

Performance Tuning Sakai for Production

Jason Shao, Rutgers University



Who is this guy?

Oops, that's my son...

Who is **this** guy?

- Application Developer
- **Rutgers University**
- Office Instructional & Research Technology
- Sakai quasi-Tech Lead
- formerly of myRutgers fame...



What's he talking about?

- Approach
- Systems Architecture
- Tuning: Browsers, DB, JVM
- Testing: Load, Performance, QA
- Monitoring

Approach

1. Requirements
2. Measure Performance
3. Identify Bottlenecks
4. Fix Bottlenecks
5. Wash, Rinse, Repeat (2-4)

Measurements

Measurements

- Responsiveness

Measurements

- Responsiveness
- Memory Usage

Measurements

- Responsiveness
- Memory Usage
- CPU Utilization

Measurements

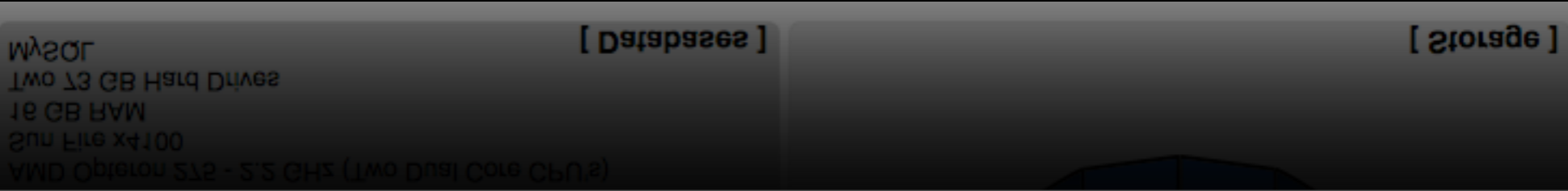
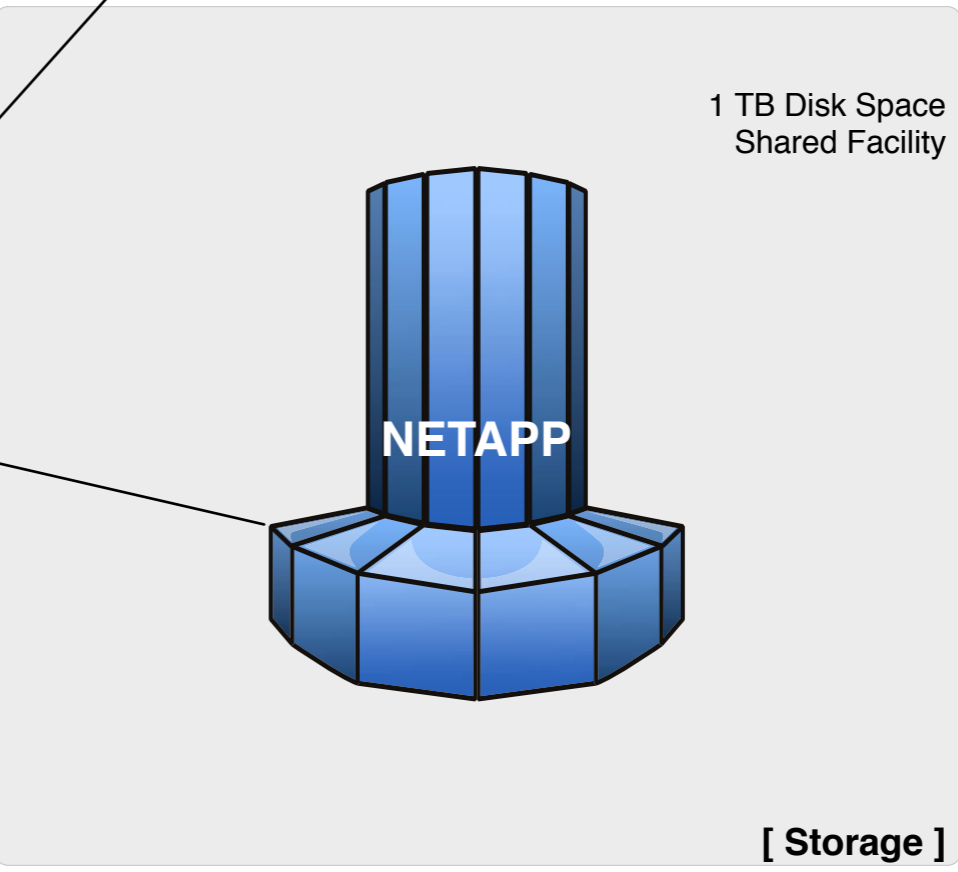
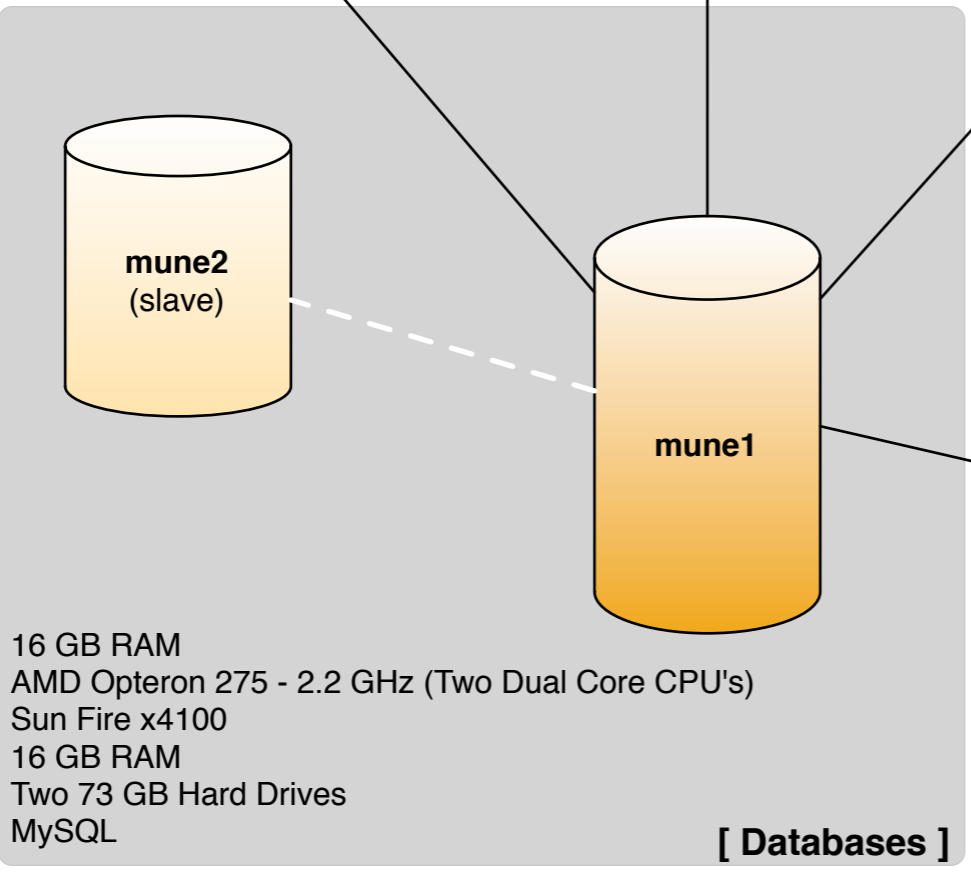
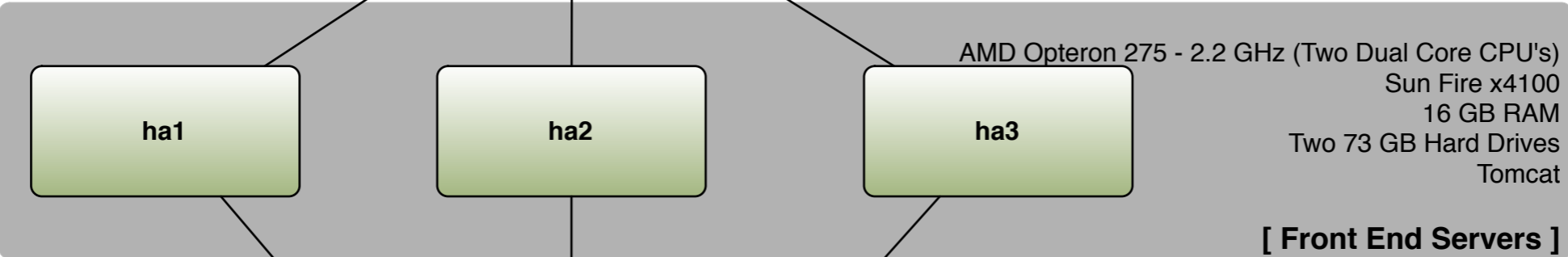
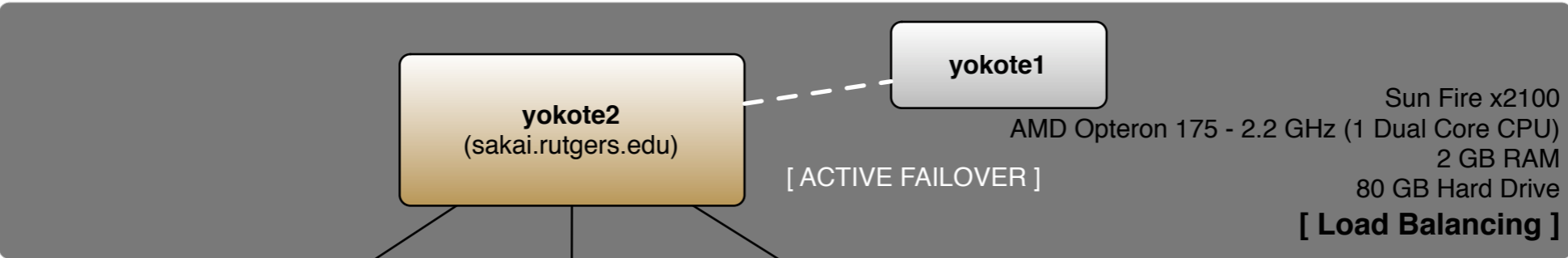
- Responsiveness
- Memory Usage
- CPU Utilization
- Scalability

