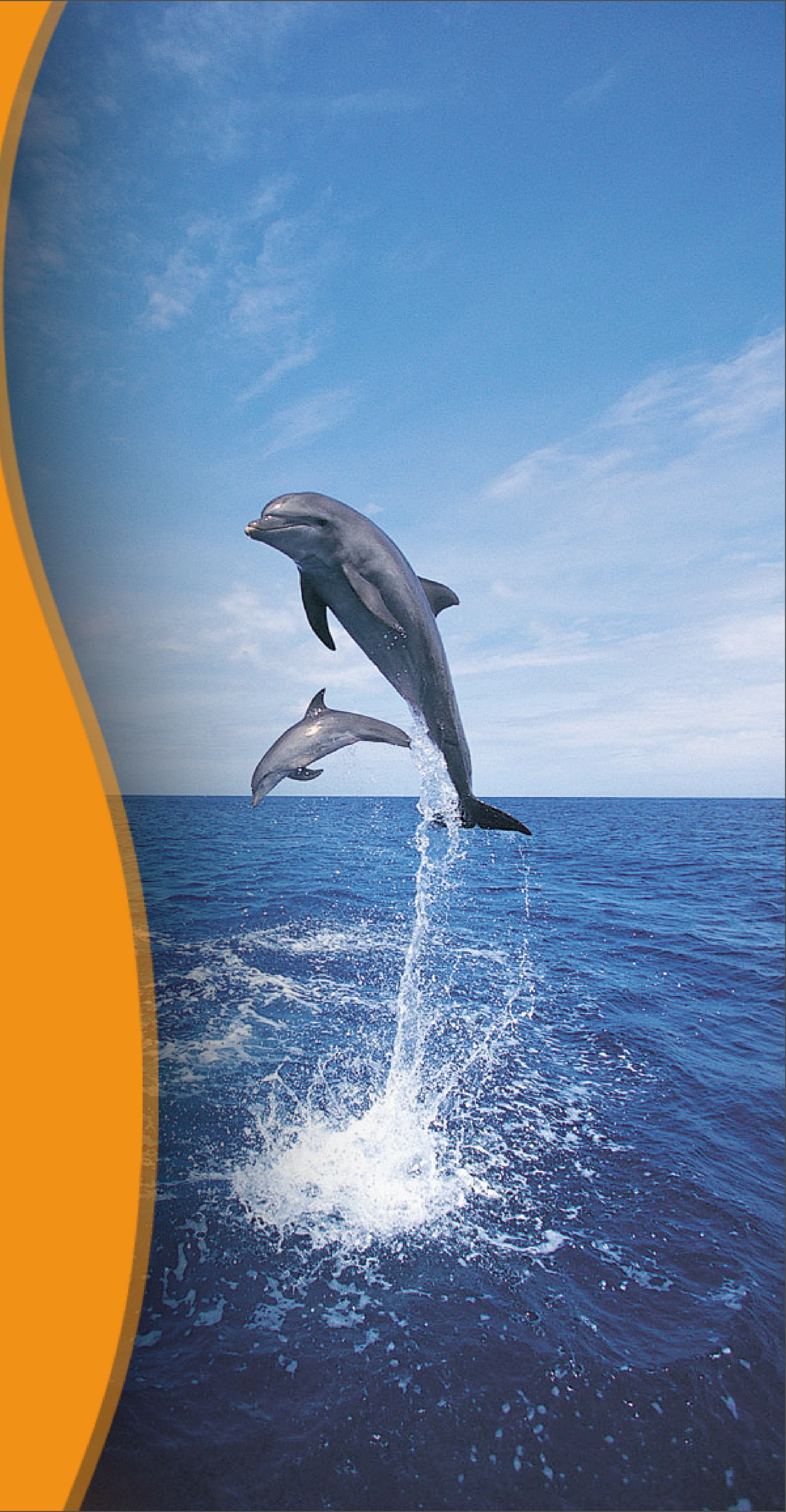
Writing a social application in PHP/ MySQL

and what happens when a million people show up on opening day



Duleepa "Dups" Wijayawardhana
MySQL Community Team

Who the hell am I?

Who the hell am I?

- PHP/MySQL Developer since 1999/2000

Who the hell am I?

- PHP/MySQL Developer since 1999/2000
- MySQL Community Relations Manager in North America since... oh... last week :)

Who the hell am I?

- PHP/MySQL Developer since 1999/2000
- MySQL Community Relations Manager in North America since... oh... last week :)
- Web Developer for MySQL

Who the hell am I?

- PHP/MySQL Developer since 1999/2000
- MySQL Community Relations Manager in North America since... oh... last week :)
- Web Developer for MySQL
- Various positions at BioWare Corp. 2001-2007

Who the hell am I?

- PHP/MySQL Developer since 1999/2000
- MySQL Community Relations Manager in North America since... oh... last week :)
- Web Developer for MySQL
- Various positions at BioWare Corp. 2001-2007
- Also run the Annual St. Patricks Day Drunk Dial (http://www.stpatsdrunkdial.com)

What makes me qualified?

What makes me qualified?

- Spent a lot of time screaming at Murphy.

What makes me qualified?

- Spent a lot of time screaming at Murphy.
- Hired by BioWare to help create the BioWare Community Site for Neverwinter Nights (launched June 2002)

What makes me qualified?

- Spent a lot of time screaming at Murphy.
- Hired by BioWare to help create the BioWare Community Site for Neverwinter Nights (launched June 2002)
- Had no clue what to expect, but the rest as they say now is history.

What makes me qualified?

- Spent a lot of time screaming at Murphy.
- Hired by BioWare to help create the BioWare Community Site for Neverwinter Nights (launched June 2002)
- Had no clue what to expect, but the rest as they say now is history.
- Learned lots. Still learning...

What makes me qualified?

