Raj Alexander
Travelers, Lead Architect
ralexan2@travelers.com

Roman Kharkovski
IBM, Executive IT Specialist
kharkovski@us.ibm.com

Competitive Comparison: WebSphere Application Server vs. Tomcat, JBoss and WebLogic

Session # PEJ-3714

February 23, 2016

Gartner middleware software market report

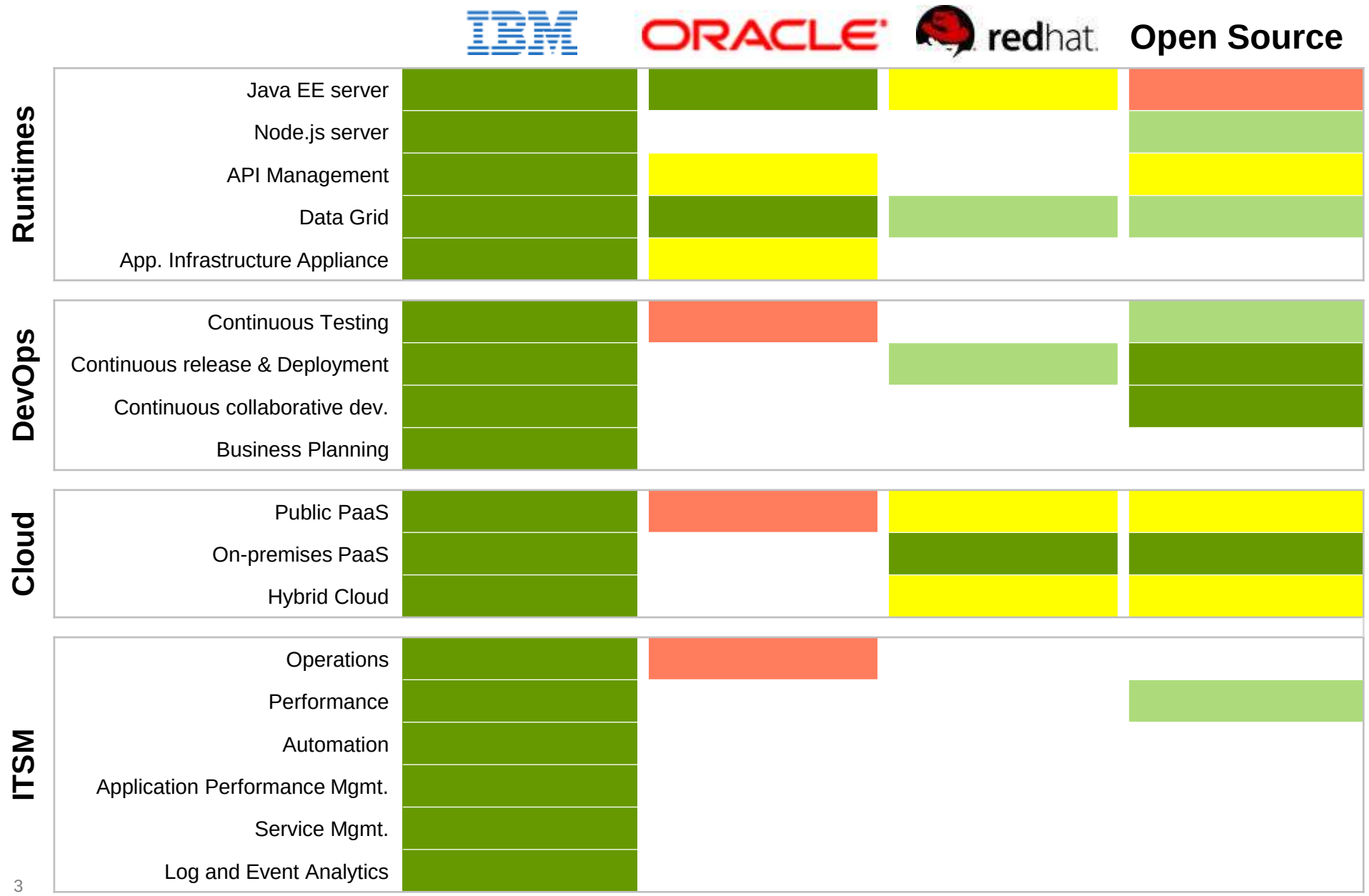
According to Gartner, IBM holds #1 position for the past 13 years

	2014 ($B)	YTY growth	IBM rank	IBM share	IBM growth	ORACLE rank	ORACLE share	ORACLE growth	redhat rank	redhat share	redhat growth
BPM	2.5	6.4%	# 1	28.5%	1.4%	# 3	7.9%	2.3%	# 22	0.2%	-
ESB	2.77	10.4%	# 1	29.0%	7.5%	# 2	21.6%	2.6%	# 16	0.4%	36.6%
MOM	1.35	8%	# 1	75.0%	6.5%	-	-	-	# 9	0.3%	38.9%
MFT Suites	0.7	15.6%	# 1	31.3%	11.4%	# 13	0.9%	-	-	-	-
TP Monitors	1.86	2.8%	# 1	85.2%	4.%	# 2	10.0%	-5.9%	-	-	-
Appliances AIM	0.8	15.2	# 1	18.5%	7.1%	# 3	0.8%	1.5%	-	-	-
B2B	0.9	3.8%	# 1	21.2%	0.7%	-	-	-	-	-	-
App Servers	5.4	14.6%	# 2	28.5%	10.2%	# 1	33.5%	1.6%	# 5	2.6%	36.6%
Portals	1.8	1.4%	# 2	24.4%	-8.5%	# 3	21.1%	2.4%	# 10	0.8%	30.2%
Svc Governance	6.2	17.5%	# 1	11.2%	6.1%	# 2	10.6%	1.5%	-	-	-
Other AIM	4.9	5.1%	# 2	28.2%	30.0%	# 17	0.6%	-17%	# 28	0.2%	32.5%

Source: Gartner, Market Share Analysis: Enterprise Software Market Share, Worldwide. Published March , 2015

