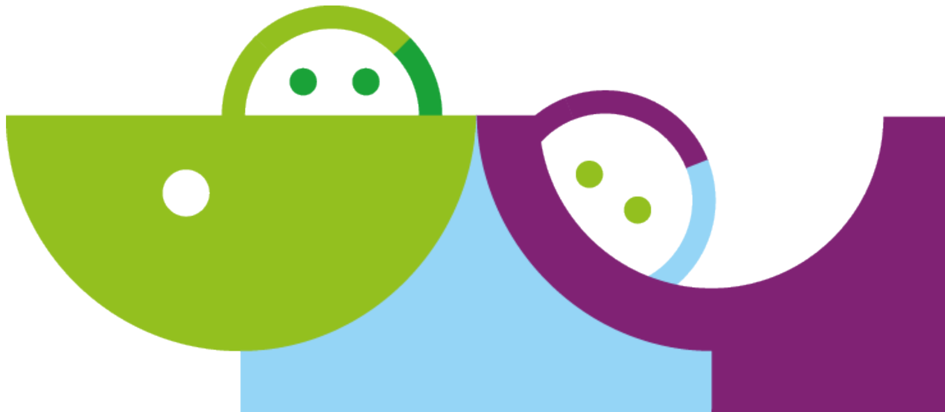
WebSphere Portal Performance

Klaus Nossek



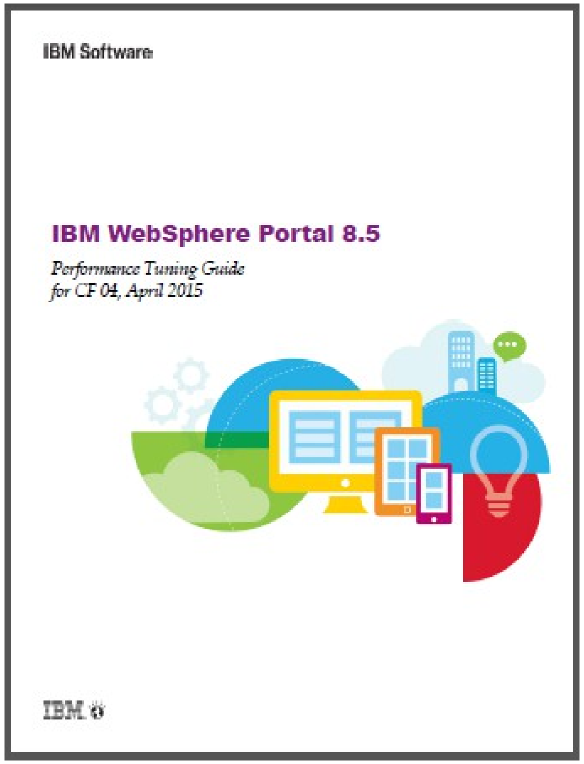
Agenda

- WebSphere Portal Performance Guidelines
- Precise Planning and Managing of Performance
- Techniques to improve Performance
  - Caching Techniques
  - Reducing the number of requests
  - Shrinking the page size
  - AJAX
- Do you have a Performance Problem ?
  - Performance Tuning
  - Performance Troubleshooting
  - Performance Tools

WebSphere Portal Performance Guidelines

WebSphere Portal Performance Guidelines

The WebSphere Portal Performance Team is responsible to run performance benchmarks and measurements for portal releases and features before they ship out to the customers to make sure that they perform well

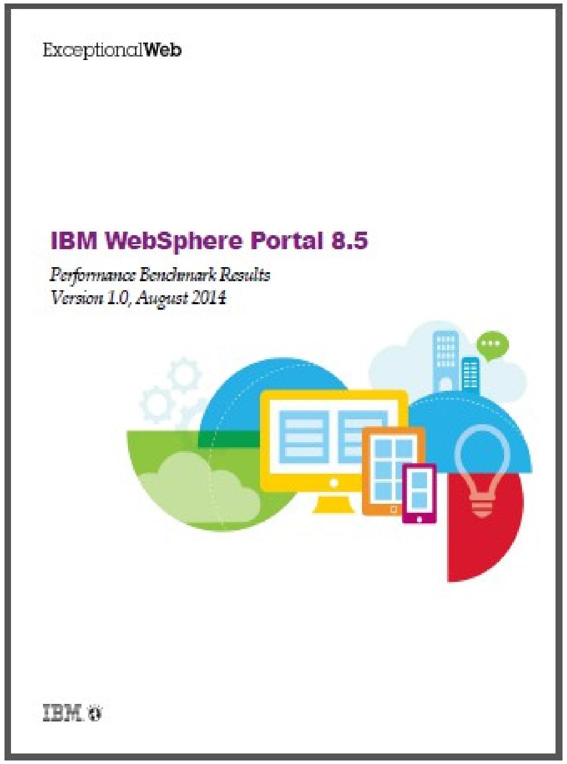


IBM Software

IBM WebSphere Portal 8.5

Performance Tuning Guide
for CF 04, April 2015

IBM

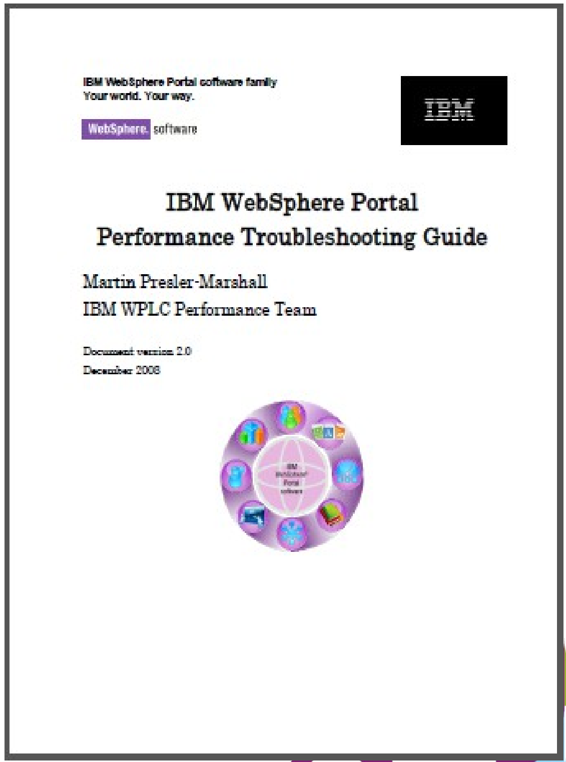


ExceptionalWeb

IBM WebSphere Portal 8.5

Performance Benchmark Results
Version 1.0, August 2014

IBM



IBM WebSphere Portal software family
Your world. Your way.