Variables

- Tools & Services
- Usage Patterns
 - Scenarios (e.g. use cases)
 - Traffic Patterns

Systems Architecture

- Availability
- Capacity
- Complexity



Perceived Performance

- Mostly browser-side
- Loading Resources
- Building Browser Model
 - DOM, CSS, JS

Browser Tuning

- Caching
- Connections
- Rendering
- Tools

HTTPS

- Changes default browser caching
- Less info cached to disk
- May not be getting caching benefit you think you are

E-tags

- Bad, Bad, Bad
- Typically based off i-node/file handle
- Kill caching in clustered environment

ETag Format

- Apache: inode-size-timestamp
- IIS: Filetimestamp:ChangeNumber

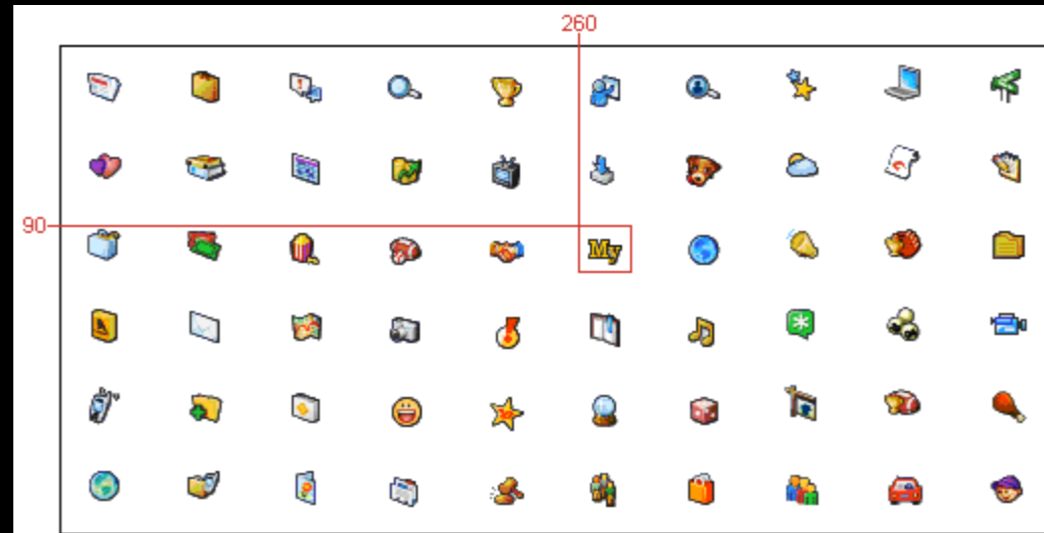
Expires

- Good, Good, Good
- Prevent round trips to the server

Connections

- Most browsers use 2 threads per host to retrieve resources

CSS Sprites



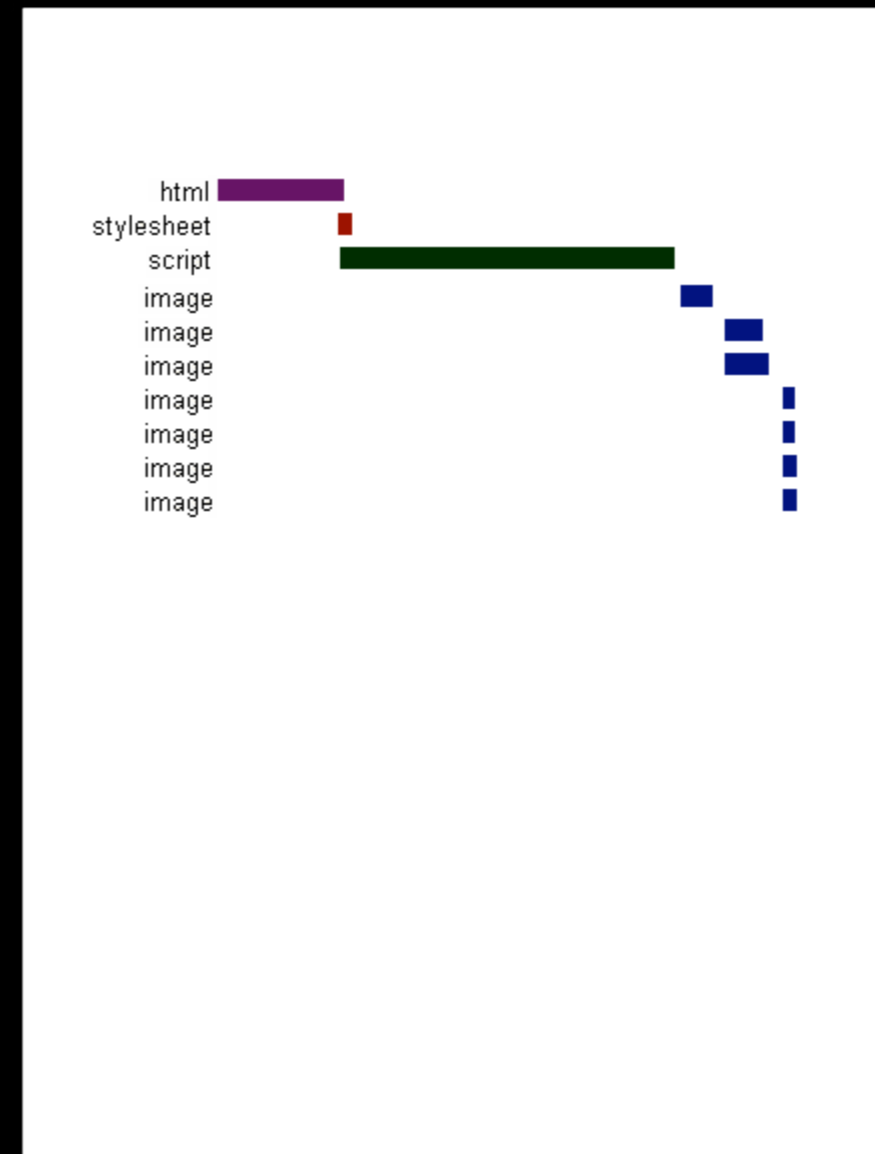
Gzip

CSS at top

stylesheets block rendering in IE

Scripts at the bottom

- scripts block rendering
- & downloads



Externalize

(generally)

Minimize Redirects

- 3xx status codes – mostly 301 and 302
 - HTTP/1.1 301 Moved Permanently
- add Expires headers to cache redirects
- Watch for trailing slashes (esp. w/Apache)

DB Tuning

- Indexing
- Caching
- Connection Pooling
- Table Size
- Query Tuning

Index, Index, Index

- Some Hibernate bugs: indexes not created
- Slight differences between MySQL/Oracle
 - Remember: Primary/Foreign keys

Caching

- Hibernate Caching
- Provider Caching
- Database-side caching & tuning
 - e.g. MySQL QueryCaching

MySQL Example

- url@javax.sql.DataSource=jdbc:mysql://localhost:3306/sakai?useUnicode=true&characterEncoding=UTF-8&useServerPrepStmts=false&cachePrepStmts=true&prepStmtCacheSize=4096&prepStmtCacheSqlLimit=4096

Some tables get big...

- SAKAI_SESSION
- SAKAI_EVENT

Query Tuning

- Slow query logging or explain can help you find queries that need optimization or tables that need indexing, etc.