Excellent
Good
Limited
Very Limited
No support

Application Infrastructure Platform

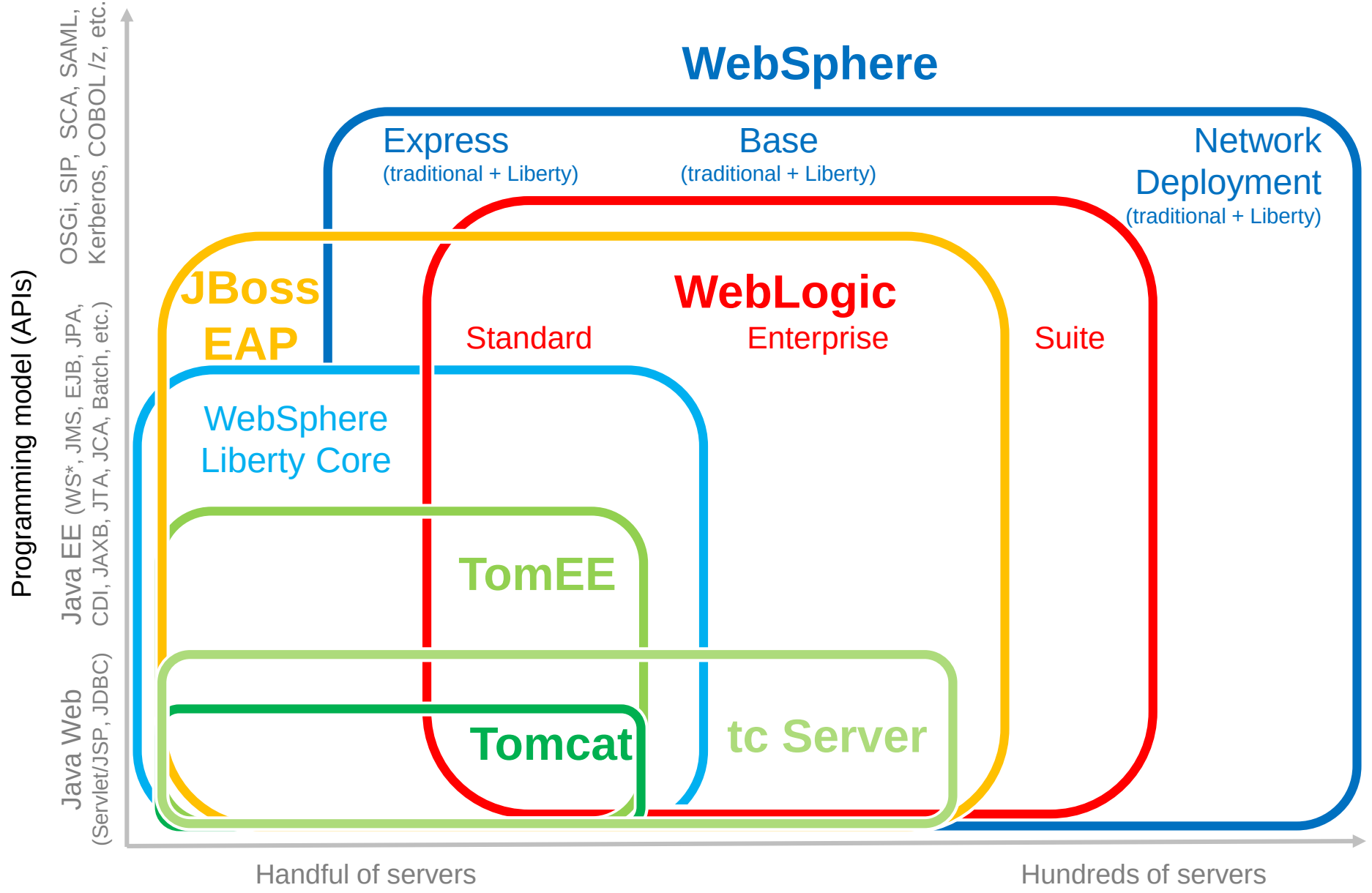


Excellent
Good
Limited
Very Limited
No support

Application Infrastructure Platform

		IBM	ORACLE	redhat	Open Source
Runtimes	Java EE server	WebSphere App. Server	WebLogic Server	JBoss EAP	Tomcat / WildFly
	Node.js server	StrongLoop	n/a	n/a	Node.js
	API Mgmt.	API Connect	Axway (OEM)	n/a	WSO2 Api Manager
	Data Grid	WXS	Coherence	JBoss Data Grid	Infinispan, etc…..
	App. Infrastructure appliance	PureApplication System	Exalogic	n/a	n/a
DevOps	Continuous testing	Rational Test Workbench	App. Testing Suite	n/a	Selenium, JMeter, etc.
	Continuous release & deployment	Urban Code Deploy	n/a	Ansible (OEM)	Jenkins, maven, chef, puppet
	Continuous collaborative Dev.	Rational Team Concert	n/a	n/a	Git, Subversion, etc.
	Business Planning	Rational Doors NG	n/a	n/a	n/a
Cloud	Public PaaS	Bluemix	Oracle Cloud	OpenShift Online	Cloud Foundry, OpenShift
	On-premises PaaS	Bluemix Local	n/a	OpenShift Enterprise	Cloud Foundry, OpenShift
	Hybrid Cloud	Cloud Orchestrator	n/a	CloudForms & Ansible (OEM)	Cloud Foundry, CloudForms
ITSM	Operations	Netcool / Omnibus	Enterprise Manager	n/a	n/a
	Performance	Transaction / System	n/a	n/a	Nagios and Dérivatives
	Automation	WorkLoad and Provision	n/a	n/a	n/a
	Application Performance Mgmt.	Deep Dive Monitoring	n/a	n/a	n/a
	Service Mgmt.	Control Desk	n/a	n/a	n/a
	Log and event analytics	Log and Predictive	n/a	n/a	n/a

Java servers



~~Minimize License and support cost (TCA)?~~

Cloud support?

Time to market?

Management and administration?

Monitoring and diagnostic tools?

Standards support and programming model?

Documentation support practices?

Performance and scalability?

High availability and reliability?

Minimize TCO

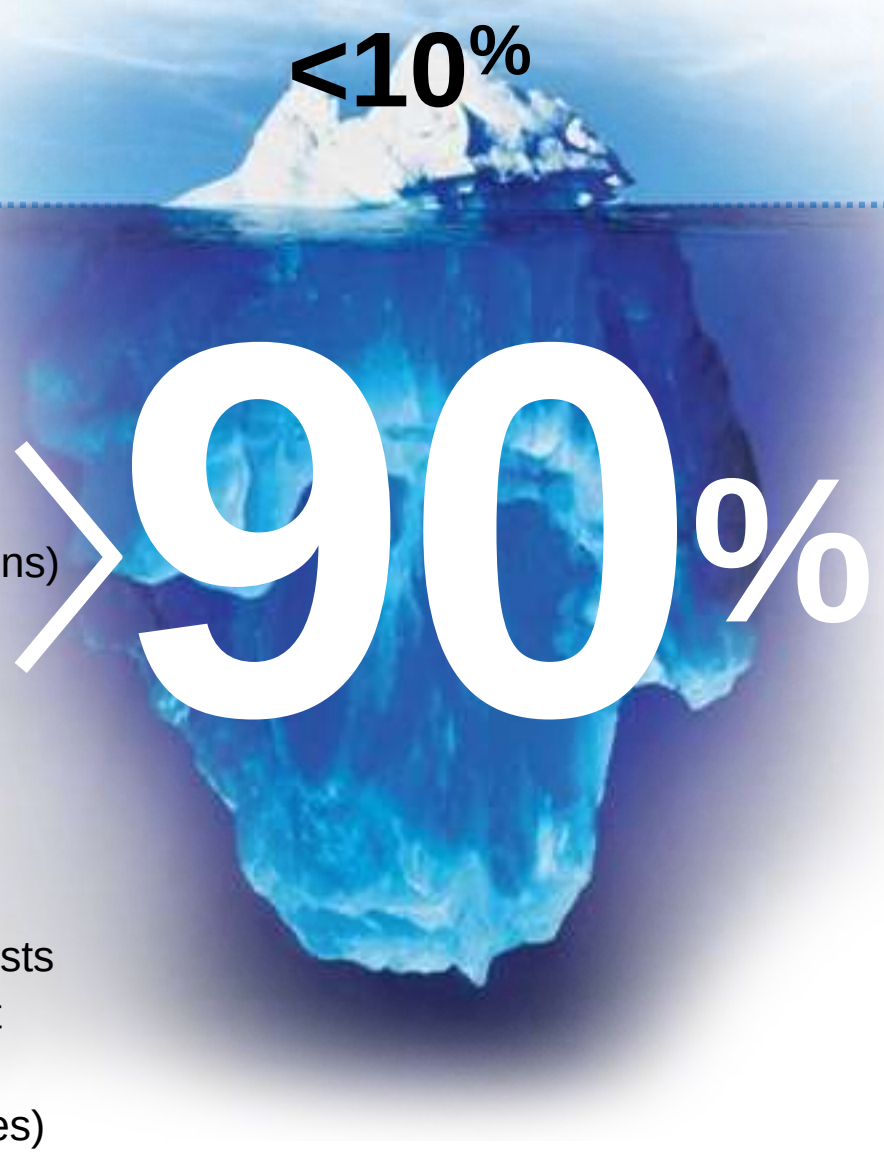
PaaS?

OS and DB support?

User and administrative security?

TCA < TCO

- Software license & subscription costs[1]



- Hardware and networking costs
- Downtime costs (planned and unplanned)
- Upgrades cost
- SLA penalties
- Deployment cost
- Operational support cost (day to day operations)
- Performance costs
- Cost of selection of the vendor software
- Requirements analysis cost
- Developer, admin and end-user training cost
- Application design and development costs
- Cost of integration with other systems
- Quality, user acceptance and other testing costs
- Application enhancements and bug fixes cost
- Replacement costs
- Cost of other risks (including security breaches)

(1) Source: http://bit.ly/1yH5oKZ

Free like in beer

NO CHARGE WebSphere Developer Tools for Eclipse

NO CHARGE WAS for Developers & Liberty Profile

- Free runtime for the developer desktop/laptop
- Free support for those who have production licenses

NO CHARGE production runtime – Liberty Core for ISVs

- ISV's customers can run the app on Liberty Core free of charge

NO CHARGE production – Liberty for up to 1GB on BlueMix

- Non-stop running Liberty instance for any purpose

NO CHARGE production runtime – Liberty for up to 2 GB

- Any number of instances, while sum total Java heap is <=2GB

Flexible licensing options from IBM

More flexibility

- JBoss, tc Server do not have socket, or per user pricing
- tc Server, WebLogic and JBoss do not have On-Demand per day pricing

Lower cost

- WAS user based licenses are order of magnitude less than the competition
- WAS socket based pricing is significantly lower than JBoss core based subscription
- WAS cloud prices are lower than the competition

Cloud vs. on-premises

- Any of the licenses shown below can be used in the cloud (BYOL) or on-prem, except for the 1 hour cloud-only license

	Perpetual licenses				Pay as you go					
	Core (PVU)	*Socket*	*20 users*	*Unlimited license*	*1 hour (cloud)*	*1 day (Power)*	*1 month (PVU)*	*1 month (socket)*	*1 year (socket)*	*1 year (PVU)*
Liberty Bluemix	n/a	n/a	n/a		$0.07	n/a	n/a	n/a	n/a	n/a
WAS Express	$ 29.00	n/a	$ 730		$0.21	$ 25.42	$ 1.21	n/a	n/a	$11.75
WAS Liberty Core	$ 29.00	n/a	$ 730	Contact	$0.21	$ 25.50	$ 1.21	n/a	n/a	$11.75
WAS Base	$ 58.75	$14,900	n/a	IBM	$0.30	$ 51.09	$ 2.45	$ 621	$5,960	$23.50
WAS ND	$220.00	n/a	n/a		$0.70	$191.67	$ 9.17	n/a	n/a	$91.00

sw+hw+lots of "free" labor - courtesy of Bluemix PaaS :-)