WebSphere. software

IBM

IBM WebSphere Portal
Performance Troubleshooting Guide

Martin Presler-Marshall
IBM WPLC Performance Team

Document version 2.0
December 2008

WebSphere Portal Performance Guidelines

The experiences our team has gathered is published in the performance guidelines.

The purpose of these papers is to provide guidelines for performance planning and tuning:

- Websphere Portal Performance **Benchmark Results**

https://w3-03.ibm.com/tools/cm/iram/oslc/assets/E4A38100-7B02-F39C-A3D3-3870A7E26BB7/1.0

- Websphere Portal Performance **Tuning Guide**

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/IBM_WebSphere_Portal_V_8.5_Performance_Tuning_Guide

Websphere Portal Performance **Troubleshooting Guide**

http://www-10.lotus.com/ldd/portalwiki.nsf/dx/02092009093345AMWEBK46.htm

- Websphere Portal Performance **Wiki**

http://www-10.lotus.com/ldd/portalwiki.nsf/xpViewCategories.xsp?lookupName=Performance

Precise Planning and Managing of Performance

Precise Planning and Managing of Performance

Precise planning and managing of performance is a crucial factor for the success of the project

<u>Sizing and Capacity Planning</u>

- Architecting your system for performance requires understanding of the business value and workload characteristics
- Techline Software Sizing
- The performance goals are usually defined in terms of:
    - Response Time = time to complete a unit of work (lower is better)
    - Throughput = units of work per time, pages per second (higher is better)
- The goals are usually a combination of the two
    - X pages per second with average response time < Y seconds

Precise Planning and Managing of Performance

Precise planning and managing of performance is a crucial factor for the success of the project

<u>Design your system up front</u>

- Consider techniques/best practices which supports you in achieving the performance goals
- Determine your caching strategy depending on your business requirements

<u>Plan performance tests</u> which helps you to identify the performance of your application

- You need performance tools - performance must be measured - We use the benchmark tool Rational Performance Tester
- You need a performance test environment - measurements must be made in a system as similar as possible to the production one

Techniques to improve Performance

- Caching Techniques
- Reducing the number of requests
- Shrinking the page size
- AJAX

Caching Techniques

Why Caching ?

Portal is a complex system

- Non-trivial amount of CPU processing to layout & render portlets
- Many database requests from Portal itself, WCM & your application

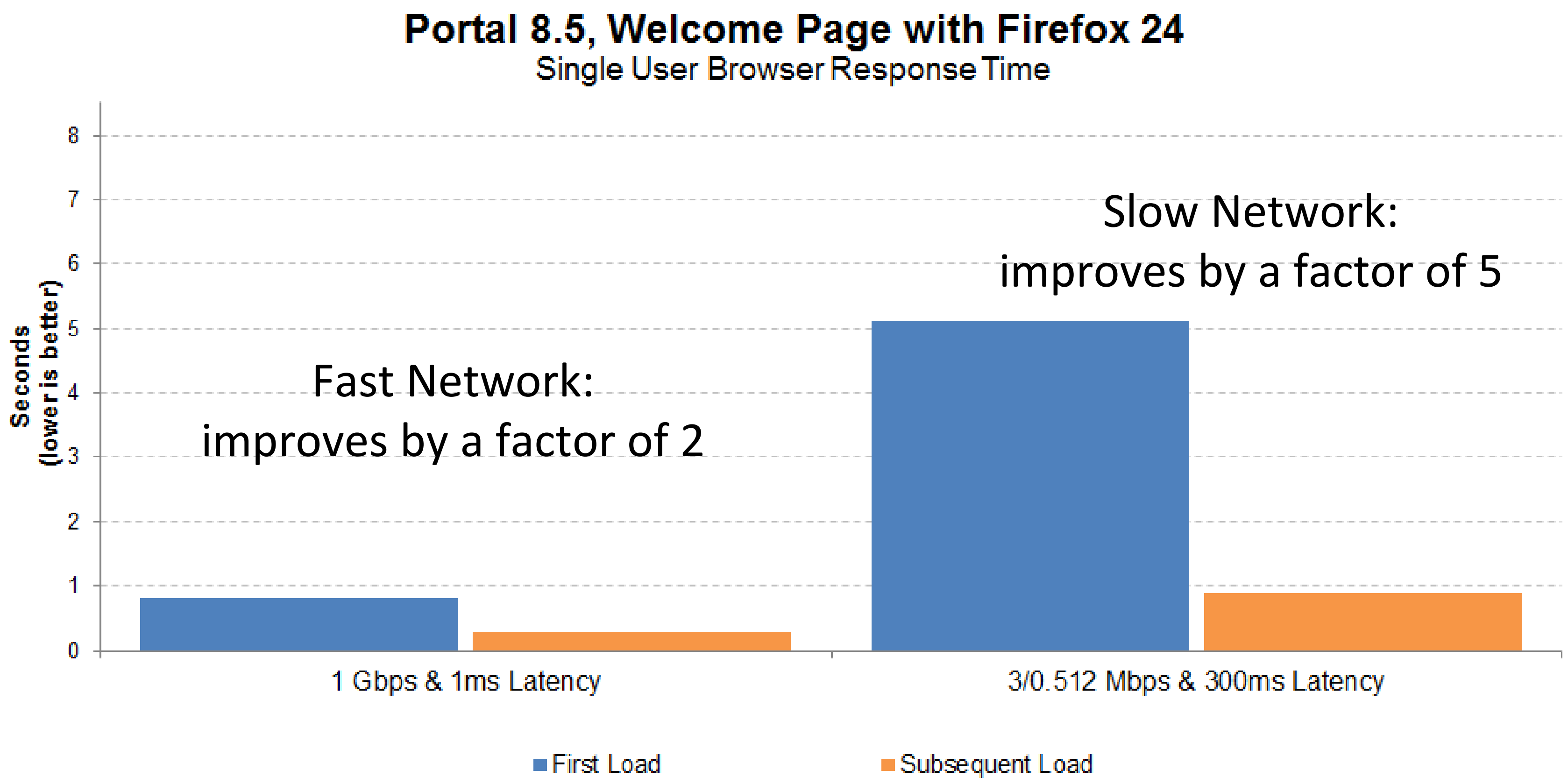
Rich web applications have a lot of content