- Spent a lot of time screaming at Murphy.
- Hired by BioWare to help create the BioWare Community Site for Neverwinter Nights (launched June 2002)
- Had no clue what to expect, but the rest as they say now is history.
- Learned lots. Still learning...
- Insanity...

What are we going to cover?

What are we going to cover?

- A History of Disaster

What are we going to cover?

- A History of Disaster
- Launching the BioWare Community

What are we going to cover?

- A History of Disaster
- Launching the BioWare Community
- Pain Points of an Application

A History of Disaster

A History of Disaster

- What skill must you have to be a web developer/sysadmin in charge of a major web application?

A History of Disaster

- What skill must you have to be a web developer/sysadmin in charge of a major web application?
- The answer: "You must be slightly insane!"

A History of Disaster

- What skill must you have to be a web developer/sysadmin in charge of a major web application?
- The answer: "You must be slightly insane!"
- ... and perhaps slightly masochistic :)

A History of Disaster

A History of Disaster

- Web Server Crashes (typical)

A History of Disaster

- Web Server Crashes (typical)
- Load Balancer Crashes

A History of Disaster

- Web Server Crashes (typical)
- Load Balancer Crashes
- Major Mail System crashes

A History of Disaster

- Web Server Crashes (typical)
- Load Balancer Crashes
- Major Mail System crashes
- File System Crashes (or how I came to hate NFS)

A History of Disaster

- Web Server Crashes (typical)
- Load Balancer Crashes
- Major Mail System crashes
- File System Crashes (or how I came to hate NFS)
- Database Crashes (sigh)

A History of Disaster

- Web Server Crashes (typical)
- Load Balancer Crashes
- Major Mail System crashes
- File System Crashes (or how I came to hate NFS)
- Database Crashes (sigh)
- Power outage in the building

A History of Disaster

A History of Disaster

- Sweden has an incredibly hot summer. Who knew.

A History of Disaster

- Sweden has an incredibly hot summer. Who knew.
- Building nearby burns down and takes down the city grid, we sit on the balcony and watch the fire...

A History of Disaster

- Sweden has an incredibly hot summer. Who knew.
- Building nearby burns down and takes down the city grid, we sit on the balcony and watch the fire...
- The toilet explodes and floods.

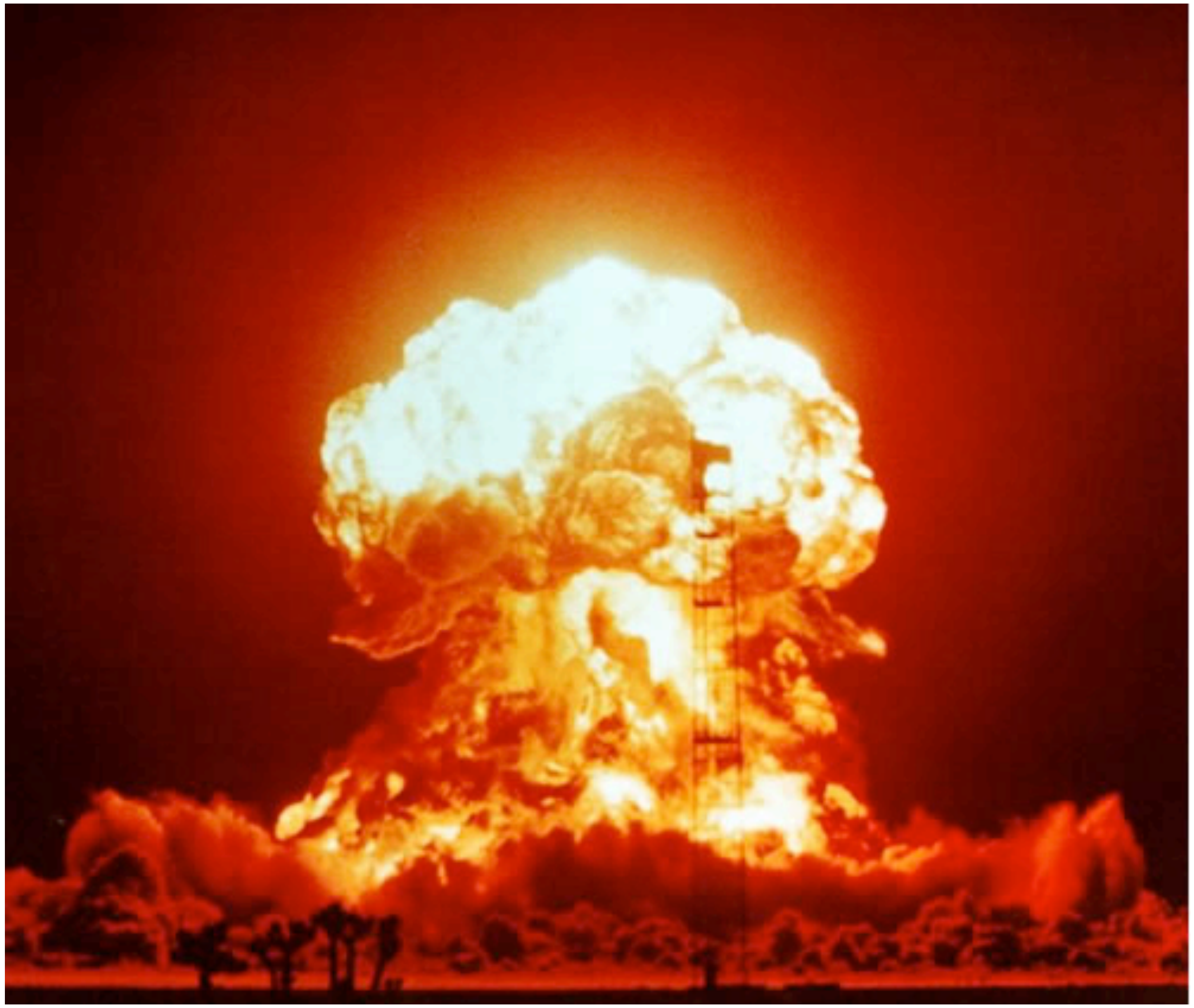
A History of Disaster

- Sweden has an incredibly hot summer. Who knew.
- Building nearby burns down and takes down the city grid, we sit on the balcony and watch the fire...
- The toilet explodes and floods.
- Someone connects the storm drain to the kitchen sink. Oops.