 More about pricing: http://whywebsphere.com/2013/09/26/software-costs

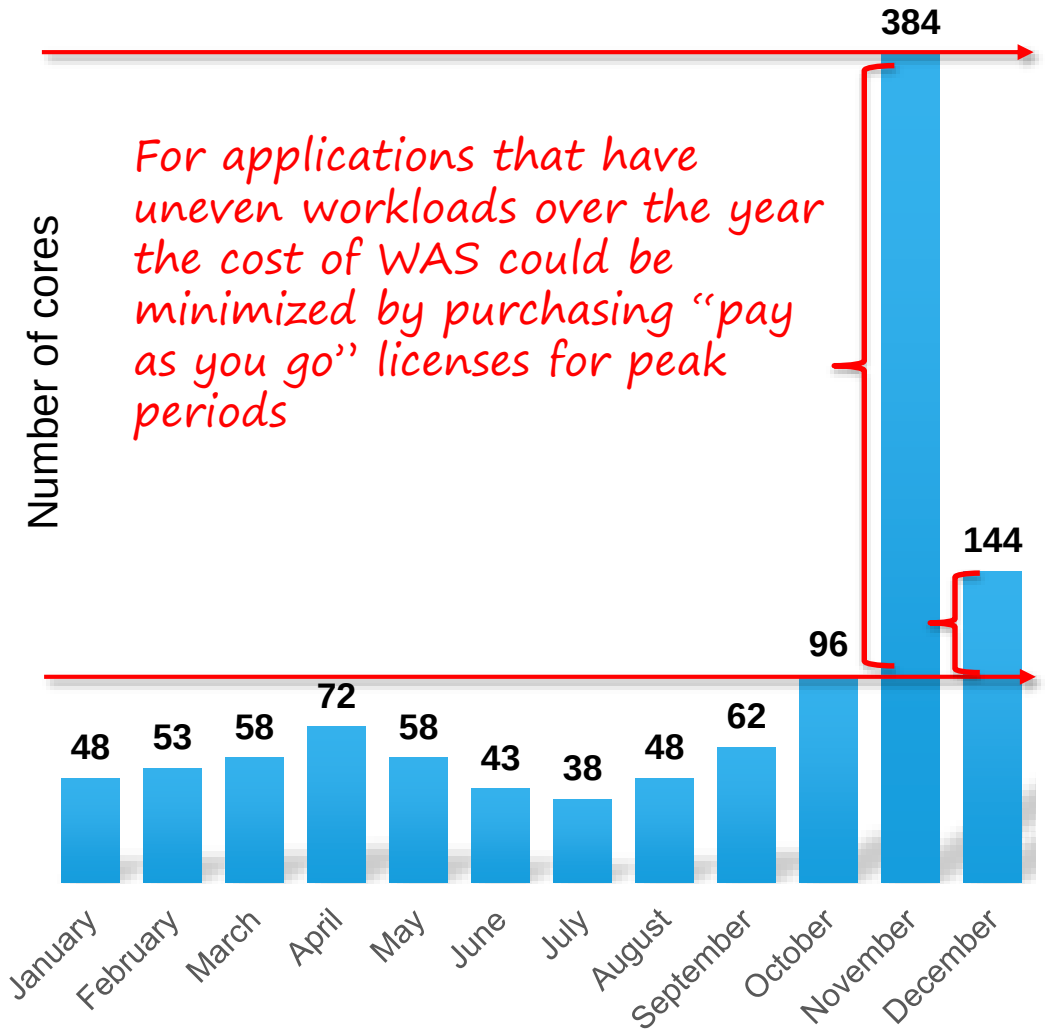
License cost of add-ons for JBoss and Tomcat

	WAS	WAS ND	JBoss EAP or Tomcat
Management and monitoring	Included	Included	Included (in “managed” bundles)
JON configuration DBMS	n/a	n/a	$6,900 / CPU / year (PostgreSQL)
Hardware for the JON database	n/a	n/a	~ $15,000 + support (3rd party)
Load Balancer	Extra $	Included	~ $20,000 / device + support (3rd party)
Dynamic content caching proxy	Extra $	Included	$2,500 / 16 cores / year (JBoss EWS)
Page fragment & POJO caching	Included	Included	~ $1,000 / server / year (3rd party)
HTTPSession persistence DBMS	Included	Included	$6,900 / CPU / year (PostgreSQL)
LDAP	Included	Included	$9,000 / server / year (3rd party)
JDK	Included	Included	OpenJDK is supported on RHEL $5,000 / core (Oracle JDK)
Troubleshooting tools	Included	Included	$?,000 / year (3rd party)
HTTP Server	Included	Included	$2,500 / 16 cores / year (JBoss EWS)
App Server Hardware	$X	$X	$X + 30% (due to lower performance)

Software costs: http://whywebsphere.com/2013/09/26/software-costs/

Example of the use of monthly term license

Workload distribution example over calendar year (hypothetical)
Servers are 2 sockets, 12 cores each



JBoss subscription licenses: 384=16*24
JBoss 5 year cost = $1.44M

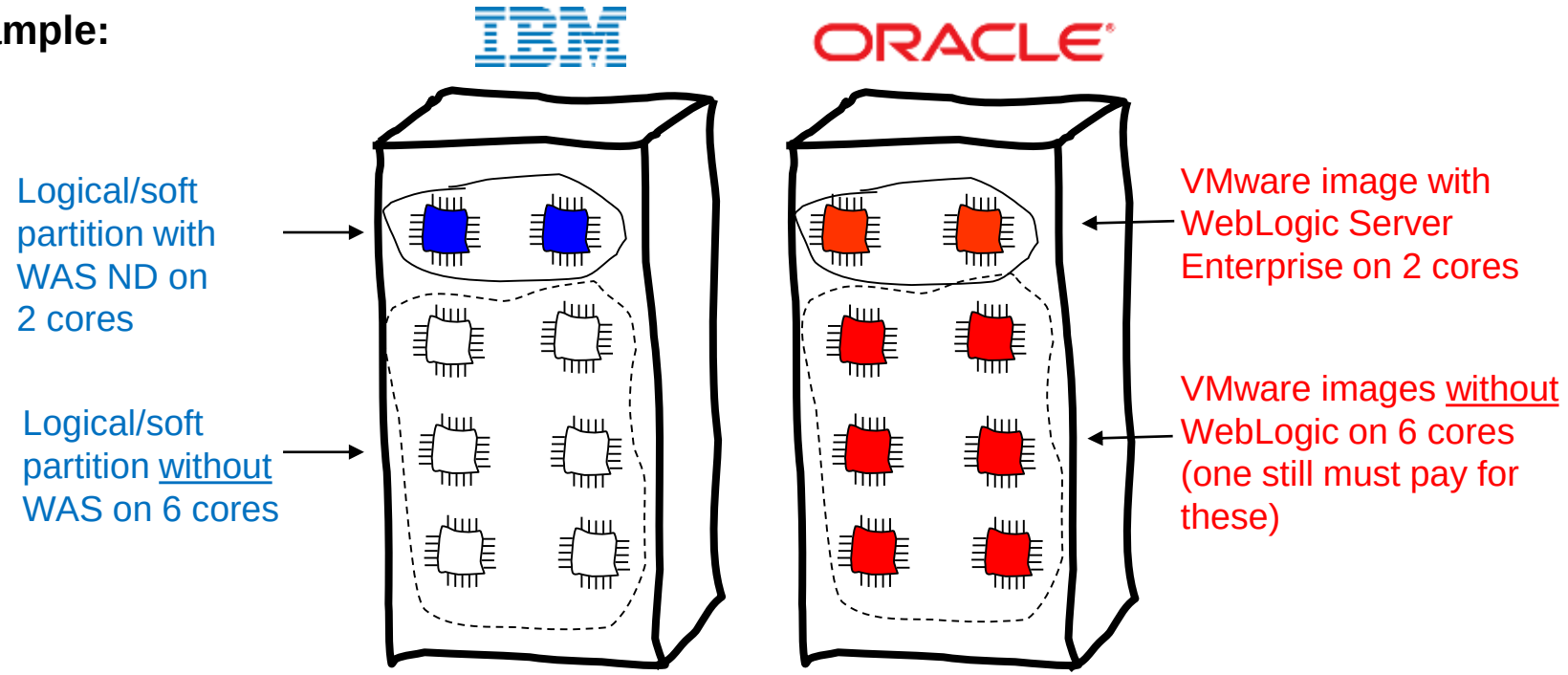
WAS perpetual licenses: 8 sockets
WAS monthly licenses: 336
IBM 5 year cost = $283K

Oracle licensing does not permit soft partitioning

You pay Oracle for all CPUs on a server vs. CPUs that are assigned to the logical VM. Oracle does not allow the use of soft partitioning as a means to determine or limit the number of software licenses required for any given server.

Read detailed analysis here: http://bit.ly/1JZFy8V

Example:



WebSphere AS ND is licensed for 2 cores
License & support cost for 5 years= **$53,928**

WebLogic Server Enterprise is licensed for 8 cores.
License & support cost for 5 years = **$210,000**

Based on publicly available information as of 7/10/2015 comparing Oracle WebLogic Server Enterprise Edition to IBM WebSphere Application Server Network Deployment. Both include maintenance and support for 5 years. IBM: 70 Processor Value Units per core, Oracle: 0.5 processor multiplier, both are on an x86 server, 2 sockets, quad core each.