- Complex CSS
- JavaScript Frameworks
- Many UI Images
- Network bandwidth can slow delivery to end users

Many users will be accessing Portal at the same time

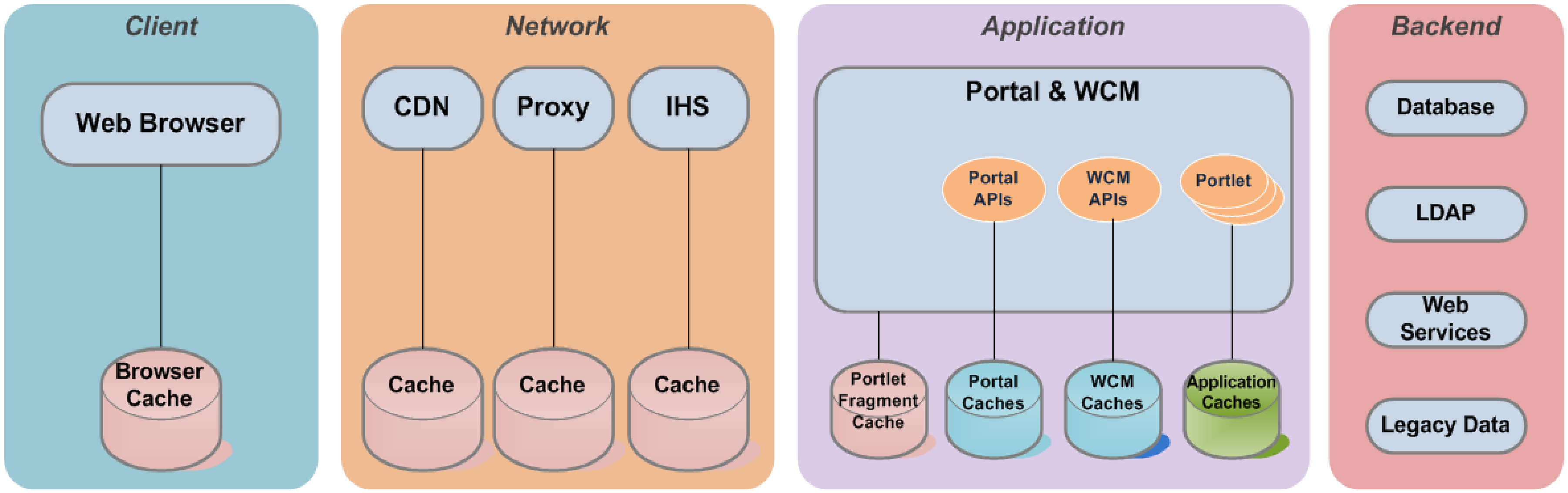
Without any caching, Portal would be inoperable

Caching: Fast versus Slow Network



First Load without Caching – Subsequent Load with Caching

Caching Architecture



HTTP Responses:	HTTP Responses:	HTTP Responses from a portlet
Web Pages	Web Pages	Internal Portal & WCM data
Images	Images	Backend query results
CSS	CSS	Custom application data
JavaScript	JavaScript	
Private, Public	**Public**	

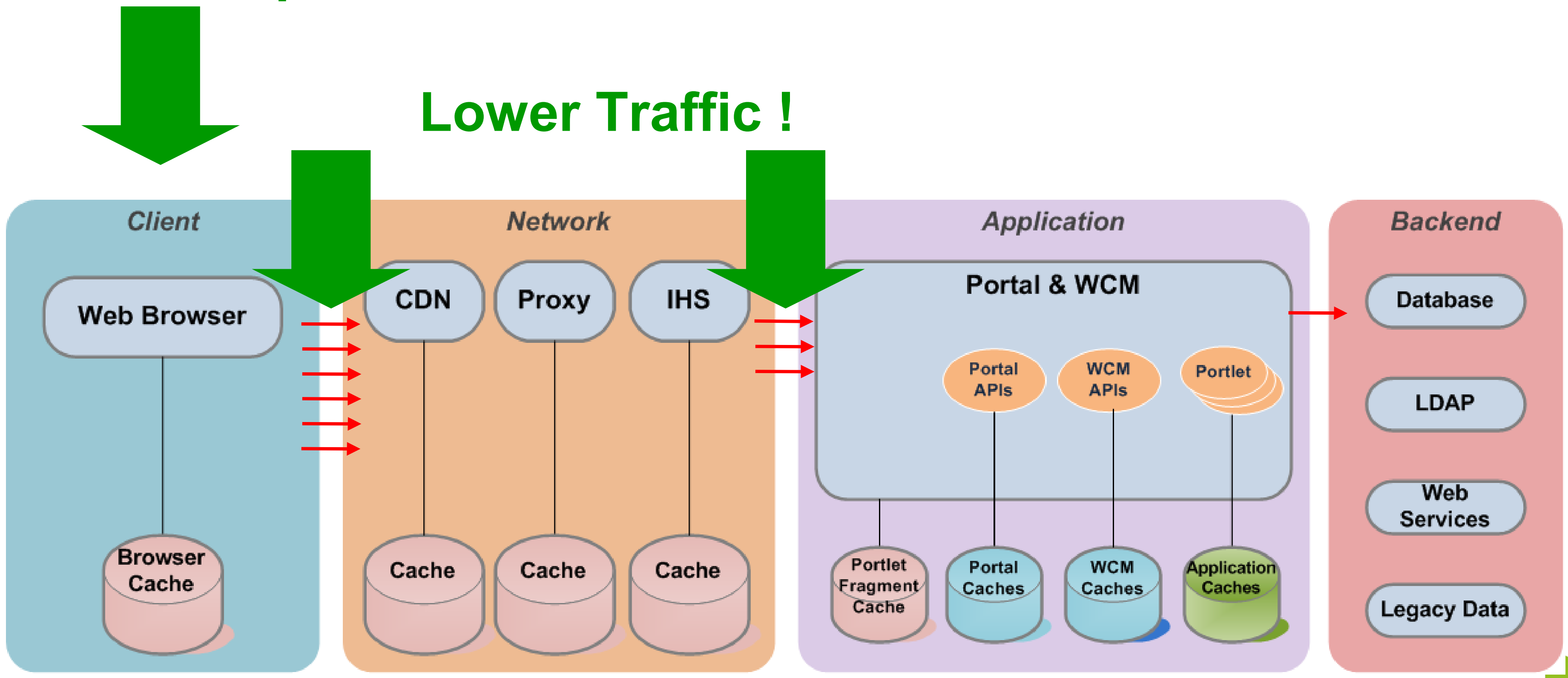
Browser & Network Caching

- How Caching happens is defined in the HTTP 1.1 specification
  - Cache-Control **header:**
    - » max-age: cache lifetime, in seconds
    - » private: cachable by a non-shared cache (i.e. browser)
    - » public: cachable by shared caches (i.e. browsers & proxies)
  - Last-Modified **header**
- Cache-Control headers are usually set in the application tier
- Caching: best for static resources, but also caches dynamic pages
- Browser:
  - Cache-Control: **public** or **private**
- Network: CDN, Proxy, Http Server
  - Cache-Control: **public only**
  - Big benefit: page loaded by single user ⇒ page cached for all users

Big Benefit of Browser & Network Caching

Better Response Time !

