BOOTCAMP 2011

High Performance dotCMS

dotCMS

BOOTCAMP 2011

Overview

- OS/Hardware
- Clustering
- Java
- Tomcat
- dotCMS Settings
- Page Design

dotCMS

BOOTCAMP 2011

Hardware

good hardware = good performance

Hardware is the cheapest performance tweak you can do

Virtualization can be done, but there is a significant performance impact. Same holds true for db server… Test!

Note about the cloud: Amazon's largest "Cluster CPU Mega Instance" is only equal to a dual Xeon Server.

dotCMS

BOOTCAMP 2011

Clustering

dotCMS has been developed to scale horizontally

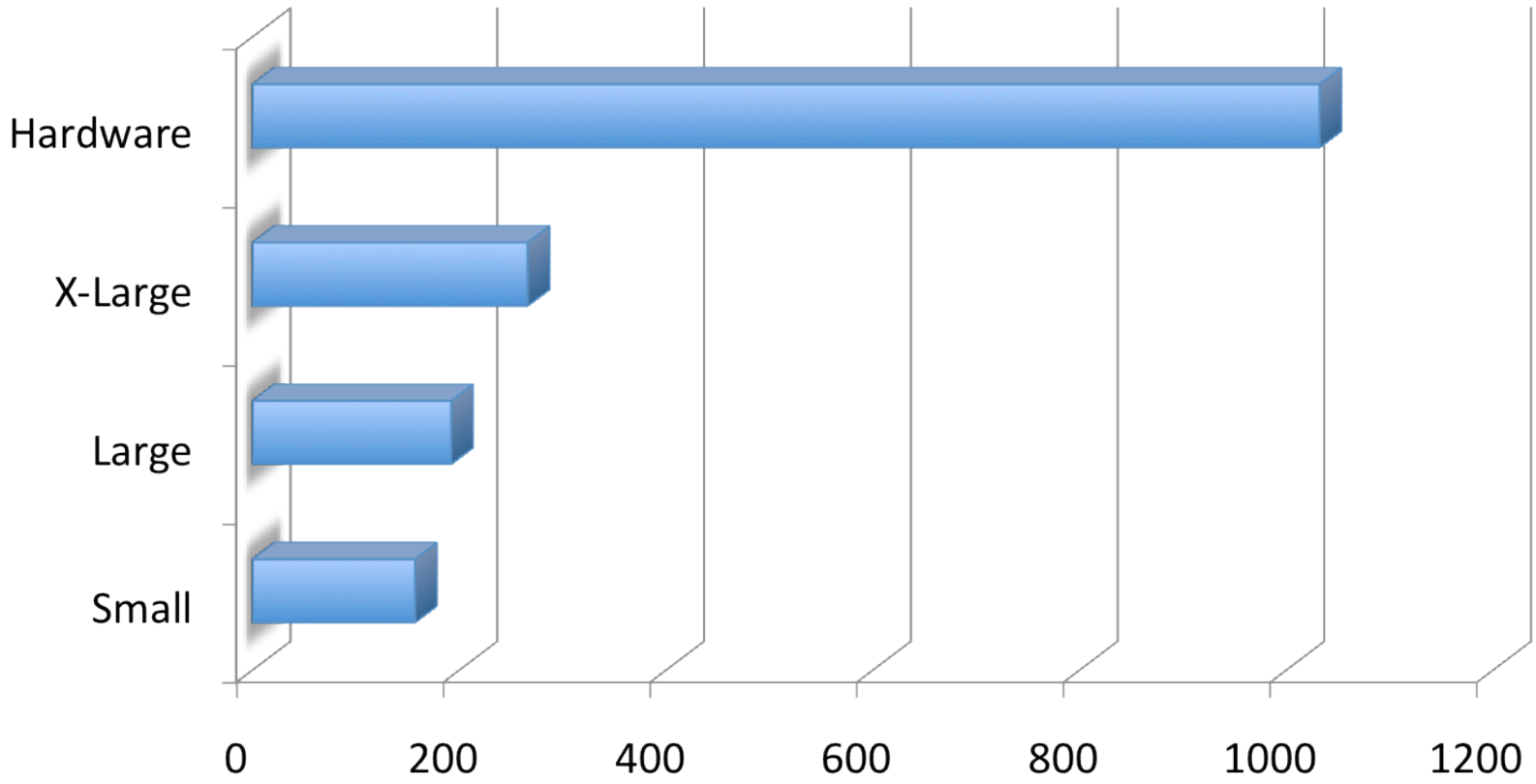
dotCMS Nodes (once cache is loaded) share nothing on page views.