Virtualization and server partitioning support

	Support[1]		Reduced pricing[2]	
	IBM	Oracle	IBM	Oracle
VMware	Yes	no	Yes	no
IBM z/VM	Yes	no	Yes	no
IBM PR/SM	Yes	no	Yes	no
IBM PowerVM LPAR	Yes	Yes	Yes	Yes/no[3]
Xen	Yes	no	Yes	no
Red Hat KVM	Yes	no	Yes	no
Hyper-V	Yes	Yes	Yes	no
Xen	Yes	no	Yes	no
Oracle VM	Yes	Yes	Yes	Yes/no[4]
Solaris containers	Yes	Yes	Yes	Yes/no[4]

1 - Oracle does not certify nor supports certain 3rd party software hypervisors as shown in the Oracle column (No means no support, Yes means support).
2 - Oracle charges up to full capacity of the servers, regardless of the number of cores used, except for some hypervisors (No means charge for all cores on a server, Yes means charge only for cores assigned to a VM).
3 - Turbocharged cores are not eligible for reduced Oracle pricing on Power7.
4 – Not all configurations of OracleVM and Solaris Containers are eligible for reduced Oracle pricing.
5 - Read more details here: http://whywebsphere.com/2012/02/16/ibm-and-oracle-software-licensing-and-support-in-virtualized-private-cloud-environments/

Oracle charges more for backup and DR

Oracle charges for the main cluster and hot backup

- IBM does too

Oracle charges full license cost for “Warm” backup servers

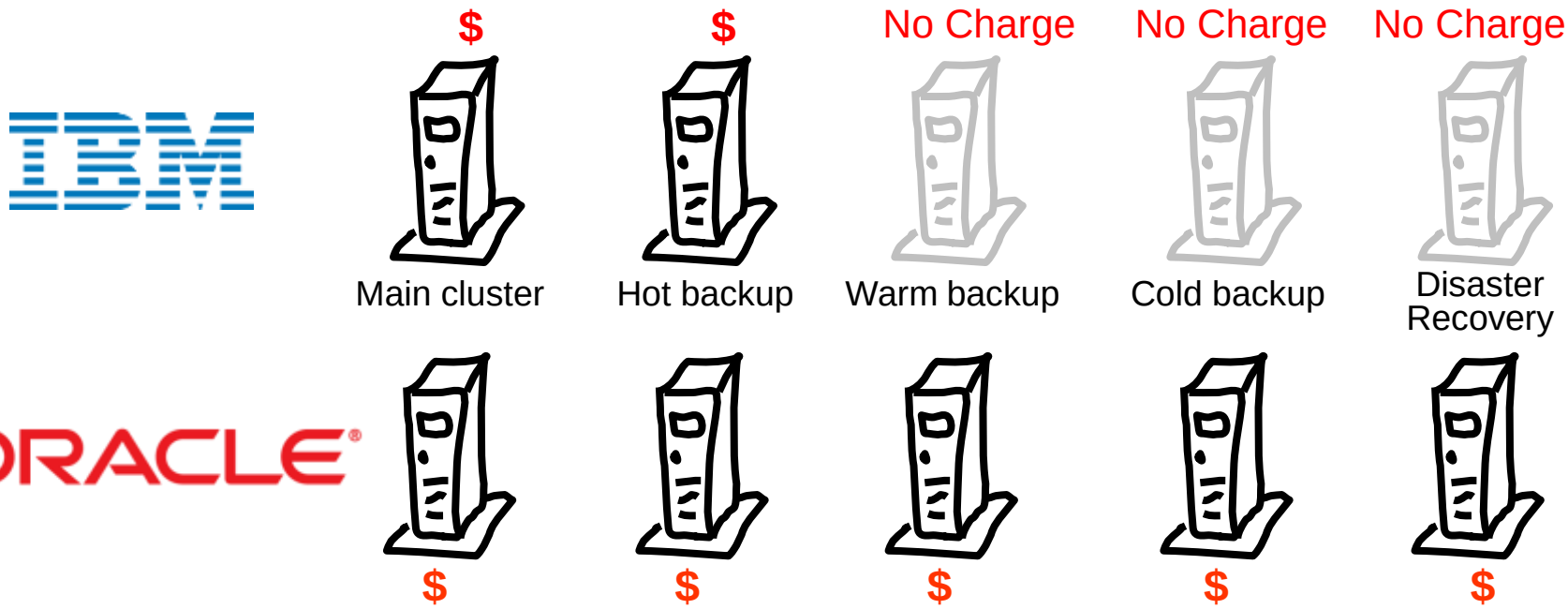
- IBM does not

Oracle charges full license cost for “Cold” backup servers in DR setup

- IBM does not

Oracle charges for “Cold” backup when failover is > 10 days

- IBM does not



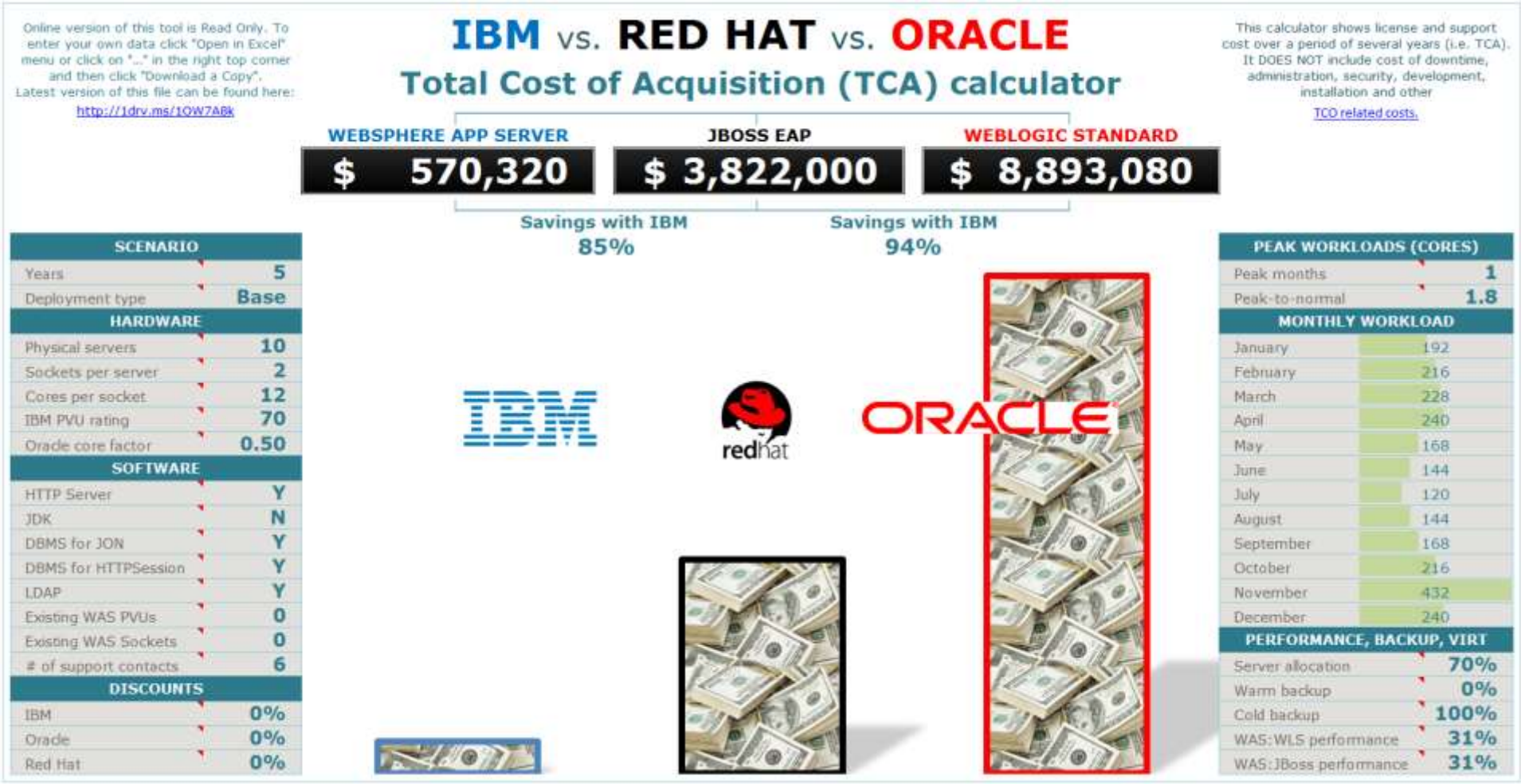
List prices are used for cost comparisons. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

Support costs

IBM			ORACLE
Support includes version upgrades	☑	☑	Support includes version upgrades
First year of support is at no extra charge	☑	☒	First year of support costs additional 22% on top of the license cost
Support is 20% of the license cost	☑	☒	Support is 22% of the license cost
Support cost is a % of the PPA entitled price	☑	☒	Support cost increases at least 4% per year until reaches list price
Critical fixes are provided even without current support	☑	☒	Critical fixes are not provided without current support
IBM software running on 3rd party hypervisors is supported	☑	☒	Oracle software running on 3rd party hypervisors is not supported*

* - More details on virtualization support: http://smarterquestions.org/2012/02/ibm-and-oracle-software-licensing-and-support-in-virtualized-private-cloud-environments