Lower Traffic !



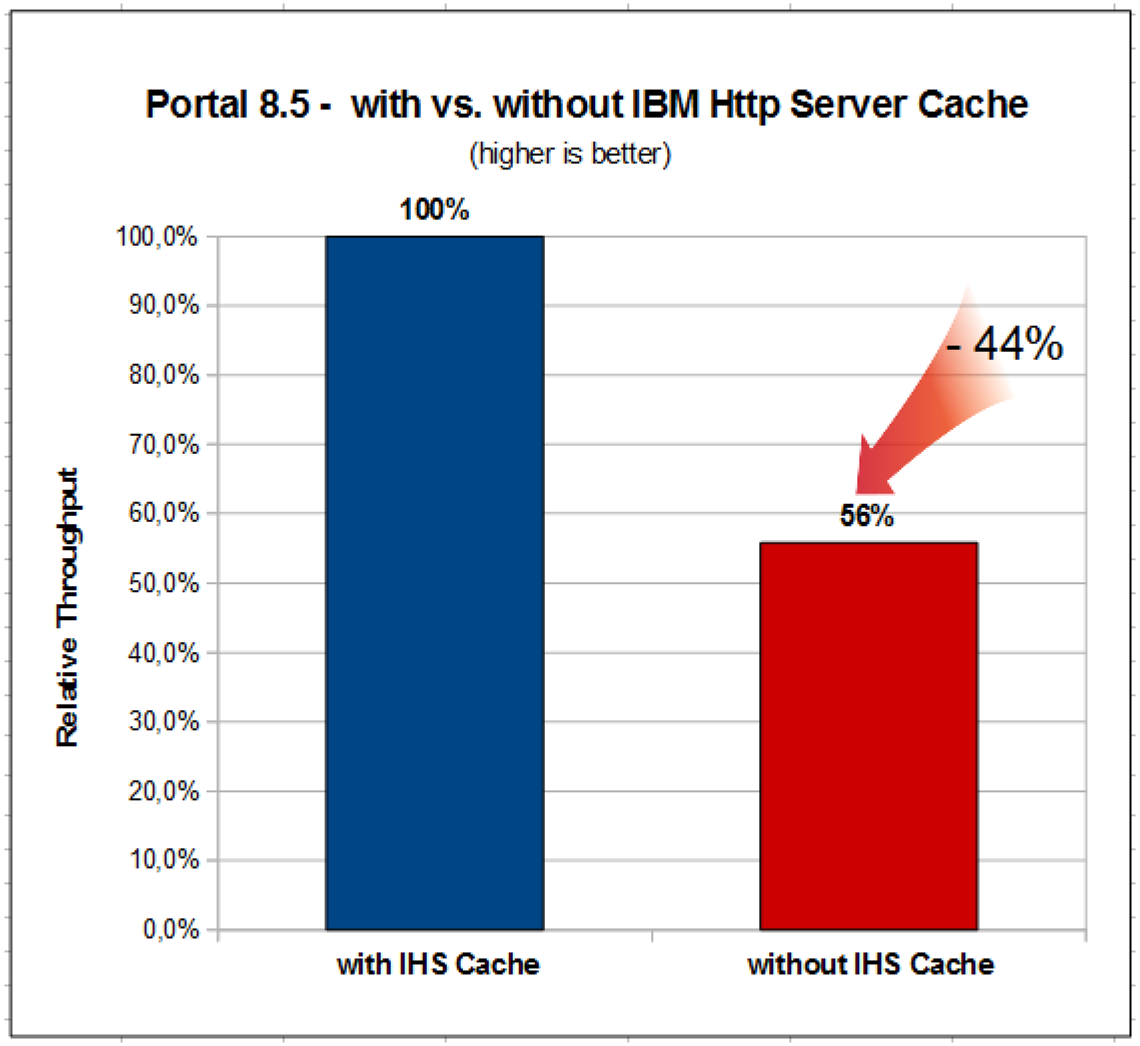
Caching in the IBM Http Server

All IHS deployments should enable caching
- Removes load of serving theme and other static resources from Portal
- Very low overhead

Caching is managed by mod_disk_cache
- Consider using a disk partition just for the cache directory
- Use noatime option in Linux
- Operating system's file cache → memory like performance
  - Assuming enough system memory
- http://httpd.apache.org/docs/current/mod/mod_cache_disk.html

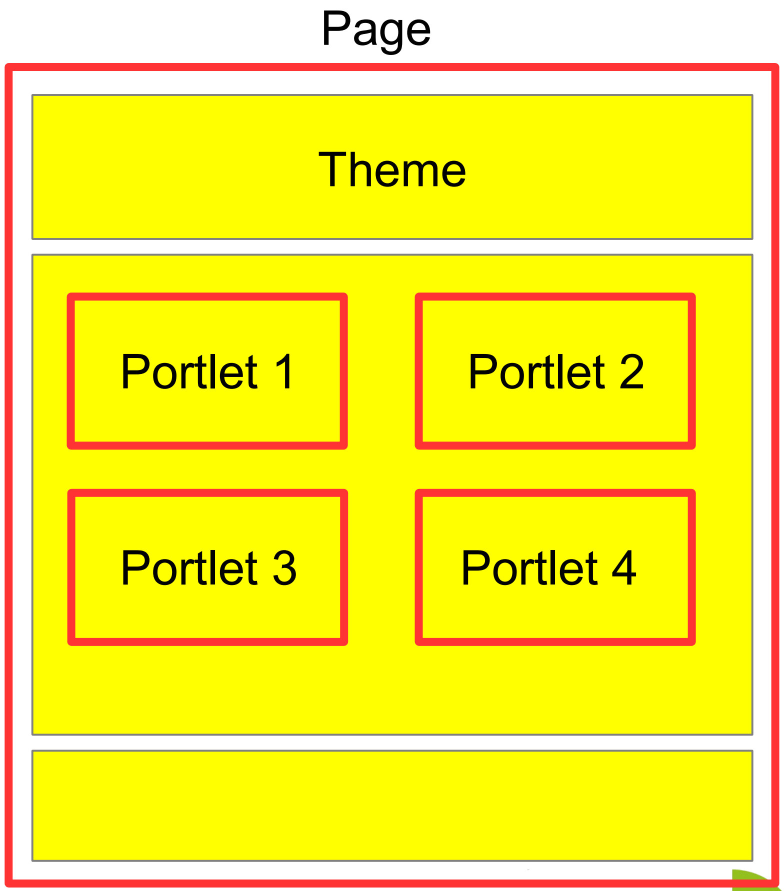
IHS does allow logging of cache control headers, cache hits & misses

