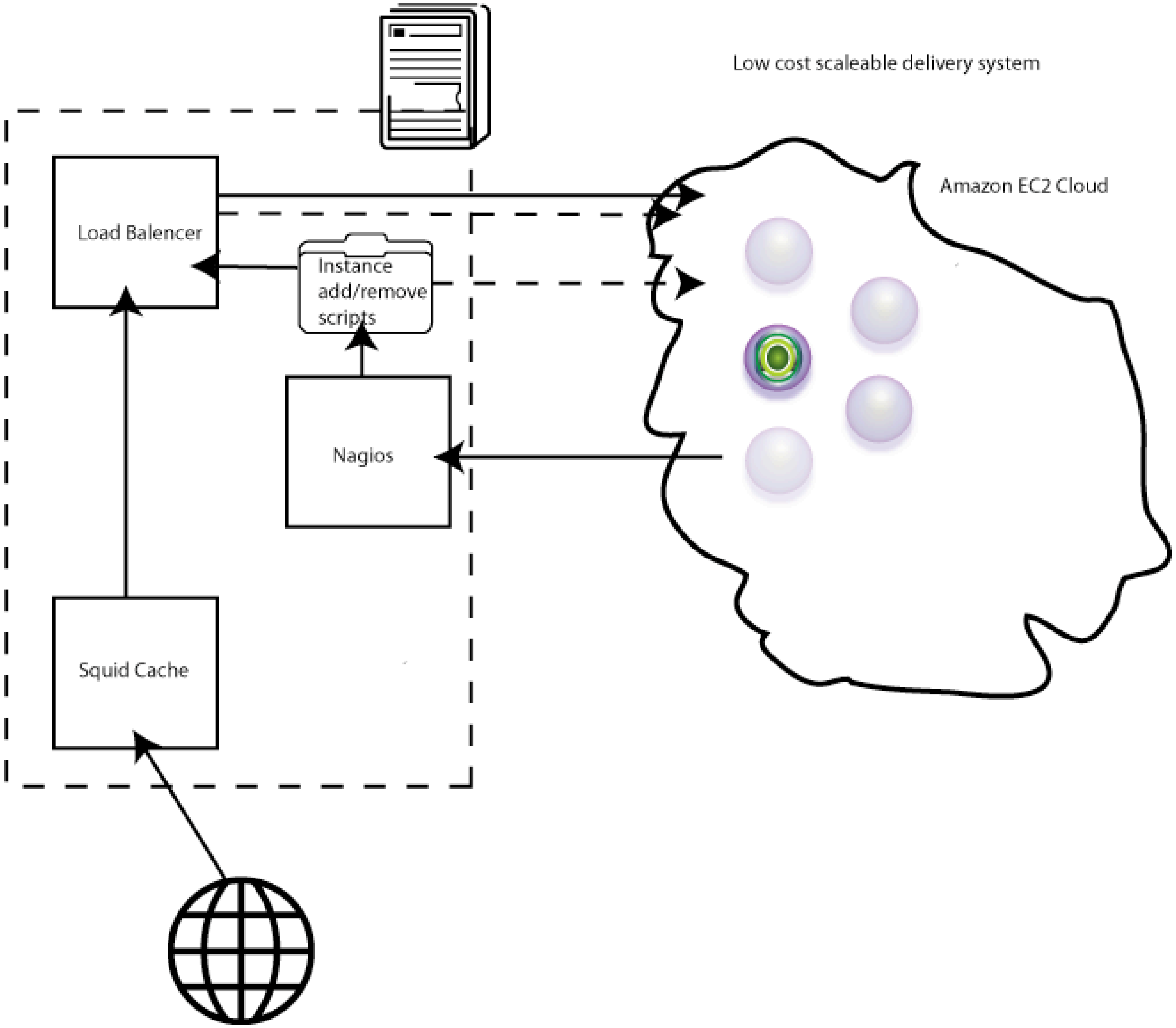
Ok	Warning	Unknown	Critical	Pending
35	0	0	0	0