On-prem license cost calculator, blog and video



WAS vs. WLS: http://bit.ly/1JZFy8V

WAS vs. JBoss: http://bit.ly/22aOKkz

TEI of migrating from OSS App Servers to Liberty

https://ibm.box.com/forrester-was-oss

Forrester Consulting interviewed two and surveyed 30 organizations that migrated from OSS app servers to WAS Liberty with the composite organization:

- Is a US-based organization with $5.7 billion in annual revenue and 33,000 employees
- Has 35 applications to migrate from open source Java EE application servers to WAS Liberty, and is standardizing on WAS Liberty for future application development
- Has 300 developers working on the 35 applications
- Has been using WAS Liberty for one year
- Prior to migrating, the organizations used a mix of Liberty and open source Java EE application servers and had difficulty scaling applications and quickly developing and delivering them. The organizations also struggled with high support costs

Benefits

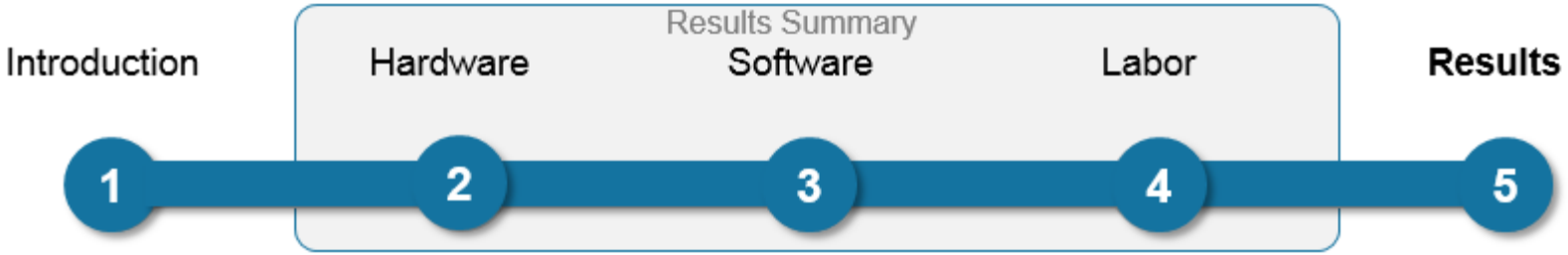
- Deployment time savings – 2 hours per application release
- Startup time savings – 40 minutes per day per developer
- Development productivity – by 12% per developer
- Administration time savings – 3,600 hours per year for per organization
- Support cost savings - $525K per year per organization by year 3 (via consolidation)
- Infrastructure cost savings - $350K per yea per organization
- Other benefits achieved, but not quantified

Composite financial summary

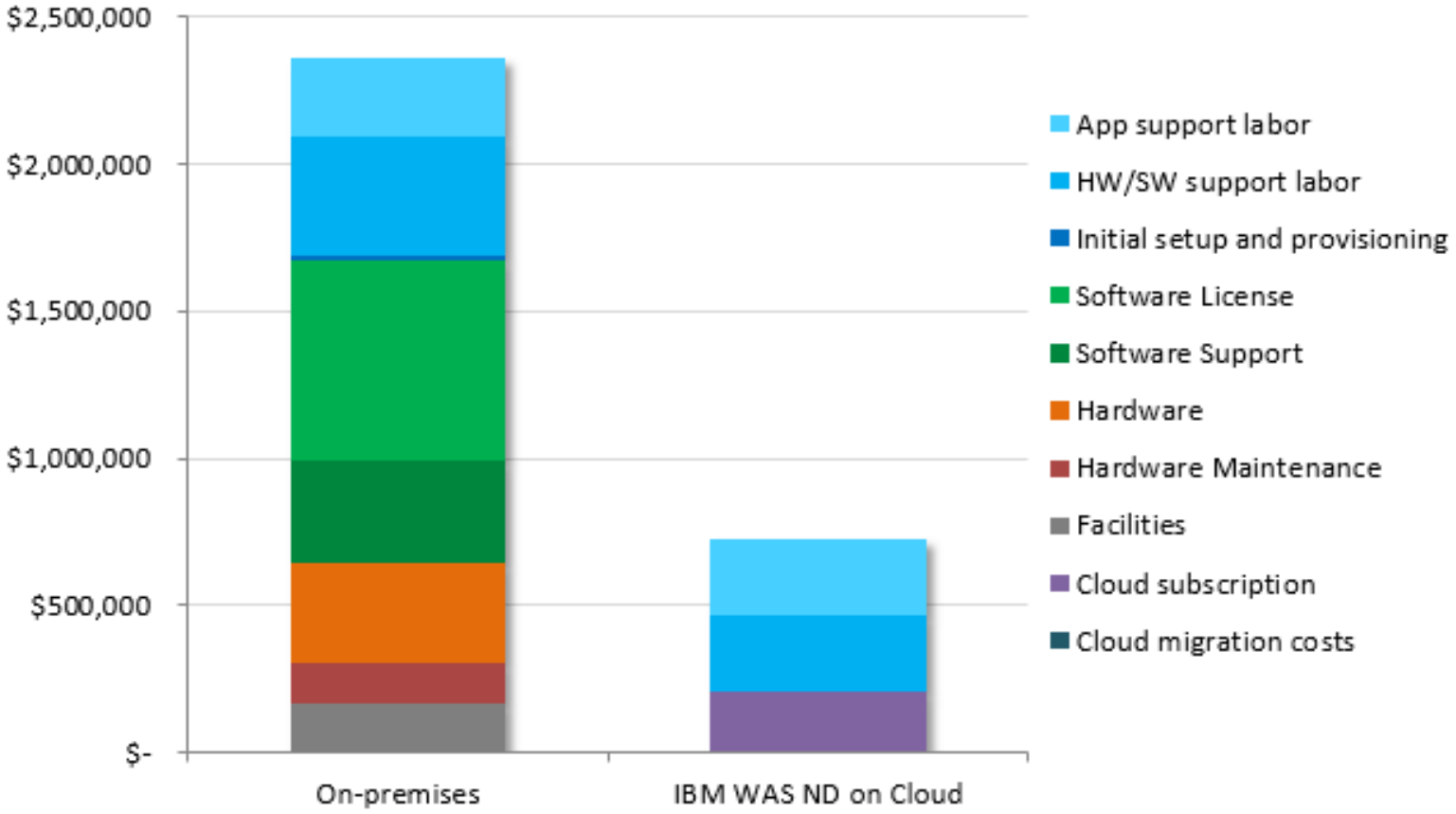
ROI:	NPV:	Payback:	Developer productivity:
122%	**$3,508,142**	**16 months**	**12%+**

WAS on Cloud TCO Calculator

Coming as a web app soon – watch http://IBMadvantage.com for announcement



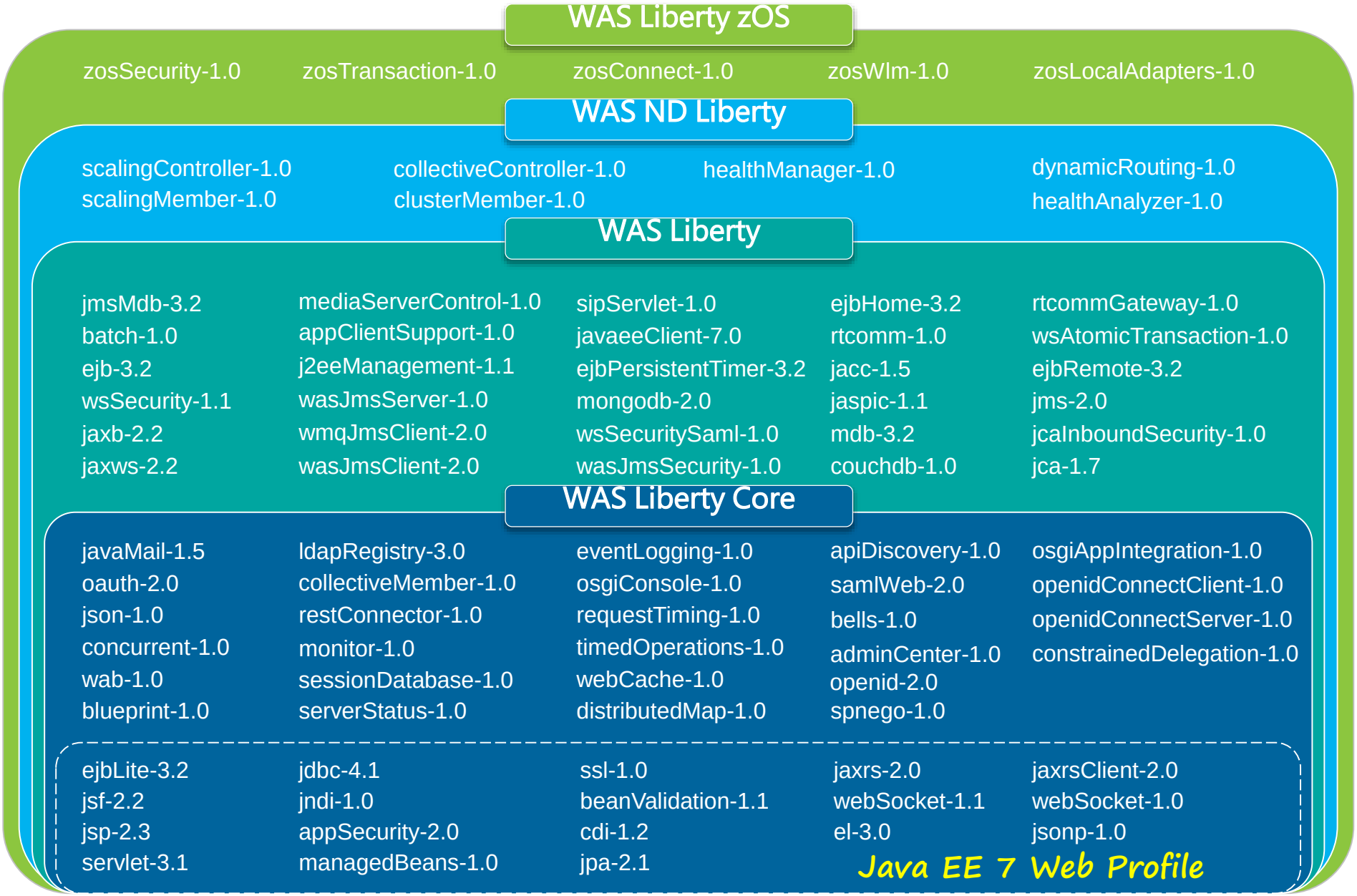
Potential savings over 36 months: $1,637,946