Caching in the IBM Http Server



Portal Adaptive Page Caching

- Pages, Themes and Portlets can have cache settings
- What happens if a page has a portlet with conflicting configuration?
  - Lowest common denominator wins
  - Everything on page must be SHARED for public
  - Minimum value of all expirations will be used for max-age
- Since portlets default to no caching, pages will not be cached by just changing the page cache settings



Portal Adaptive Page Caching - Example

- Cache Expiration

Portal page	100 seconds
Theme	40 seconds
Global maximum	50 seconds
First portlet	20 seconds
Second portlet	100 seconds
Resulting value	20 seconds

- `Cache-Control: max-age=20`

- Cache Scope

Portal page	SHARED
Theme	SHARED
Global maximum	SHARED
First portlet	SHARED
Second portlet	NON_SHARED
Resulting value	NON_SHARED

- `Cache-Control: private`

Object Caching in Portal and WCM

Portal and WCM have a couple of internal object caches

Application objects can also be cached

Useful if

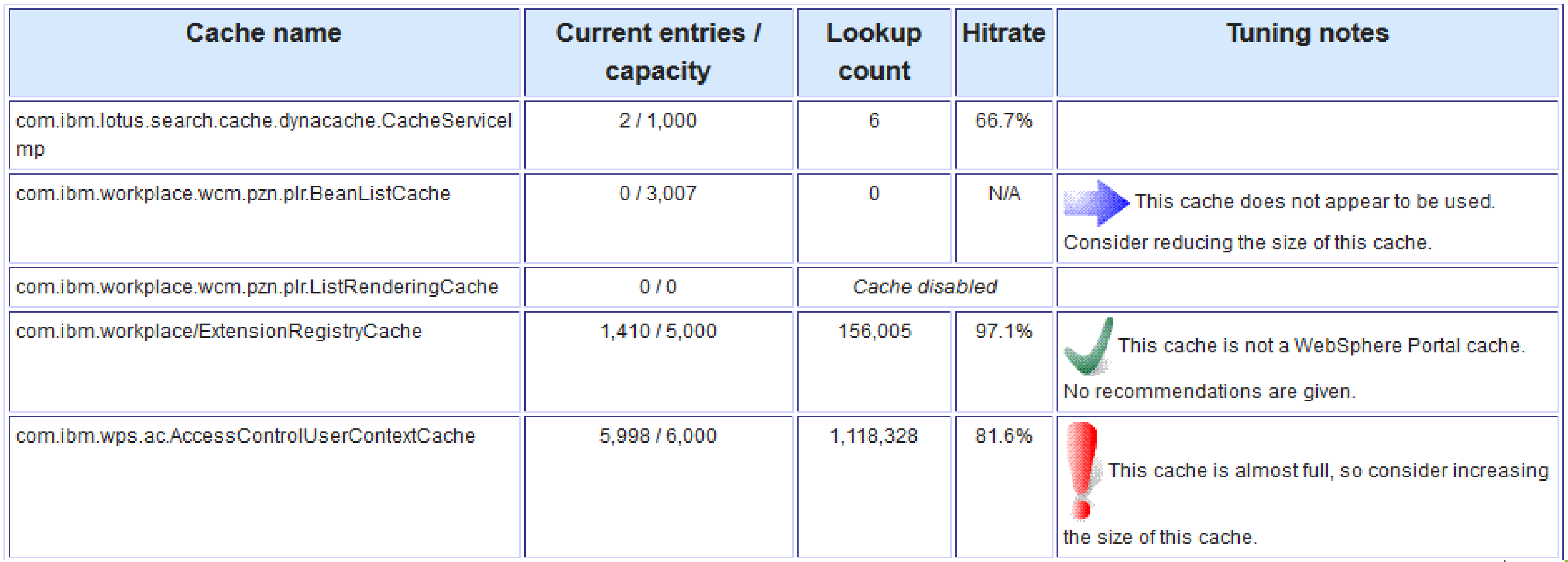
- Pages cannot be cached
- Backend queries are expensive
- Object creation is expensive

Most often stored in memory

- Increases JVM heap requirements

Object Cache Monitoring

Cache Viewer Portlet on Catalog:
https://greenhouse.lotus.com/plugins/plugincatalog.nsf/assetDetails.xsp?action=editD



Cache name	Current entries / capacity	Lookup count	Hitrate	Tuning notes
com.ibm.lotus.search.cache.dynacache.CacheServiceImp	2 / 1,000	6	66.7%	
com.ibm.workplace.wcm.pzn.plr.BeanListCache	0 / 3,007	0	N/A	➡ This cache does not appear to be used. Consider reducing the size of this cache.
com.ibm.workplace.wcm.pzn.plr.ListRenderingCache	0 / 0	*Cache disabled*		
com.ibm.workplace/ExtensionRegistryCache	1,410 / 5,000	156,005	97.1%	✓ This cache is not a WebSphere Portal cache. No recommendations are given.
com.ibm.wps.ac.AccessControlUserContextCache	5,998 / 6,000	1,118,328	81.6%	❗ This cache is almost full, so consider increasing the size of this cache.

Reducing the number of requests

Reducing Number of Requests

Bad: Many pages use too many CSS or JavaScript files
- Each one requires a separate request
- Example: 10 CSS x 60ms = 0.6 seconds latency!