Adding nodes to a cluster can improve performance almost linearly

dotCMS

BOOTCAMP 2011

Hardware vs. Amazon Instances (hits per sec)



dotCMS

BOOTCAMP 2011

#1 : Hardware

I/O is critical for dotCMS performance:

- H2 Cache
- Lucene
- Logs

Make sure the /dotCMS/dotsecure directory and the /assets directory are on fast disks and the log files are on separate partitions/disks

SSDs – anecdotal evidence is that these can make a SERIOUS improvement in performance.

For more on SSDs, see (warning, bad language):
http://www.livestream.com/oreillyconfs/video?clipId=pla_3beec3a2-54f5-4a19-8aaf-35a839b6ecaa

dotCMS

BOOTCAMP 2011

Hardware

DB setup is important to dotCMS

We generally recommend a dedicated DB server (dotCMS can still pull and display data from other databases)

Make sure your DB is tuned and performant before launching your site.

dotCMS

Java

Java has tons of performance tweaks…

-XX:+UseStringCache

-XX:+OptimizeStringConcat

-XX:+UseFastAccessorMethods

-XX:+AggressiveOpts

-XX:+UseLargePages

-server, etc…

Java

Unfortunately, these provide little to improve dotCMS responsiveness.

Couple of caveats:

-XX+UseLargePages has the benefit of preventing java from being swapped from memory.

-XX:PermSize should be increased if lots of plugins or 3rd party libs are used in dotCMS

Java

Garbage collection:

- GC logging can be used to identify problems on production systems
- The default GC is still our choice – it works as well or better than almost any other garbage collection system, ever…