A History of Disaster

A History of Disaster

- Someone pours water on the electric mainboard and explodes your main electrical supply the day before a release. Big Oops.



A History of Disaster

A History of Disaster

- Imagine running through one amazingly crazy blizzard, drunk as you watch transformers explode and the sweeping cone of darkness spread across the city.... You have the presence of mind to do a sequenced shut down but you can't see straight to bring anything back up so you sleep on the server room floor to sober up.

What is a Social Application?

What is a Social Application?

- A site which primarily focuses on interactions between users.

What is a Social Application?

- A site which primarily focuses on interactions between users.
- MySQL.com is not a social application,

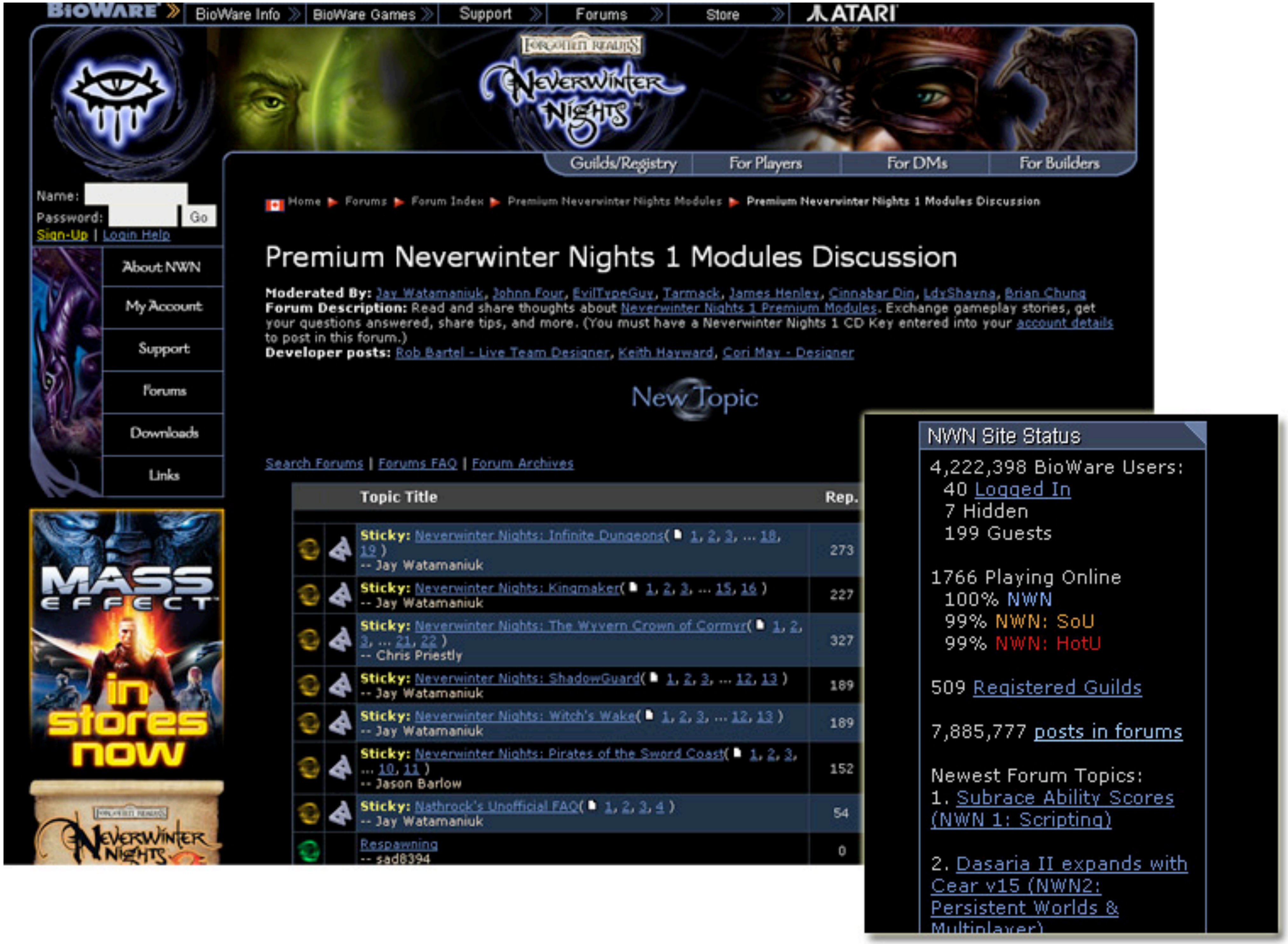
What is a Social Application?

- A site which primarily focuses on interactions between users.
- MySQL.com is not a social application,
- “Web 2.0” applications: Facebook, LinkedIn, MySpace, most community sites.

What is a Social Application?

- A site which primarily focuses on interactions between users.
- MySQL.com is not a social application,
- "Web 2.0" applications: Facebook, LinkedIn, MySpace, most community sites.
- Developing and Launching a social application has special "challenges"

Neverwinter Nights Community...