Bad: Pages with too many images to make is visually appealing
- Each one requires a separate request

Combine server side:
- Good: Combine CSS or JavaScript files server side
- Good: Use image spriting: one file containing many images
    - » One request retrieves all the images needed
    - » Client side processing extracts the individual images

Shrinking the page size

Compression

Http & proxy servers can compress content via GZIP

- Compress HTML, CSS and JavaScript
- Do not compress images or other binary data
- In IHS compression is managed by mod_deflate

Benefits:

- In the cache gzip content is stored → the cache content size decreases
- Less data is send over the network

Theme elements (ra:collection URLs) are compressed by Portal by default

Compression

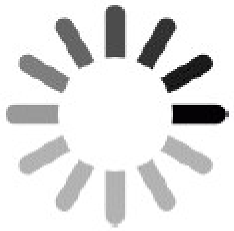
Should always be enabled

- **<u>Savings of 50 – 65% in file size</u>** for HTML, JavaScript & CSS files
  - Reduces network bandwidth
  - Reduces response time
- Can increase CPU requirements on web server
  - Depends on amount of uncached content

On *one* server. Vary & Auth HTTP headers and some security configurations may cause intermediate servers to uncompress and recompress HTTP responses

- Pick a server to perform compression and ensure upstream servers are not doing additional compression
- Firewalls, load balancers or security servers are often good candidates

AJAX

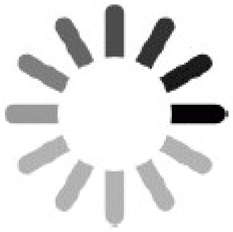


AJAX

AJAX stands for Asynchronous JavaScript and XML

Good: The AJAX pattern applied to heavy weight portlets or portlets with a slow backend yields significant performance enhancements

- Benefits:
  - » The perceived end user experience for the first page load improves
  - » Updates do not require a full page refresh, only the component that changes



Bad: Implemented every single portlet using AJAX

- Previous intranet home page loaded in 2 seconds
- New AJAX implementation
  - » 15 second page loads in single user scenario
  - » 30+ seconds under load
- Reason: higher request traffic to the server, higher cpu consumption on the client

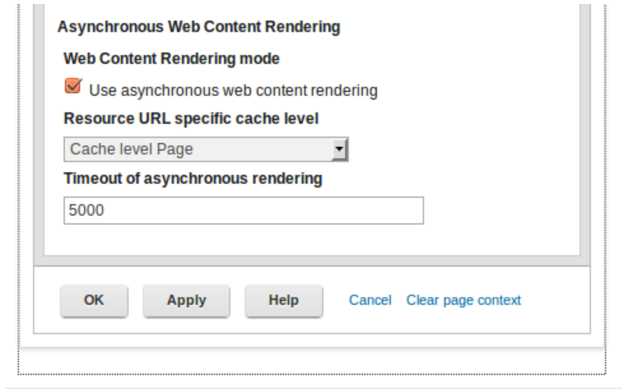
AJAX is a good idea when

- A portlet will take longer than a second to render
- The content of the portlet is not common between users
  - Portlet takes longer than a second to render
  - Content may change often and is minimally cacheable
- A portlet will likely be interacted with
  - Likely in place refreshes save entire page refreshes
  - Better user experience
  - Better performance
- When a portlet connects to a system which may be unreliable
  - Prevents portal from hanging if system fails
  - Lets user interact with remainder of portal

Asynchronous Web Content Rendering

New: Asynchronous Web Content Rendering will be introduced with WebSphere Portal Version 8.5 CF06

- Asynchronous Web Content Rendering is based on the "AJAX" technology
- A page editor can switch individual web content viewer portlets to use „asynchronous render mode" by setting a new portlet preference in Edit Shared Settings/Configure section



Asynchronous Web Content Rendering

Web Content Rendering mode

[✓] Use asynchronous web content rendering

Resource URL specific cache level

Cache level Page

Timeout of asynchronous rendering

5000

Do you have a Performance Problem ?

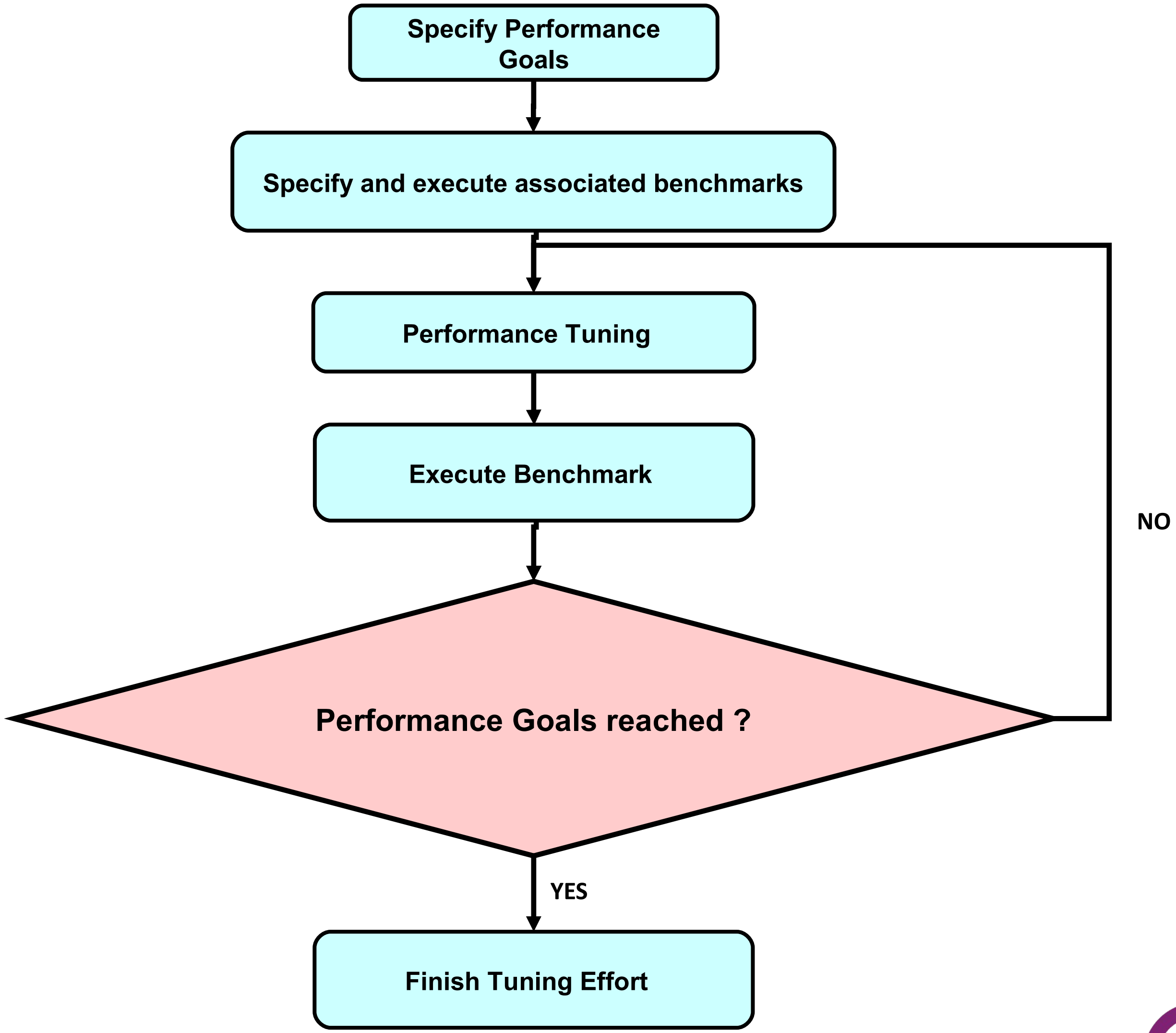
- Performance Tuning
- Performance Troubleshooting
- Performance Tools