All Problems	All Types
0	35

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
cartman	PING	OK	04-14-2005 17:45:00	0d 6h 25m 37s	1/3	PING OK - Packet loss = 0%, RTA = 0.46 ms
fileserver	Disk Space FILESERVER C:	OK	04-14-2005 17:45:23	0d 0h 28m 38s	1/3	Drive C:\ - Total: 5 GB - Free: 2.41 GB (48%) - Used: 2.6 GB (52%)
	Disk Space FILESERVER E:	OK	04-14-2005 17:43:46	0d 0h 28m 14s	1/3	Drive E:\ - Total: 13.67 GB - Free: 8.26 GB (60%) - Used: 5.41 GB (40%)
	Disk Space FILESERVER F:	OK	04-14-2005 17:44:11	0d 0h 27m 50s	1/3	Drive F:\ - Total: 24.41 GB - Free: 2.95 GB (12%) - Used: 21.46 GB (88%)
	Disk Space FILESERVER G:	OK	04-14-2005 17:44:34	0d 0h 27m 26s	1/3	Drive G:\ - Total: 12.85 GB - Free: 650 MB (5%) - Used: 12.22 GB (95%)
	PING	OK	04-14-2005 17:45:01	9d 0h 52m 48s	1/3	PING OK - Packet loss = 0%, RTA = 0.41 ms
guarani	PING	OK	04-14-2005 17:45:27	0d 3h 24m 22s	1/3	PING OK - Packet loss = 0%, RTA = 0.22 ms
	SSH	OK	04-14-2005 17:43:51	9d 1h 18m 50s	1/3	SSH OK - OpenSSH_3.8.1p1 Debian-8.sarge.4 (protocol 2.0)
mailserver	Disk Space /	OK	04-14-2005 17:44:13	3d 1h 15m 35s	1/3	DISK OK - free space: / 567 MB (19%):
	Disk Space /var	OK	04-14-2005 17:44:37	0d 3h 35m 7s	1/3	DISK OK - free space: /var 126 MB (13%):
	Disk Space /var/spool/emails	OK	04-14-2005 17:45:06	4d 20h 9m 48s	1/3	DISK OK - free space: /var/spool/emails 1042 MB (7%):
	FTP	OK	04-14-2005 17:45:29	1d 6h 38m 17s	1/3	FTP OK - 0.004 second response time on port 21 [220 ProFTPD 1.2.10 Server (Debian) [200.223.153.244]]
	HTTP	OK	04-14-2005 17:43:53	1d 6h 38m 17s	1/3	HTTP OK HTTP/1.1 200 OK - 978 bytes in 0.002 seconds
	MYSQL	OK	04-14-2005 17:44:17	3d 1h 15m 35s	1/3	Uptime: 697664 Threads: 13 Questions: 823492 Slow queries: 0 Opens: 186 Flush tables: 1 Open tables: 64 Queries per second avg: 1.180
	POP3	OK	04-14-2005 17:44:41	1d 11h 0m 7s	1/3	POP OK - 0.002 second response time on port 110 [+OK Hello there.]
	SMTP	OK	04-14-2005 17:45:08	1d 6h 38m 17s	1/3	SMTP OK - 3.000 sec. response time
	SSH	OK	04-14-2005 17:45:33	1d 6h 38m 17s	1/3	SSH OK - OpenSSH_3.8.1p1 Debian-8.sarge.4 (protocol 2.0)
mensageiro	MYSQL	OK	04-14-2005 17:43:56	3d 1h 42m 5s	1/3	Uptime: 697012 Threads: 1 Questions: 11255 Slow queries: 0 Opens: 165 Flush tables: 1 Open tables: 62 Queries per second avg: 0.016
	PING	OK	04-14-2005 17:44:20	6d 1h 25m 58s	1/3	PING OK - Packet loss = 0%, RTA = 0.38 ms
	SSH	OK	04-14-2005 17:44:45	0d 14h 31m 17s	1/3	SSH OK - OpenSSH_3.8.1p1 Debian-8.sarge.4 (protocol 1.99)
paredao ✗	Disk Space /	OK	04-14-2005 17:45:12	7d 1h 49m 40s	1/3	DISK OK - free space: / 3318 MB (66%):
	Disk Space /root	OK	04-14-2005 17:45:37	7d 1h 48m 8s	1/3	DISK OK - free space: /root 978 MB (48%):
	Disk Space /tmp	OK	04-14-2005 17:44:00	7d 1h 51m 24s	1/3	DISK OK - free space: /tmp 959 MB (95%):
	Disk Space /var/cache	OK	04-14-2005 17:44:24	7d 1h 53m 39s	1/3	DISK OK - free space: /var/cache 7535 MB (82%):
	Disk Space /var/log	OK	04-14-2005 17:44:48	0d 2h 27m 59s	1/3	DISK OK - free space: /var/log 2086 MB (22%):
	Disk Space /var/www	OK	04-14-2005 17:45:15	0d 3h 31m 32s	1/3	DISK OK - free space: /var/www 1749 MB (17%):
	HTTP	OK	04-14-2005 17:45:39	9d 1h 17m 7s	1/3	HTTP OK HTTP/1.1 200 OK - 14033 bytes in 0.003 seconds
	MYSQL	OK	04-14-2005 17:44:03	8d 1h 46m 22s	1/3	Uptime: 697523 Threads: 3 Questions: 24155488 Slow queries: 0 Opens: 58 Flush tables: 1 Open tables: 52 Queries per second avg: 34.630
	PING	OK	04-14-2005 17:44:27	9d 1h 18m 24s	1/3	PING OK - Packet loss = 0%, RTA = 0.43 ms
	SQUID	OK	04-14-2005 17:44:51	6d 1h 23m 58s	1/3	HTTP OK HTTP/1.0 200 OK - 14137 bytes in 0.018 seconds
	SSH	OK	04-14-2005 17:45:20	9d 1h 17m 41s	1/3	SSH OK - OpenSSH_3.8.1p1 Debian-8.sarge.4 (protocol 2.0)

Concept: A Dynamic Scaleable architecture for Startups


Low cost scaleable delivery system
Amazon EC2 Cloud
Load Balencer
Instance add/remove scripts
Nagios
Squid Cache

As load increases and trips thresholds in Nagios monitor, action scripts add dynamic nodes to EC2 farm to increase capacity, linking them into the load balancer. As load decreases, it will automatically remove nodes from the farm.

You get Big Farm load resilience for peanuts, you only pay for what you use.

Put it all together

- Use PHP acceleration techniques to optimize your code performance.
- Tune your mysql, config and queries. Often the biggest bottleneck.
- Use multi-node systems if possible.
- Use the right architecture.
- Load balance your system correctly.
- Monitor, Monitor, Monitor.

Questions?

Thank you for your Time.

thawkins@yahoo-inc.com