Home ▶ Forums ▶ Forum Index ▶ Premium Neverwinter Nights Modules ▶ Premium Neverwinter Nights 1 Modules Discussion

Premium Neverwinter Nights 1 Modules Discussion

Moderated By: Jay Watamaniuk, Johnn Four, EvilTypeGuy, Tarmack, James Henley, Cinnabar Din, LdyShayna, Brian Chung
Forum Description: Read and share thoughts about Neverwinter Nights 1 Premium Modules. Exchange gameplay stories, get your questions answered, share tips, and more. (You must have a Neverwinter Nights 1 CD Key entered into your account details to post in this forum.)
Developer posts: Rob Bartel - Live Team Designer, Keith Hayward, Cori May - Designer

New Topic

Search Forums | Forums FAQ | Forum Archives

Topic Title	Rep.
Sticky: Neverwinter Nights: Infinite Dungeons(1, 2, 3, ... 18, 19) -- Jay Watamaniuk	273
Sticky: Neverwinter Nights: Kingmaker(1, 2, 3, ... 15, 16) -- Jay Watamaniuk	227
Sticky: Neverwinter Nights: The Wyvern Crown of Cormyr(1, 2, 3, ... 21, 22) -- Chris Priestly	327
Sticky: Neverwinter Nights: ShadowGuard(1, 2, 3, ... 12, 13) -- Jay Watamaniuk	189
Sticky: Neverwinter Nights: Witch's Wake(1, 2, 3, ... 12, 13) -- Jay Watamaniuk	189
Sticky: Neverwinter Nights: Pirates of the Sword Coast(1, 2, 3, ... 10, 11) -- Jason Barlow	152
Sticky: Nathrock's Unofficial FAQ(1, 2, 3, 4) -- Jay Watamaniuk	54
Respawning -- sad8394	0

NWN Site Status

4,222,398 BioWare Users:
40 Logged In
7 Hidden
199 Guests

1766 Playing Online
100% NWN
99% NWN: SoU
99% NWN: HotU

509 Registered Guilds

7,885,777 posts in forums

Newest Forum Topics:
1. Subrace Ability Scores (NWN 1: Scripting)

2. Dasaria II expands with Cear v15 (NWN2: Persistent Worlds & Multiplayer)

A primer for launching a social application...

A primer for launching a social application...

1. Plan to have about 10 times the number of users that you conservatively expect.

A primer for launching a social application...

1. Plan to have about 10 times the number of users that you conservatively expect.
2. Be prepared for getting 100 times if your marketing has been good.

A primer for launching a social application...

1. Plan to have about 10 times the number of users that you conservatively expect.
2. Be prepared for getting 100 times if your marketing has been good.
3. Be prepared to scale *EVERY* aspect of your application: Web, DB, Mail etc.

A primer for launching a social application...

1. Plan to have about 10 times the number of users that you conservatively expect.
2. Be prepared for getting 100 times if your marketing has been good.
3. Be prepared to scale *EVERY* aspect of your application: Web, DB, Mail etc.
4. Be smart, launch softly.

A primer for launching a social application...

1. Plan to have about 10 times the number of users that you conservatively expect.
2. Be prepared for getting 100 times if your marketing has been good.
3. Be prepared to scale *EVERY* aspect of your application: Web, DB, Mail etc.
4. Be smart, launch softly.
5. Be even smarter, don't launch on a Friday evening.

Before the launch

- All cocky and sure of myself
- What could go wrong?

After the launch

- A picture is worth a thousand words

What we did (Don’t do at home!)

What we did (Don’t do at home!)

- Launched on a Friday afternoon, idea was to have less traffic.

What we did (Don’t do at home!)

- Launched on a Friday afternoon, idea was to have less traffic.
- Site contained a function to send an alert if database was down, when site went down, it triggered 5,000+ emails in a few mins and took down the mail server

What we did (Don’t do at home!)

- Launched on a Friday afternoon, idea was to have less traffic.
- Site contained a function to send an alert if database was down, when site went down, it triggered 5,000+ emails in a few mins and took down the mail server
- Not enough slaves to allow the site to function. Ripped apart desktop computers to create functional DB slaves.

Key to Success

Key to Success

- Figure out the “pain” points of an application

Key to Success

- Figure out the “pain” points of an application
- Be prepared to scale every part of your application.

Key to Success

- Figure out the “pain” points of an application
- Be prepared to scale every part of your application.
- Be prepared to sacrifice performance for availability, chances are good you won’t be doing the other way around

Key to Success

Key to Success

- Become omniscient and omnipotent.

Key to Success

- Become omniscient and omnipotent.
- Identify Single Points of Failure (SPoF)...

Key to Success

- Become omniscient and omnipotent.
- Identify Single Points of Failure (SPoF)...
- If you have an SPoF... guaranteed it will fail

SPoFs and how to get yourself fired :)

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:
  - ★ external dependencies (isp etc.)

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:
  - ★ external dependencies (isp etc.)
  - ★ physical infrastructure: power.

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:
  - ★ external dependencies (isp etc.)
  - ★ physical infrastructure: power.
  - ★ people

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:
  - ★ external dependencies (isp etc.)
  - ★ physical infrastructure: power.
  - ★ people
  - ★ servers (db, web, load, firewall, dns...)

SPoFs and how to get yourself fired :)

- Do an SPoF audit on your application, SPoFs can be:
  - ★ external dependencies (isp etc.)
  - ★ physical infrastructure: power.
  - ★ people
  - ★ servers (db, web, load, firewall, dns...)
  - ★ application hooks/CRONs

Pain Point #1: The Web and File Servers

Pain Point #1: The Web and File Servers

- A typical PHP application with lots of visitors will have to run on a cluster of web servers.