Performance Tuning

- **WebSphere Portal Performance Tuning Guide** is always a good starting point - consider all resources which could cause your system to perform poor
- WebSphere Portal needs to be performance tuned for each specific **<u>customer environment</u>**:
  - HW, Operating System, Network, Database, LDAP, Http Server, Proxy, Load Balancer, Backend Service
- WebSphere Portal needs to be performance tuned for each specific **<u>customer application</u>**:
  - Application Complexity, Theme, Caching, JVM Tuning, WebSphere Tuning
- A ConfigEngine **tuning task** was introduced with WP 8.0.0.1 CF06

http://www-01.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/install/w

Performance Tuning - Workflow


Specify Performance Goals
Specify and execute associated benchmarks
Performance Tuning
Execute Benchmark
Performance Goals reached ?
NO
YES
Finish Tuning Effort

Do you have a Performance Problem ?

- Did you apply the recommended fixes ?
- Have you taken a look at the tuning guide ?
- <u>If performance goals are not met, something is the performance bottleneck.</u> This performance bottleneck must be eliminated !
- In order to scale, portal should be the bottleneck, not LDAP, DB2 or other backend systems
- You <u>need problem determination tools</u> which assist you in quickly troubleshooting performance problems or gathering diagnostic data
- **WebSphere Portal Performance Troubleshooting Guide** is always a good starting point
- You need help – engage the IBM Service Team

Performance Tools

Benchmark Tool
- Run multi user load tests to identify the performance of your application
- We use: Rational Performance Tester

Browser Analysis Tool
- Measure response time
- View cache settings, compression, timeline, download size
- We use: HttpWatch, Fiddler, Firebug

Performance Profiling
- Otimize the code path length of your application
- We use: Performance Inspector, YourKit

Problem Determination Tools
- Analyze memory, threads, locks, etc.
- We use: IBM Support Assistant

System / Application / Cache – Monitoring
- We use: nmon, WAS Admin Console, Cache Viewer Portlet

Questions ?

Backup

Newer Hardware

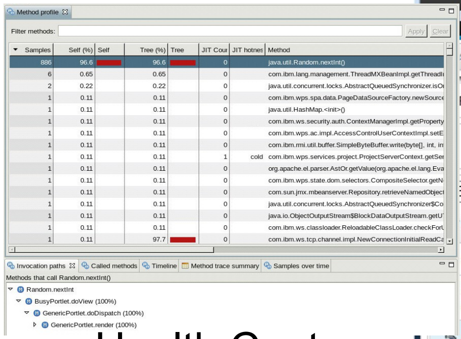
- Newer hardware is much faster
- ~2008 Intel → 2010 (Core i7 / Nehalem) => **2x capacity**
  - » Much more memory bandwidth => faster garbage collection
  - » 3 GHz in 2008 ≠ 3 GHz in 2010 ≠ 3 GHz in 2014
- Power 5 → Power 7 => **2x capacity**
- z10 → z196 => **30%+ throughput**
- Continued improvements:
  - » Intel Sandy Bridge is **~10% faster** than Westmere; Ivy Bridge is another ~20% gain
  - » Power 7+ is **10 – 15% faster** than Power 7

IBM Support Assistant

http://www-01.ibm.com/software/support/isa/download.html

provides rich troubleshooting tools in a modular way

CPU



Method profile

Filter methods: Apply Clear

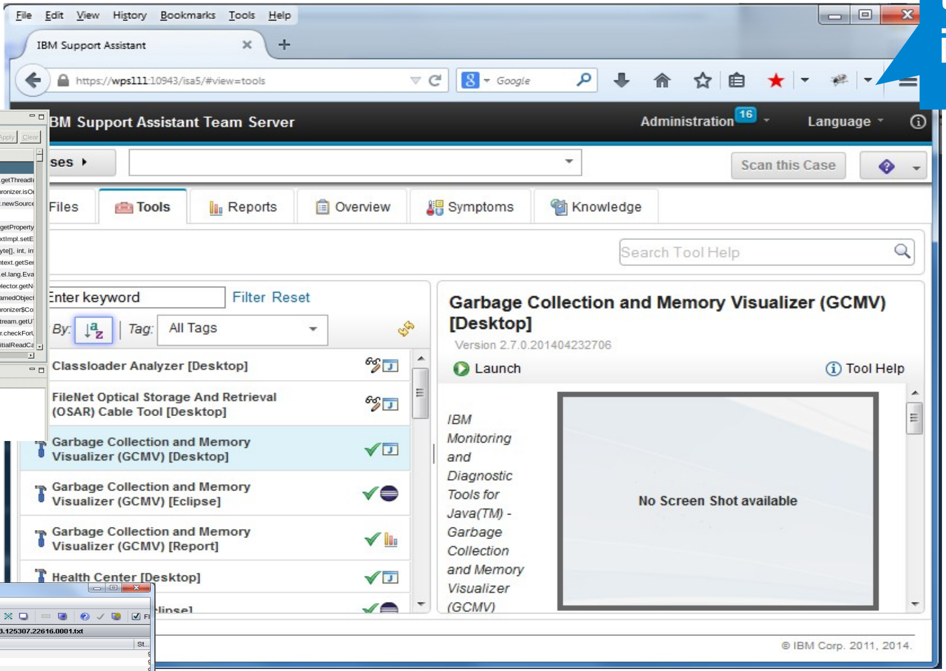
Samples	Self (%)	Self	Tree (%)	Tree	JIT Cour	JIT hotnes	Method
886	96.6	🟥	96.6	🟥	0		java.util.Random.nextInt()
6	0.65		0.65		0		com.ibm.lang.management.ThreadMXBeanImpl.getThreadI
2	0.22		0.22		0		java.util.concurrent.locks.AbstractQueuedSynchronizer.isO
1	0.11		0.11		0		com.ibm.wps.spa.data.PageDataSourceFactory.newSource
1	0.11		0.11		0		java.util.HashMap.<init>()
1	0.11		0.11		0		com.ibm.ws.security.auth.ContextManagerImpl.getProperty
1	0.11		0.11		0		com.ibm.wps.ac.impl.AccessControlUserContextImpl.setE
1	0.11		0.11		0		com.ibm.rmi.util.buffer.SimpleByteBuffer.write(byte[], int, in
1	0.11		0.11		1	cold	com.ibm.wps.services.project.ProjectServerContext.getSe
1	0.11		0.11		0		org.apache.el.parser.AstOr.getValue(org.apache.el.lang.Eva
1	0.11		0.11		0		com.ibm.wps.state.dom.selectors.CompositeSelector.getN
1	0.11		0.11		0		com.sun.jmx.mbeanserver.Repository.retrieveNamedObject
1	0.11		0.11		0		java.util.concurrent.locks.AbstractQueuedSynchronizer$Co
1	0.11		0.11		0		java.io.ObjectOutputStream$BlockDataOutputStream.getU
1	0.11		0.11		0		com.ibm.ws.classloader.ReloadableClassLoader.checkForU
1	0.11		97.7	🟥	0		com.ibm.ws.tcp.channel.impl.NewConnectionInitialReadCa