Features map

	Liberty 8.5.5.7	WAS 8.5.5.7	WASND 8.5.5.7	Tomcat 8.0.26	tcServer 3.1.2	JBoss EAP 6.4	WLS 12.2.1
Java EE 7		Beta	Beta			WildFly	
Java EE 6	Web Profile			TomEE			
JDK 1.7 and 1.8		1.7	1.7	3rd party	3rd party	RHEL only	
Performance							
Security							
Transaction management				TomEE			
Messaging Engine				TomEE+			
Caching engine included (IMDG)	WXS	WXS	WXS			Infinispan	
Admin GUI		Single server					
Troubleshooting, profiling, tuning tools							
Admin scripting							
Simple V2V migration and upgrades							
Log analytics	Beta						
Dynamic clustering and auto-scaling	ND						
SLA enforcement and monitoring for requests							
Application versioning							
Automated server health management							
EJB and JMS clustering and failover							
HTTP plugin with WLM and HTTPSession failover							
Dynamic configuration updates (avoid restarts)							
Simple install, lightweight runtime, small footprint							
Cloud (Public / Private / Hybrid)				3rd party			
Appliance	IPAS	IPAS	IPAS				
Free sw included (WLM, HTTPD, LDAP, DBMS)	ND						
Platform certifications (OS, HW, DBMS, Adapters)							
WW support (local language, local hours)							

Liberty v8.5.5.x - **the first** production **Java EE 7**



So you say: “I simply use Tomcat”?

Lack of API support in Tomcat leads to lost development productivity and additional expense for integration and testing

- Most people use a *lot* more than just Tomcat. What about:

VS.

- Once you add all of these (and more), how do you make it work together? (i.e. 3rd party CDI wont work with 4th party JTA, etc.)
- What is the performance of all of the above with all the JAR scanning going on?
- How do you test/manage/secure/debug all of this?
- Are you coding to standards? Are you coding your app or building your own Java EE server?

APIs

Full blog post: http://bit.ly/1heO6Pi

	Liberty	WAS	WebLogic	TomEE	Tomcat	tc Server	JBoss
Java EE 7		Beta					WildFly
Java EE Web profile 7		Beta					WildFly
Java EE Web profile 6							
Java SE 7 and 8		8 in Beta					
Servlet, JSP, JSF					MyFaces JSF		
JDBC							
Java Persistence API (JPA)							
Java Message Service (JMS)	*			TomEE+		RabbitMQ	1.1
Java Transaction API (JTA)							
Bean validation							
Java Management Extensions (JMX)							
Java API for XML Web Services (JAX-WS)	*			TomEE+			
Context Dependency Injection (CDI)							
Java API for RESTful Web Services (JAX-RS)				TomEE+			
OSGi							
EJB lite							
EJB full							
WebRTC							
WebSocket (JSR 356)		Beta					
JSONP							
Oauth							
Concurrency Utilities for Java EE (JSR 236)							WildFly
Batch API (JSR 352)		Beta				Spring Batch	WildFly
JNDI					Read-only	Read-only	
SAML							
WS-Notification							
WS-Policy							
WS-Trust							
WS-ReliableMessaging							
Session Initiation Protocol (SIP)	*						
Portlet API	dev only						
WS-Addressing							
RMI-IIOP	*						
Java Connector Architecture (JCA)	*			TomEE+			
Java Auth. & Authoriz. Service (JAAS)							
JACC and JASPIC							

Excellent
Good
Limited
No support

Java servers from Developer point of view

Excellent
Good
Limited
Very Limited
No support

	Liberty 8.5.5.7	WAS 8.5.5.7	Tomcat 8.0.26	TomEE+ 1.7.2[4]	Jetty 9.3.2	Glass Fish 4.1	Web Logic 12.2.1[3]	WildFly 9.0.1	JBoss EAP 6.4
Server stop+start[5]	4.9 sec	34.1 sec	5.5 sec	11.2 sec	3.1 sec	9.4 sec		10.2 sec	9.2 sec
App redeploy[5]	1.2 sec	6.1 sec	2.3 sec	2.5 sec	2.2 sec	2.5 sec		1.2 sec	1.2 sec
RAM[5]	59 MB	175 MB	125 MB	236 MB	102 MB	376 MB		269 MB	430 MB
Download size[1]	11 to 94 MB	3 GB	10 MB	48 MB	10 MB	103 MB		127 MB	158 MB
Size installed[1]	15-123 MB	2.6 GB	17 MB	52 MB	12 MB	214 MB		159 MB	174 MB
Size per instance	0.5 MB	40 MB	0.4 MB	0.4 MB	0.4 MB	96 MB		1.5 MB	1.2 MB
Dev. Install[6]	5 sec	30 min	2 sec	3 sec	1 sec	5 sec		5 sec	5 sec
# of config files	1+	100+	8+	12+	20+	14+		16+	16+
Dynamic config[2]	99%	80%	20%	20%	20%	20%		60%	60%
IDE	Eclipse, IntelliJ IDEA, NetBeans are supported with some minor differences								
Configuration Editor	Eclipse UI	Browser UI	None	None	None	Browser UI	Browser UI	Browser UI	Browser UI
DevOps	Maven, Jenkins, Ant, Chef and other DevOps tools are supported with some minor differences								
Java EE	Java EE 7	Java EE 6+	JSP/ Servlet	Java EE 6 Web Prof.	JSP/ Servlet	Java EE 7	Java EE 7	Java EE 7	Java EE 6
Free Dev. License	IBM	IBM	Apache 2.0	Apache 2.0	EPL 1.0	CDDL 1.1	Oracle	LGPL 2.1	LGPL 2.1
Free Dev. Support	IBM[7]	IBM[7]	Self	Self	Self	Self	$	Self	Red Hat[8]

Source: http://bit.ly/1NRHPHq

Zero migration by design

Why upgrade?

- Gain new APIs in a new Java EE level, stay on a supported version of the Application Server, protect against security vulnerabilities, gain speed by moving to the faster new version of JDK or newer hardware

How easy is it to upgrade? Does it cost a fortune?

Liberty implements Zero Migration by design

- There is no migration needed for Liberty configuration - the same server configurations can be used with different versions and service levels
- Existing features will not change behavior, new feature ‘versions’ will be added and will contain all updates and changes