Pain Point #1: The Web and File Servers

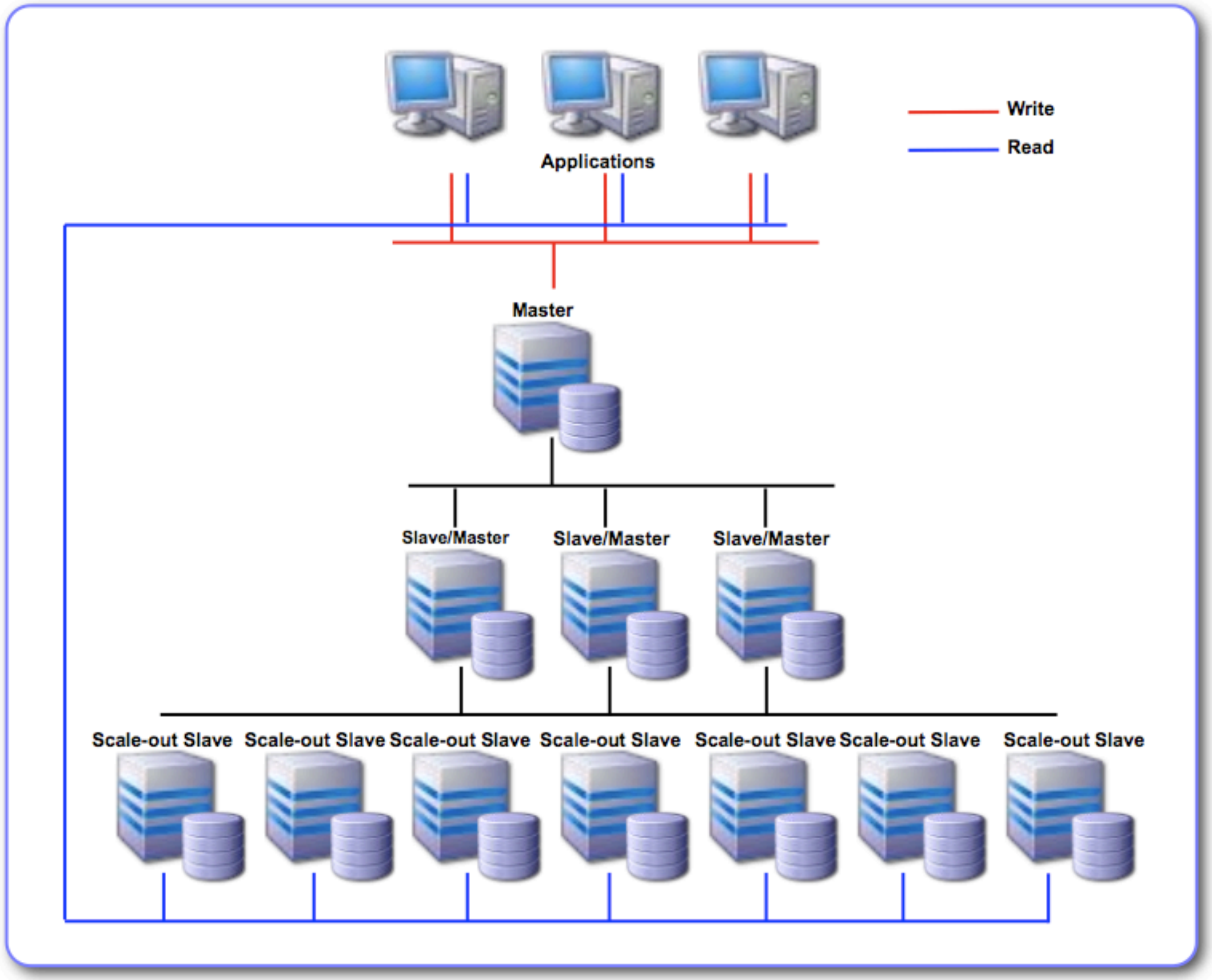
- A typical PHP application with lots of visitors will have to run on a cluster of web servers.
- Centralized file server or pushed file system?

Pain Point #1: The Web and File Servers

- A typical PHP application with lots of visitors will have to run on a cluster of web servers.
- Centralized file server or pushed file system?
- Centralized file server can be a bottleneck, pushed file system limits some programming options.