Invocation paths | Called methods | Timeline | Method trace summary | Samples over time

Methods that call Random.nextInt()

```
Random.nextInt
  BusyPortlet.doView (100%)
    GenericPortlet.doDispatch (100%)
      GenericPortlet.render (100%)
```

Health Center



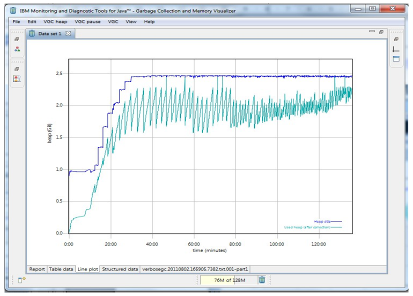
Classloader Analyzer [Desktop]
FileNet Optical Storage And Retrieval (OSAR) Cable Tool [Desktop]
Garbage Collection and Memory Visualizer (GCMV) [Desktop]
Garbage Collection and Memory Visualizer (GCMV) [Eclipse]
Garbage Collection and Memory Visualizer (GCMV) [Report]
Health Center [Desktop]

Garbage Collection and Memory Visualizer (GCMV) [Desktop]
Version 2.7.0.201404232706
Launch
Tool Help

IBM Monitoring and Diagnostic Tools for Java(TM) - Garbage Collection and Memory Visualizer (GCMV)

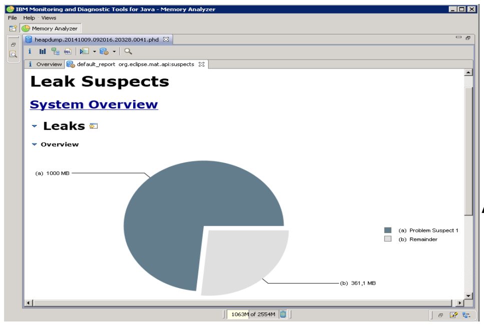
No Screen Shot available

© IBM Corp. 2011, 2014.

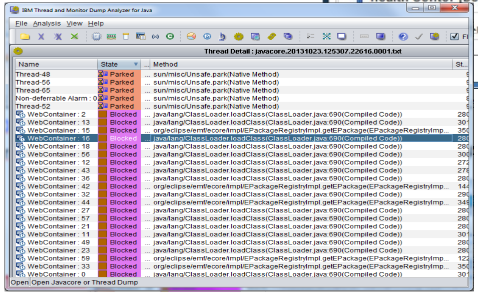


Garbage Collection and Memory Visualizer

Memory



Memory Analyzer



IBM Thread and Monitor Dump Analyzer for Java

File Analysis View Help

Thread Detail : javacore.20131023.125307.22616.0001.txt

Name	State	...	Method	St...
Thread-48	Parked	...	sun/misc/Unsafe.park(Native Method)	[illegible]
Thread-56	Parked	...	sun/misc/Unsafe.park(Native Method)	[illegible]
Thread-65	Parked	...	sun/misc/Unsafe.park(Native Method)	[illegible]
Non-deferrable Alarm : 0	Parked	...	sun/misc/Unsafe.park(Native Method)	[illegible]
Thread-52	Parked	...	sun/misc/Unsafe.park(Native Method)	[illegible]
WebContainer : 2	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 13	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	301
WebContainer : 15	Blocked	...	org/eclipse/emf/ecore/impl/EPackageRegistryImpl.getEPackage(EPackageRegistryImp...	350
WebContainer : 16	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 18	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 56	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	300
WebContainer : 12	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	272
WebContainer : 43	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	278
WebContainer : 36	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 42	Blocked	...	org/eclipse/emf/ecore/impl/EPackageRegistryImpl.getEPackage(EPackageRegistryImp...	144
WebContainer : 32	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	290
WebContainer : 44	Blocked	...	org/eclipse/emf/ecore/impl/EPackageRegistryImpl.getEPackage(EPackageRegistryImp...	349
WebContainer : 27	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 57	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 21	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 11	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	301
WebContainer : 49	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 23	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	280
WebContainer : 59	Blocked	...	org/eclipse/emf/ecore/impl/EPackageRegistryImpl.getEPackage(EPackageRegistryImp...	122
WebContainer : 33	Blocked	...	org/eclipse/emf/ecore/impl/EPackageRegistryImpl.getEPackage(EPackageRegistryImp...	350
WebContainer : 0	Blocked	...	java/lang/ClassLoader.loadClass(ClassLoader.java:690(Compiled Code))	301

Open | Open Javacore or Thread Dump

IBM Thread and Monitor Dump Analyzer for Java (TMDA)

Lock Contention