Tomcat and tc Server require considerable migration effort

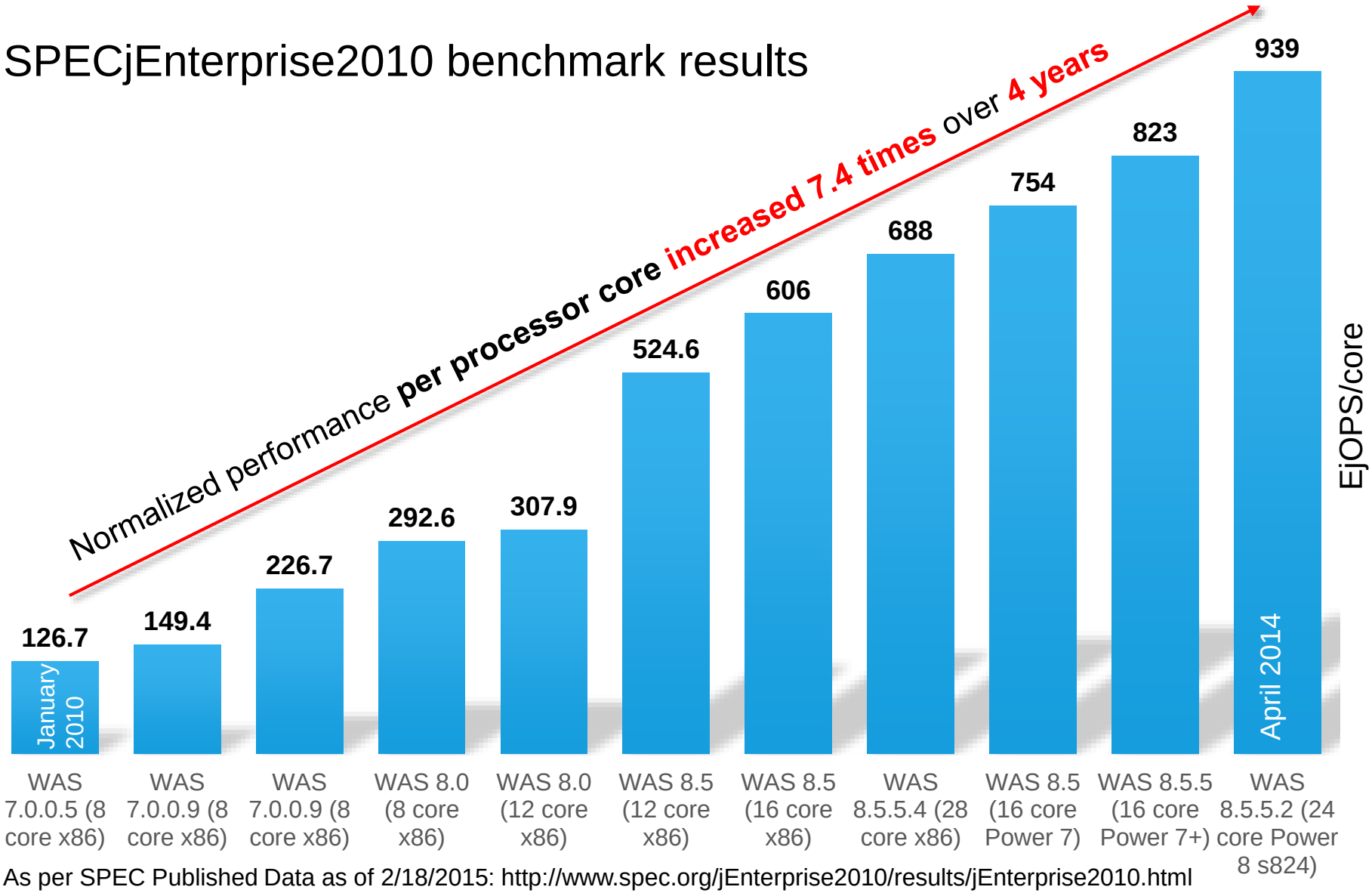
- Fresh server install and careful configuration updates and testing are required
- No automated upgrade or migration tool is provided

JBoss is not backwards compatible and migration is a pain

- Backwards compatibility was broken between JBoss v3.x, v4.x, v5, v6. Each of these releases have been disruptive and changed many properties and configuration files, scripting commands, CLI, Admin UI, APIs, etc. Upgrade path for JBoss is to manually copy configuration files and applications to new installation.
- These issues result in increased administration costs when using JBoss because of lost productivity related to unnecessary software development.

WebSphere release-to-release performance

SPECjEnterprise2010 benchmark results



As per SPEC Published Data as of 2/18/2015: http://www.spec.org/jEnterprise2010/results/jEnterprise2010.html

Benchmark results

SPECjEnterprise2010

Comparison of IBM vs. Oracle performance JOPS per core starting from 2011

More recent results

	JOPS/core	532.30	457.14	448.61	519.39	452.29	313.32	298.67
		WLS 12c on T5-2 (Jan'14)	WLS 12c on T5-8 (Sep'13)	WLS 11g on T5-8 (Mar'13)	WLS 11g on Sun x86 (Feb'12)	WLS 11g on Sun x86 (Jul'11)	WLS 11g on T4-4 (Aug'11)	WLS 11g on Dell x86 (Apr'11)
WAS 8.5.5.2 on Power8 (Apr'14)	939.31	1.76	2.05	2.09	1.81	2.08	3.00	3.15
WAS 8.5.5 on Power7+ (Apr'13)	822.57	1.55	1.80	1.83	1.58	1.82	2.63	2.75
WAS 8.5 on x3650 x86 (Nov'12)	606.03	1.14	1.33	1.35	1.17	1.34	1.93	2.03
WAS 8.5 on Power7+ (Sep'12)	681.39	1.28	1.49	1.52	1.31	1.51	2.17	2.28
WAS 8.5 on HS22 blade x86 (Apr'12)	524.62	0.99	1.15	1.17	1.01	1.16	1.67	1.76
WAS 8.5 on HS22 blade x86 (Jul'11)	307.86	0.58	0.67	0.69	0.59	0.68	0.98	1.03
WAS 8.5 on HS22 blade x86 (Jun'11)	292.64	0.55	0.64	0.65	0.56	0.65	0.93	0.98

More recent

1	even result
>1	IBM advantage
<1	Oracle advantage

Read full blog post: http://bit.ly/1dzHZBq

SPECjEnterprise2010

Comparison of IBM WAS ND vs. Oracle WLS Enterprise: $ cost per JOPS starting from 2011

More recent results

	$/JOPS	$131	$153	$251	$200	$175	$245
		WLS 12c on T5-8 (Sep'13)	WLS 11g on T5-8 (Mar'13)	WLS 11g on Sun x86 (Feb'12)	WLS 11g on Sun x86 (Jul'11)	WLS 11g on T4-4 (Aug'11)	WLS 11g on Dell x86 (Apr'11)
WAS 8.5.5 on Power7+ (Apr'13)	$81	1.62	1.90	3.11	2.47	2.16	3.03
WAS 8.5 on x3650 x86 (Nov'12)	$111	1.18	1.38	2.26	1.80	1.57	2.21
WAS 8.5 on Power7+ (Sep'12)	$223	0.59	0.69	1.13	0.90	0.78	1.10
WAS 8.5 on HS22 blade x86 (Apr'12)	$244	0.54	0.63	1.03	0.82	0.72	1.00
WAS 8.5 on HS22 blade x86 (Jul'11)	$168	0.78	0.91	1.50	1.19	1.04	1.46
WAS 8.5 on HS22 blade x86 (Jun'11)	$108	1.21	1.42	2.33	1.85	1.62	2.27

More recent

Comparison of the SPECjEnterprise2010 results

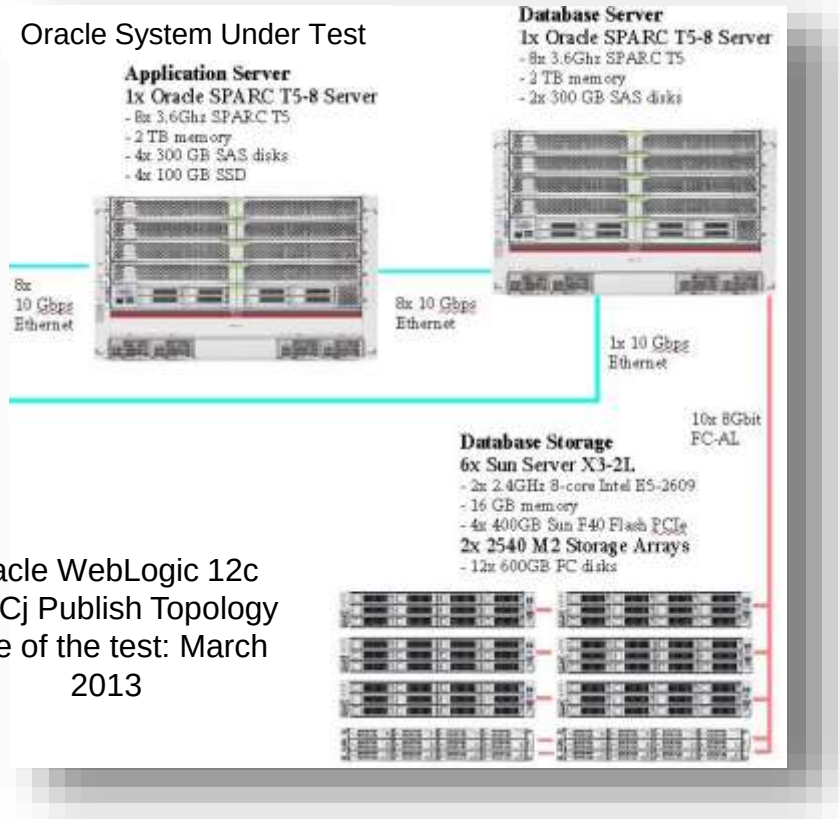
WebSphere and DB2 on Power7+ delivered 183% EjOPS per core at only 58% of the cost of Oracle WebLogic and Oracle DB on SPARC T5

ORACLE

Cost per transaction: **$109.45**

EjOPS per core: **449**

Oracle System Under Test



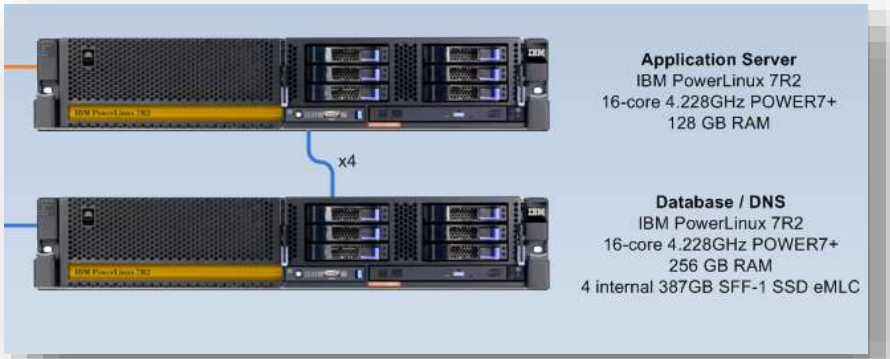
Oracle WebLogic 12c
SPECj Publish Topology
Date of the test: March 2013