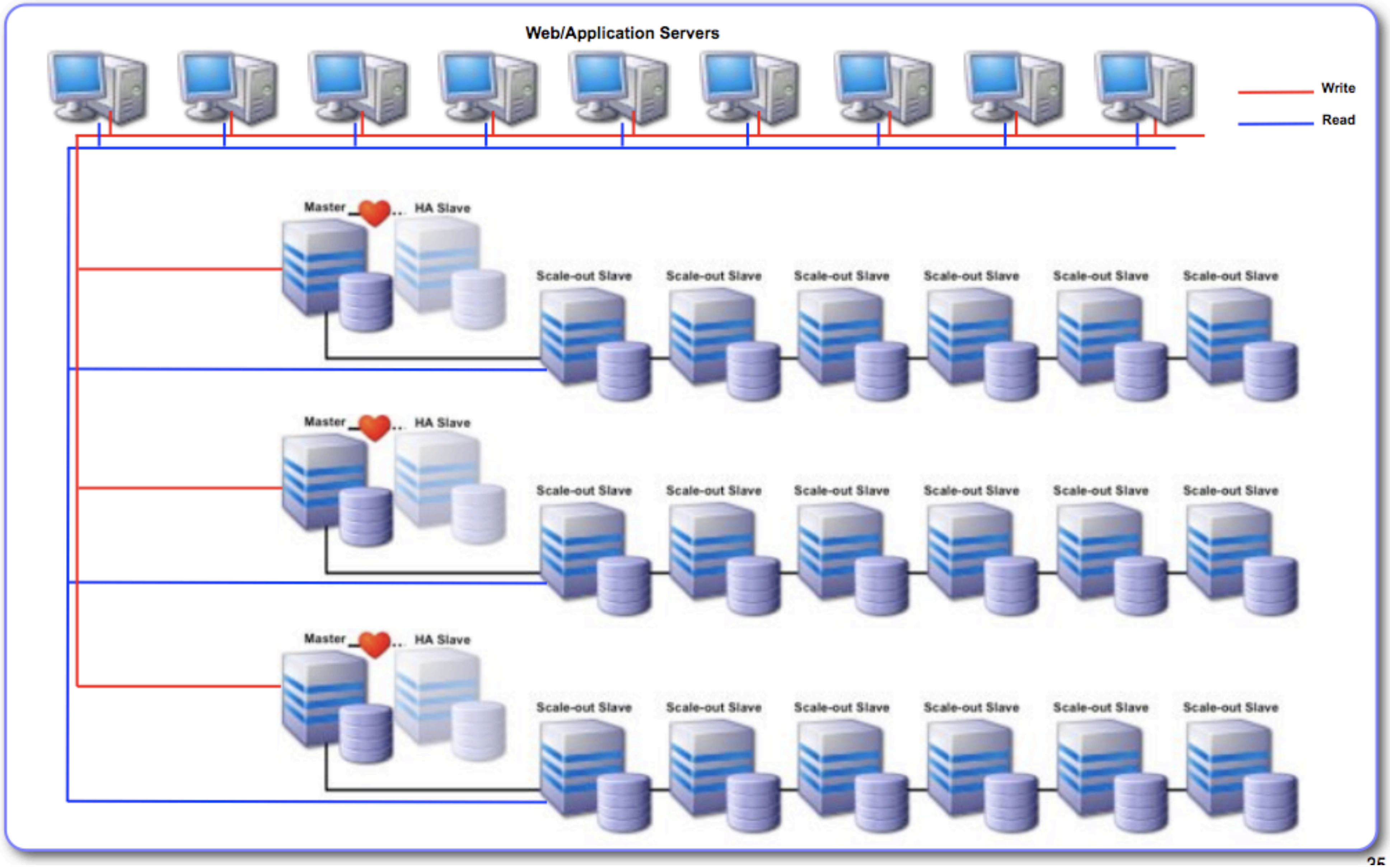
Pain Point #2: The Database

- How will you configure the database.
- Master/Slave?



Pain Point #2: The Database

- Sharding? More common amongst newer social applications.



Pain Point #2: The Database

- Perhaps MySQL Proxy?
- We ran MySQL Proxy as a test on MySQL.com, it's getting there!

- http://forge.mysql.com/wiki/MySQL_Proxy

Pain Point #2: The Database

- Perhaps look at Cloud options such as AWS.
- Allows growth at the least cost and lets someone else handle the problem of scaling for traffic!

Pain Point #3: The Mail Server

Pain Point #3: The Mail Server

- Most social applications depend on vast quantities of emails to be sent out.

Pain Point #3: The Mail Server

- Most social applications depend on vast quantities of emails to be sent out.
- What happens when your SMTP server gives up the ghost? Do you run SMTP servers on your web servers? Isolate the SMTP Servers?

Pain Point #3: The Mail Server

- Most social applications depend on vast quantities of emails to be sent out.
- What happens when your SMTP server gives up the ghost? Do you run SMTP servers on your web servers? Isolate the SMTP Servers?
- We dumped mail into a MySQL Db and sent with custom daemon.

Pain Point #4: Controlling Master/ Slave Writes

- Almost every application of this kind obviously splits out reads to read slaves and writes to masters.
- Use some sort of DNS based load balancing on your DB servers to send queries?

Pain Point #5: Data Caching

- Replicated setups == Replication Lag.
- Replicated Forum software particularly vulnerable with increased traffic, missing posts etc.
- Slave dependent queries for IDs etc. may cause issues with data integrity.

Pain Point #5: Data Caching

Pain Point #5: Data Caching

- Memcached!!! When you write to the database you write to your memcached server, read from memcached before reading from the database again.

Pain Point #5: Data Caching

- Memcached!!! When you write to the database you write to your memcached server, read from memcached before reading from the database again.
- Are you going to the database too much? Counts of Users, Activity etc.?

Pain Point #5: Data Caching

- Memcached!!! When you write to the database you write to your memcached server, read from memcached before reading from the database again.
- Are you going to the database too much? Counts of Users, Activity etc.?
- Before memcached, we used filesystem files written by system processes.

Pain Point #6: The PHP Code

Pain Point #6: The PHP Code

- XDebug. If you aren’t using it. Download it, learn it. Use it. Improve performance of your application. (http://www.xdebug.org)