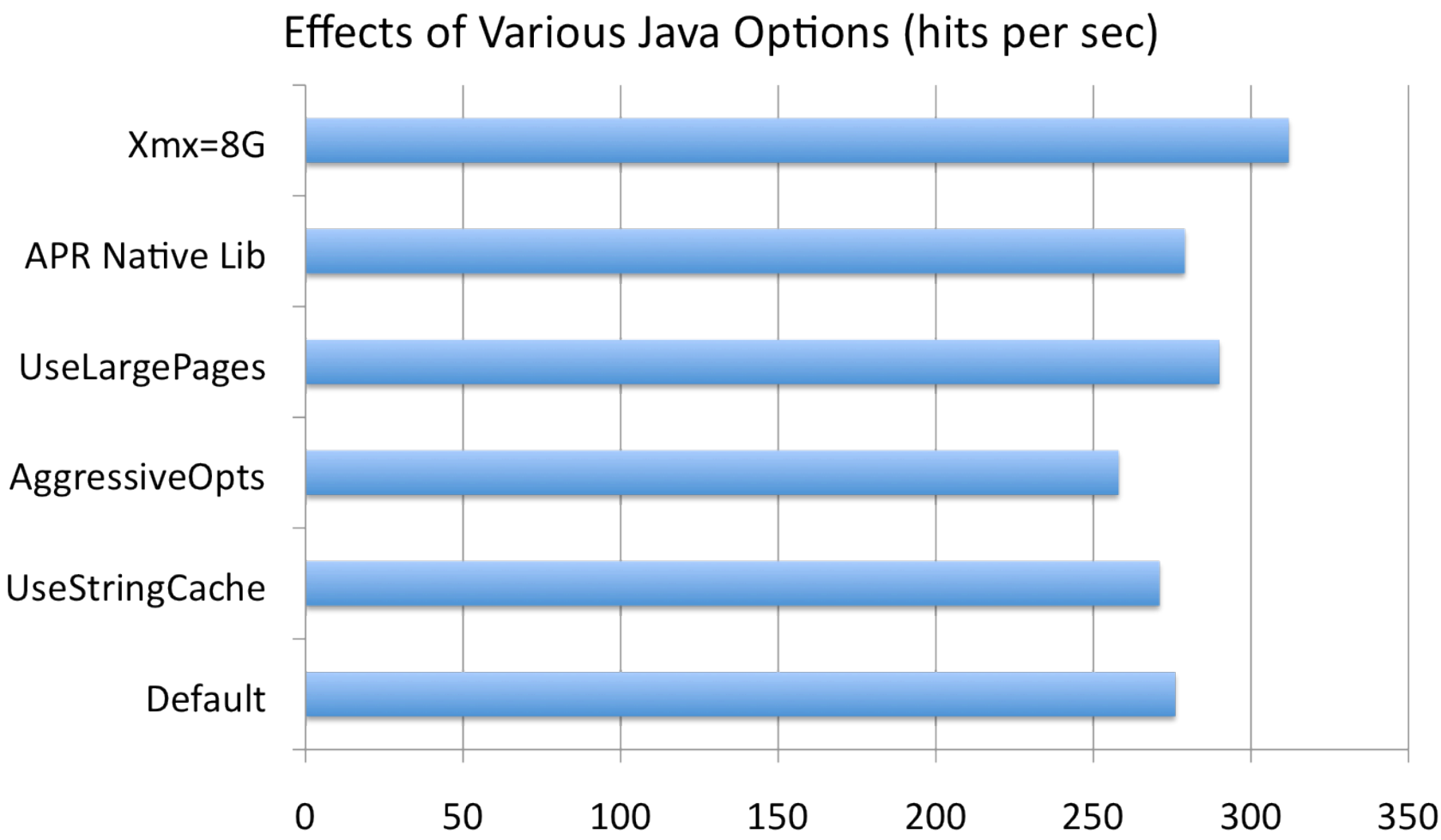
BOOTCAMP 2011

Java

-Xmx !

dotCMS

BOOTCAMP 2011



dotCMS

Tomcat

Some tomcat best practices:

- Up the maxThreads (~400)
  - Note, you will need a possible db connection for each thread
- lower the default connectionTimeout (3000)
- APR (Apache Portable runtime) does not make much of a difference
- For other tuning ideas, see: http://blog.monitis.com/index.php/2011/08/06/18-java-tomcat-application-optimization-tips/

BOOTCAMP 2011

dotCMS

If you are running with more than 6GB, use the large cache setting

Make sure your hostnames are correct – don't rely on the default host to catch traffic. Results in a 4x