Firebug

<https://addons.mozilla.org/en-US/firefox/addon/1843>

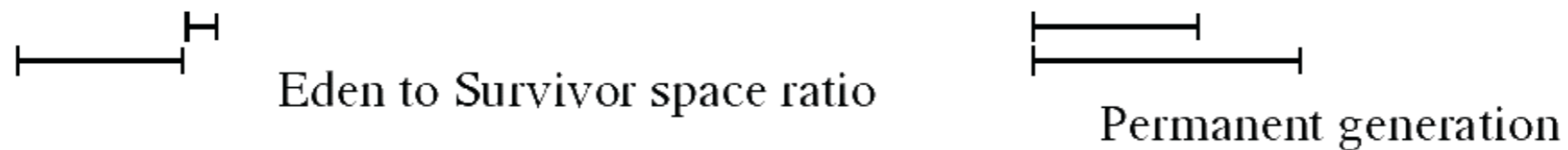
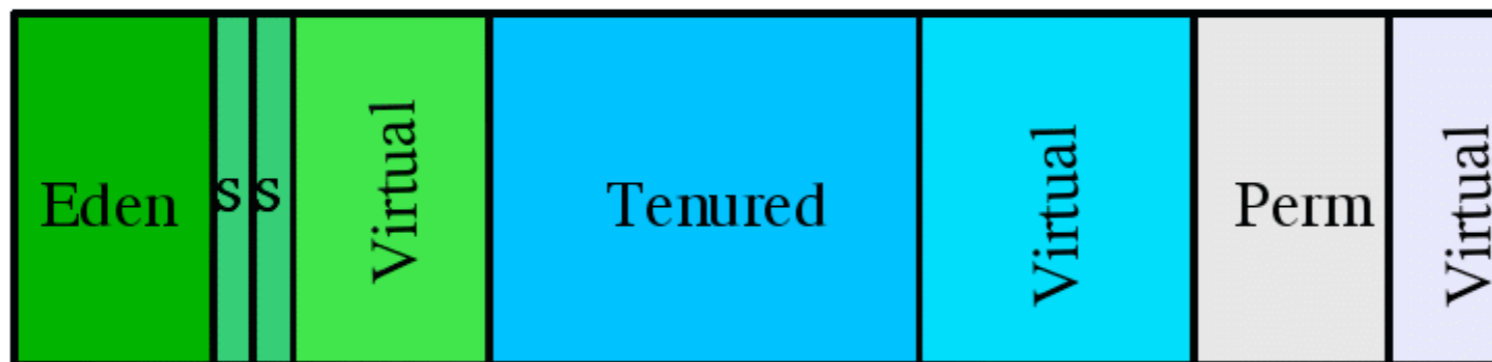
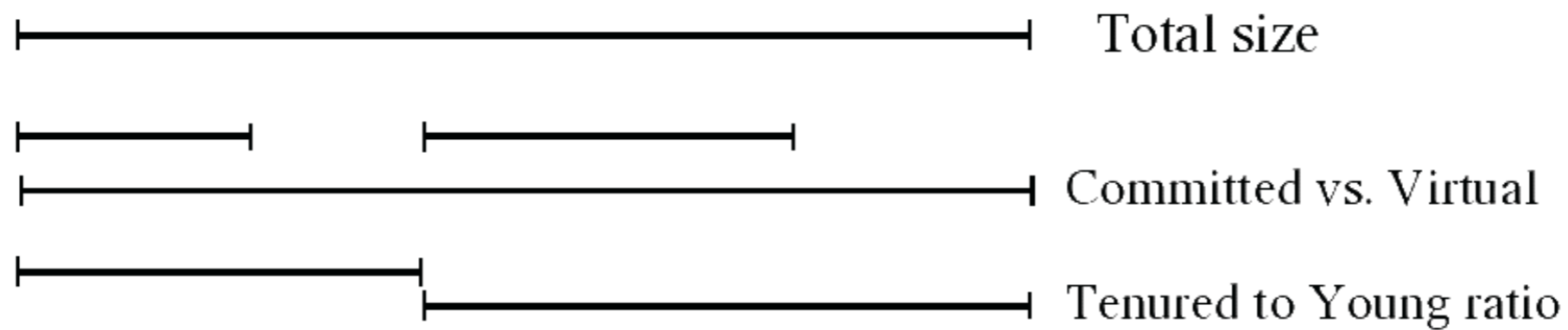
JVM Tuning

- JVM Memory Model
- JVM Versions
- Heap Settings
- GC Tuning

JVM Versions

- JDK 1.5
- JDK 1.5 (64-bit)
- JDK 6

Generational JVM Memory Model



JAVA_OPTS

```
-d64 -Xmx10000m -Xms10000m -Xmn3g  
-XX:+UseConcMarkSweepGC  
-XX:+UseParNewGC  
-XX:SurvivorRatio=8  
-XX:TargetSurvivorRatio=90  
-XX:MaxTenuringThreshold=15  
-XX:MaxPermSize=512m  
-XX:PermSize=64m
```

VerboseGC

- verbose:gc -XX:+PrintGCDetails
- XX:+PrintGCTimeStamps
- XX:+PrintTenuringDistribution
- XX:+PrintAdaptiveSizePolicy
- XX:+PrintGCApplicationConcurrentTime
- XX:+PrintGCApplicationStoppedTime

Profilers

e.g. Yourkit

Monitoring

- System Status
- Tomcat Logging
- DB Logging

Monitoring

- Cron
- Nagios, Openview, etc.
- JMX

Tomcat Logs

- catalina.out -- log4j tweaking
- grep
- products like: splunk

MySQL

- `--log--slow-queries`
 - also set in `my.cnf`
- `mysqldumpslow`

Acknowledgements

- Dan Mindler - everything I know about the JVM I learned from Dan
- Chuck Hedrick - lots of experimentation at Rutgers with tweaking & tuning
- Steve Souders @ Yahoo - great presentation on client tuning
- sakai-dev: good stuff