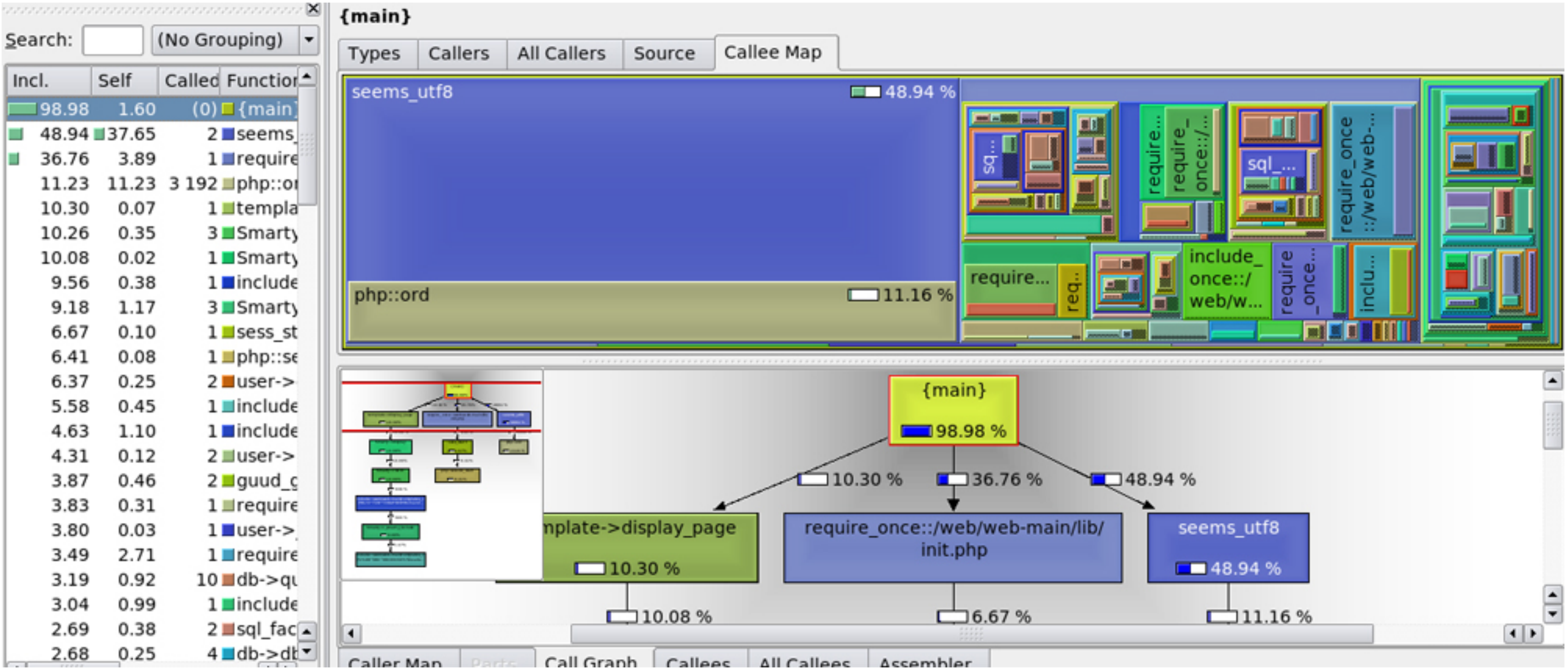
Pain Point #6: The PHP Code

- XDebug. If you aren't using it. Download it, learn it. Use it. Improve performance of your application. (http://www.xdebug.org)
- Profile your application.

Pain Point #6: The PHP Code

- XDebug. If you aren't using it. Download it, learn it. Use it. Improve performance of your application. (http://www.xdebug.org)
- Profile your application.
- Take a lesson from a high visibility site: Wikipedia, run a fraction of your requests through xdebug and profile.

Pain Point #6: The PHP Code



Search: (No Grouping)

Incl.	Self	Called	Function
98.98	1.60	(0)	{main}
48.94	37.65	2	seems_
36.76	3.89	1	require
11.23	11.23	3 192	php::or
10.30	0.07	1	templa
10.26	0.35	3	Smarty
10.08	0.02	1	Smarty
9.56	0.38	1	include
9.18	1.17	3	Smarty
6.67	0.10	1	sess_st
6.41	0.08	1	php::se
6.37	0.25	2	user->
5.58	0.45	1	include
4.63	1.10	1	include
4.31	0.12	2	user->
3.87	0.46	2	guud_c
3.83	0.31	1	require
3.80	0.03	1	user->_
3.49	2.71	1	require
3.19	0.92	10	db->qu
3.04	0.99	1	include
2.69	0.38	2	sql_fac
2.68	0.25	4	db->db

A profile of mysql.com in April 2008

Pain Point #6: Monitoring

Pain Point #6: Monitoring

- If a person falls in the forest do you hear the PHP Fatal Error?

Pain Point #6: Monitoring

- If a person falls in the forest do you hear the PHP Fatal Error?
- Be omniscient in your applications. If something goes wrong do not wait for someone to tell you.

Pain Point #6: Monitoring

- If a person falls in the forest do you hear the PHP Fatal Error?
- Be omniscient in your applications. If something goes wrong do not wait for someone to tell you.
- Build monitoring into the application, but do you want High Performance?

Pain Point #6: Monitoring

Pain Point #6: Monitoring

- Capture your errors and logging into log files which are then monitored.

Pain Point #6: Monitoring

- Capture your errors and logging into log files which are then monitored.
- Establish a good monitoring tool which monitors not only the Servers but your Application.

Pain Point #6: Monitoring

- Capture your errors and logging into log files which are then monitored.
- Establish a good monitoring tool which monitors not only the Servers but your Application.
- Shameless plug for both MySQL Enterprise Monitoring and my own open source BigDaddy (bigdaddymonitor.org) which grew out of all these pain points

Pain Point #7: Your SQL Queries

Pain Point #7: Your SQL Queries

- In the end a PHP/MySQL application lives and dies on the strength of your queries.

Pain Point #7: Your SQL Queries

- In the end a PHP/MySQL application lives and dies on the strength of your queries.
- Make sure that you have good indexes on your tables. EXPLAIN always.

Pain Point #7: Your SQL Queries

- In the end a PHP/MySQL application lives and dies on the strength of your queries.
- Make sure that you have good indexes on your tables. EXPLAIN always.
- Make sure that you have query caching turned on go examine your slow query log.

Pain Point #7: Your SQL Queries

Pain Point #7: Your SQL Queries

- Queries do not always scale!

Pain Point #7: Your SQL Queries

- Queries do not always scale!
- Use some sort of query analyzer, custom or third party.

Pain Point #7: Your SQL Queries

- Queries do not always scale!
- Use some sort of query analyzer, custom or third party.
- When you develop, try to test expensive queries against a proper data set size.