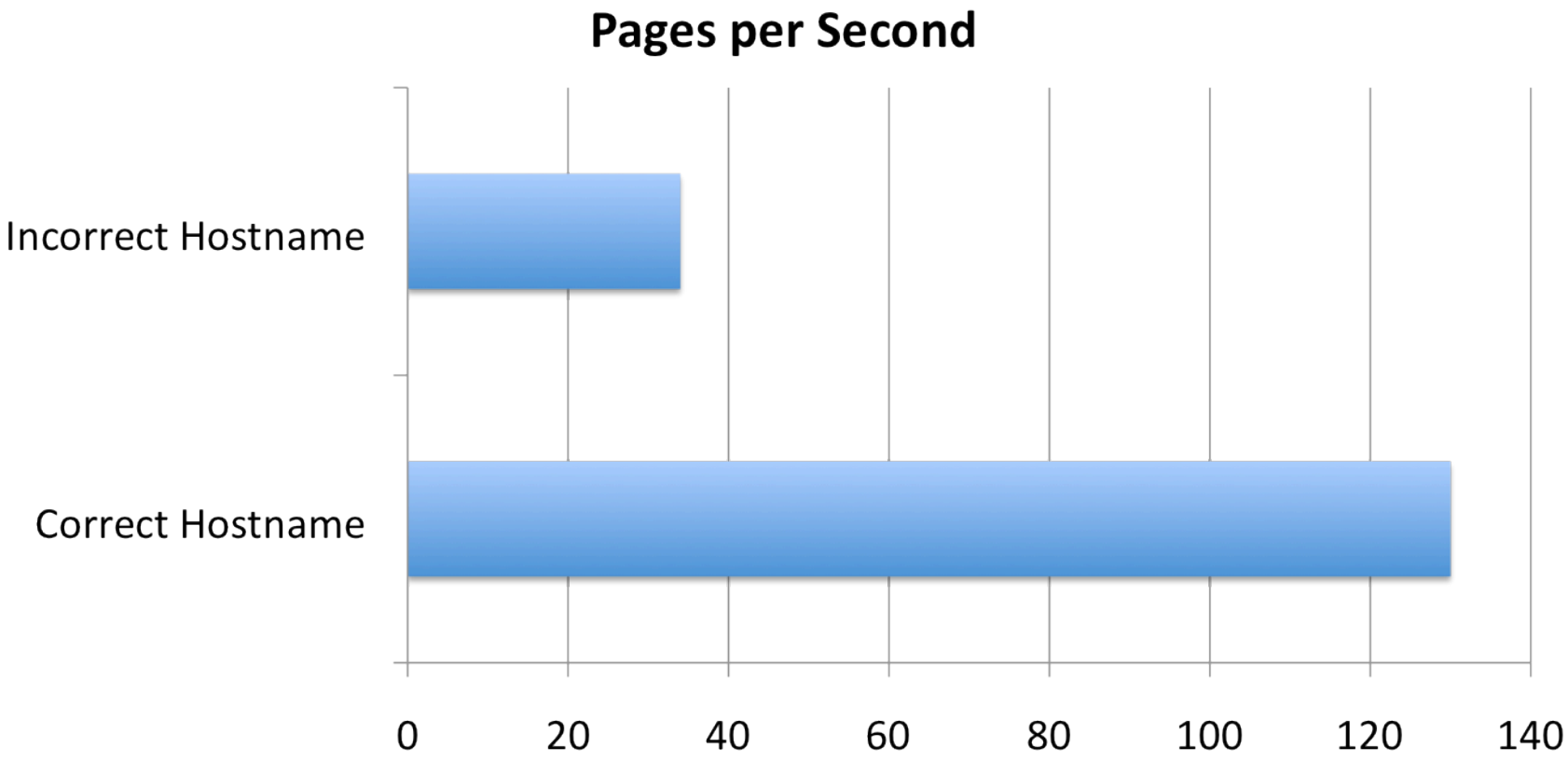
Delete old assets regularly

Turn off unused functions– ratingThread, clickstream tracking, calendar reminder, dashboard

dotCMS

BOOTCAMP 2011

Incorrect Hostnames



dotCMS

BOOTCAMP 2011

Page Design

Avoid un-cached Remote Calls – unless you can verify the performance of the remote call and network, use JavaScript instead