Pain Point #8: Ajax, Javascript

Pain Point #8: Ajax, Javascript

- App performance is what the client sees, not what the server/server-op sees

Pain Point #8: Ajax, Javascript

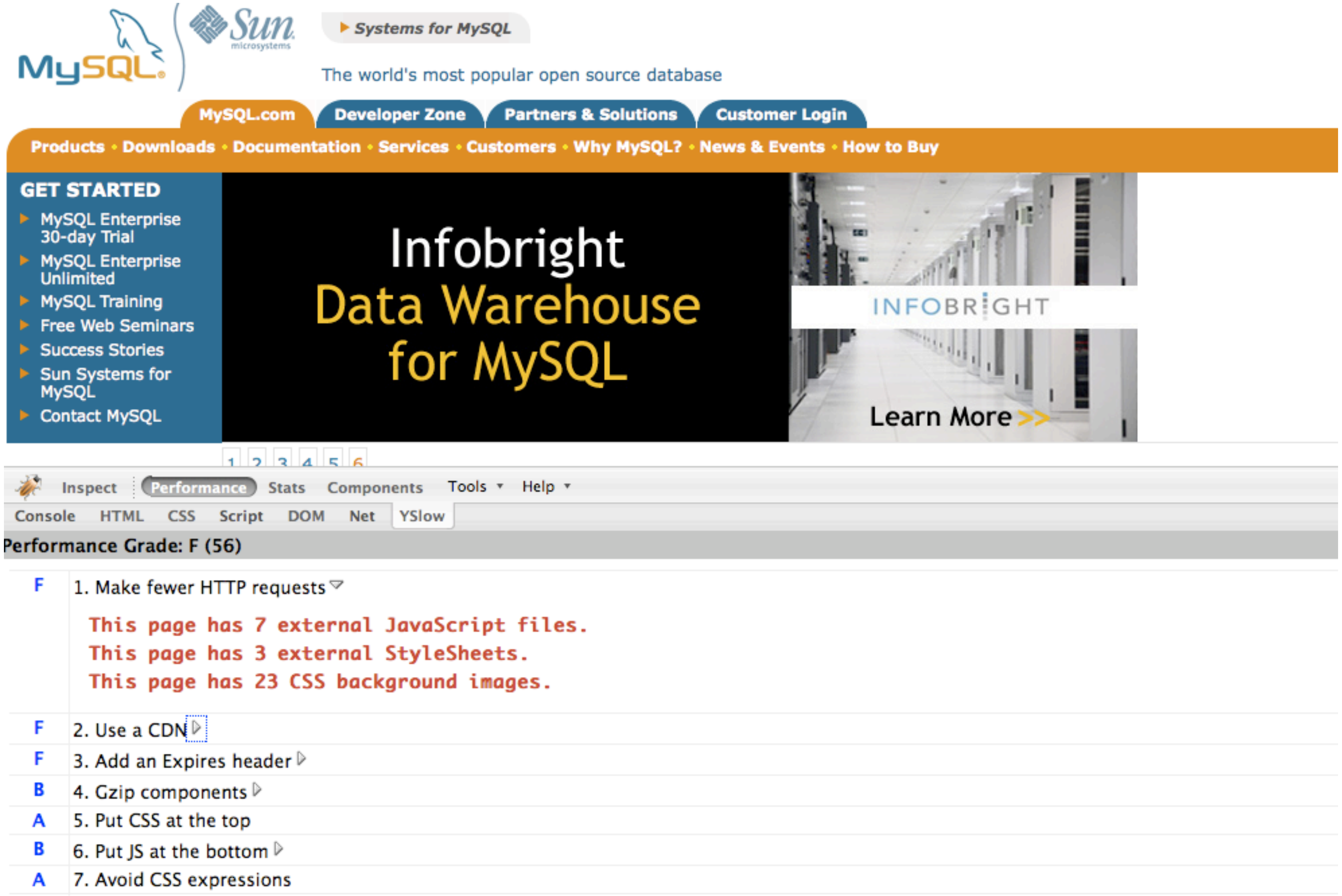
- App performance is what the client sees, not what the server/server-op sees
- DB Setup tuned for "Web 2.0" apps? Ajax applications tend to be less read heavy and more write heavy.

Pain Point #8: Ajax, Javascript

- App performance is what the client sees, not what the server/server-op sees
- DB Setup tuned for "Web 2.0" apps? Ajax applications tend to be less read heavy and more write heavy.
- InnoDB versus MyISAM for primary key lookups.

Pain Point #8: Ajax, Javascript

Client tuning is essential as much as server tuning. YSlow is one option:



Performance Grade: F (56)

F 1. Make fewer HTTP requests

```
This page has 7 external JavaScript files.
This page has 3 external StyleSheets.
This page has 23 CSS background images.
```

F 2. Use a CDN

F 3. Add an Expires header

B 4. Gzip components

A 5. Put CSS at the top

B 6. Put JS at the bottom

A 7. Avoid CSS expressions

Pain Point #9: All the other things

Pain Point #9: All the other things

- Over the years pain points have come in all shapes and sizes, including our apache logs :)

Pain Point #9: All the other things

- Over the years pain points have come in all shapes and sizes, including our apache logs :)
- We ended up creating a sharded db system with a simple perl script to dump web logs into a MySQL database.

Pain Point #9: All the other things

- Over the years pain points have come in all shapes and sizes, including our apache logs :)
- We ended up creating a sharded db system with a simple perl script to dump web logs into a MySQL database.
- Oddly worked as well if not better than a file system.

Final thought...

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.
  - ★ NFS Crash

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.
  - ★ NFS Crash
  - ★ File system corrupt

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.
  - ★ NFS Crash
  - ★ File system corrupt
  - ★ DB Crash, Table corrupted

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.
  - ★ NFS Crash
  - ★ File system corrupt
  - ★ DB Crash, Table corrupted
  - ★ Backup corrupted by another sequence of events.

Final thought...

- Murphy's Extended Law: If a series of events can go wrong, they will do so in the worst possible sequence.
  - ★ NFS Crash
  - ★ File system corrupt
  - ★ DB Crash, Table corrupted
  - ★ Backup corrupted by another sequence of events.
  - ★ I was on holiday

The moral of this sordid tale

The moral of this sordid tale

- Murphy Loves Web Application Developers.

The moral of this sordid tale

- Murphy Loves Web Application Developers.
- Everything goes wrong at some point

The moral of this sordid tale

- Murphy Loves Web Application Developers.
- Everything goes wrong at some point
- Just be prepared

The moral of this sordid tale

- Murphy Loves Web Application Developers.
- Everything goes wrong at some point
- Just be prepared
- Eliminate every SPoF (Single Point of Failure) in your system.

Sun microsystems