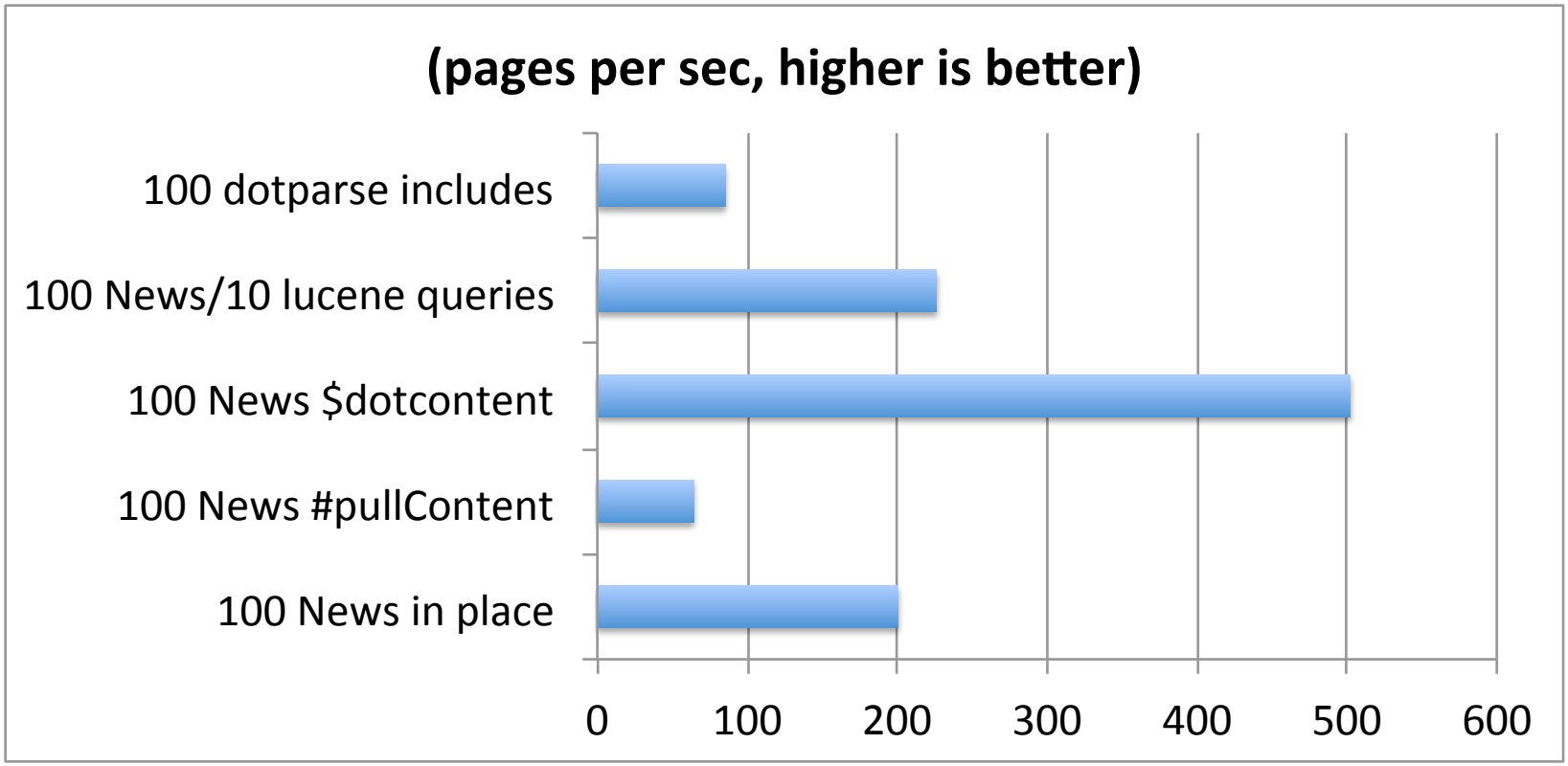
Replace all #pullContent uses with $dotcontent

Dynamic pulls and includes have a cost. Use the caches to offset this

dotCMS

BOOTCAMP 2011

Page Design and Performance



dotCMS

BOOTCAMP 2011

Page Design

Use #dotcache – takes some thought to select the proper key

```
#dotcache(”cache-heavy-pull", 60)
    …… my heavy pull……
#end
```

Use the page cache – takes some thought to design pages to take advantage of this

dotCMS

BOOTCAMP 2011

Parting Words

Test, iterate, Test again

ab for ad-hoc tests

dotCMS