IBM

Cost per transaction: **$63.79**

EjOPS per core: **823**

IBM System Under Test

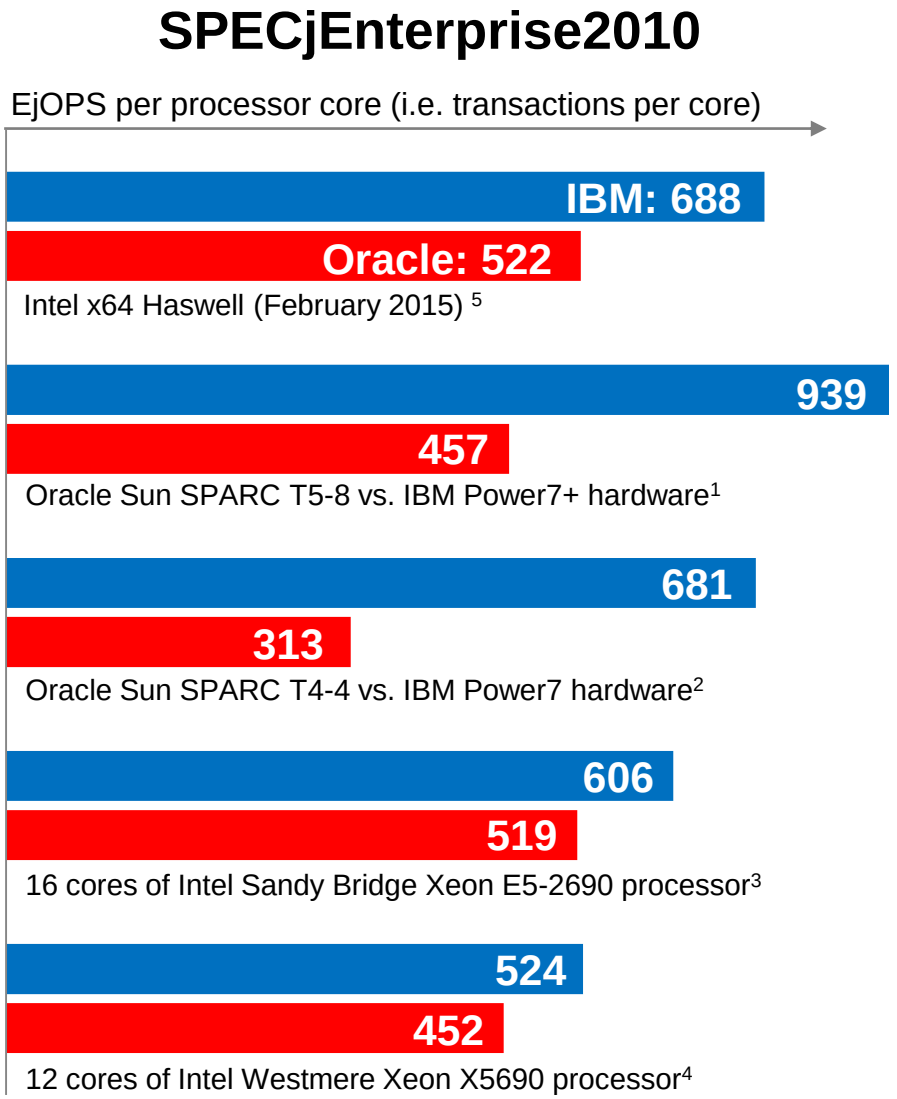


IBM WebSphere v8.5.5 SPECj Publish Topology
Date of the test: April 2013

Oracle total EjOPS: 57,422.17 http://www.spec.org/jEnterprise2010/results/res2013q1/jEnterprise2010-20130305-00041.html
IBM total EjOPS: 13,161.07 http://www.spec.org/jEnterprise2010/results/res2013q2/jEnterprise2010-20130402-00042.html

WebSphere performance compared to WebLogic

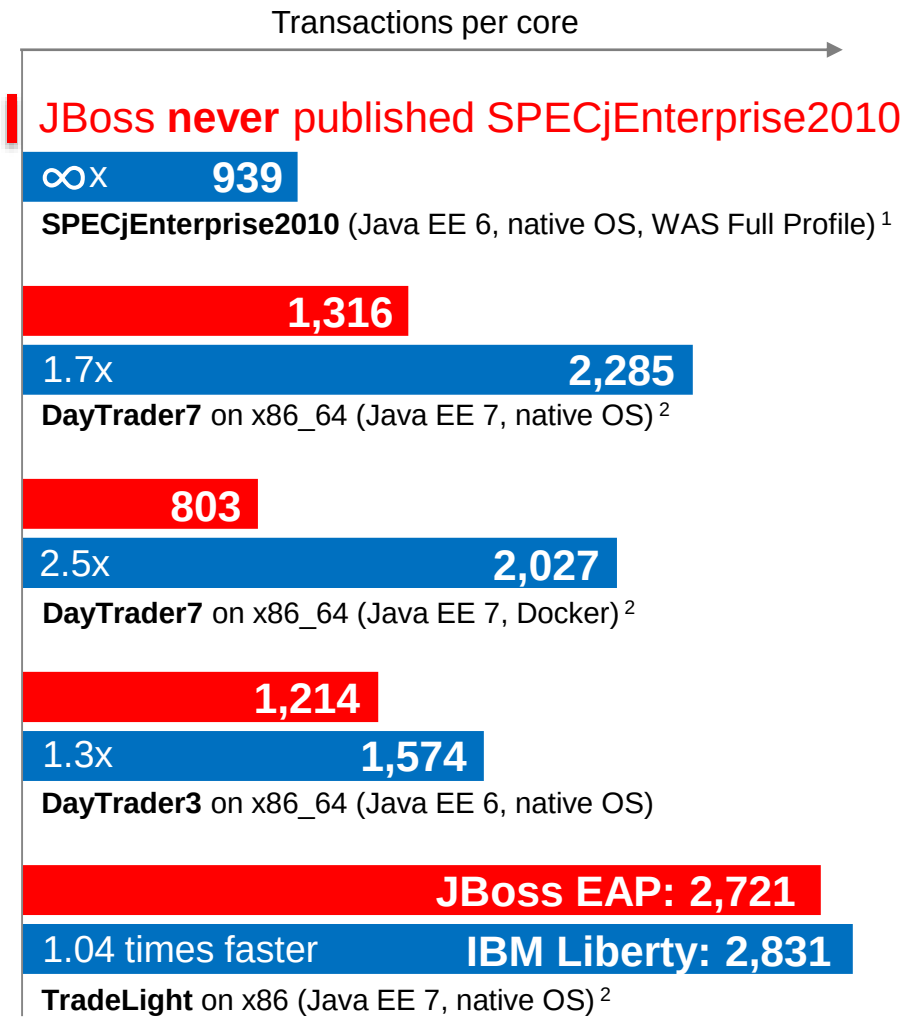
- IBM held the **most** records in ECPerf and was **FIRST to publish** SPECj2001, SPECj2002, SPECj2004, SPECjEnterprise2010
- WAS is **32%** faster per core on latest Intel Haswell at **half the cost** compared to WebLogic[1]
- On latest Intel Haswell processors WAS has the **fastest per socker, per core and biggest total EjOPS** result compared to WebLogic[2]
- WAS is **105%** faster per core at almost **half the cost** on Power7+ compared to WebLogic on SPARC T5[3]



(1) SPEC and SPECjEnterprise2010 are registered trademarks of the Standard Performance Evaluation Corporation. Results from www.spec.org as of 04/04/2013 Oracle SUN SPARC T5-8 449 EjOPS/core SPECjEnterprise2010 (Oracle's WLS best SPECjEnterprise2010 EjOPS/core result on SPARC). IBM Power730 823 EjOPS/core (World Record SPECjEnterprise2010 EJOPS/core result), (2) Results from www.spec.org as of 04/29/2012 Oracle SUN SPARC T4-4 313 EjOPS/core SPECjEnterprise2010 (Oracle's WLS best SPECjEnterprise2010 EjOPS/core result on SPARC). IBM Power780 681 EjOPS/core (World Record SPECjEnterprise2010 EJOPS/core result), (3) Results from www.spec.org as of 11/14/2012 Oracle SUN Fire X4170M3 519.39 EjOPS/core SPECjEnterprise2010 (Oracle's WLS best SPECjEnterprise2010 EjOPS/core result on Sandy Bridge). IBM WAS 8.5 System x3650 M4 Intel Sandy Bridge EjOPS/core (World Record SPECjEnterprise2010 EJOPS/core result) (4) Results from www.spec.org as of 04/29/2012 Oracle SUN Blade Server X6270 M2 452.285 EjOPS/core SPECjEnterprise2010). IBM Websphere HS 22 Blade 524.621 EjOPS/core.

WAS performance compared to JBoss and WildFly

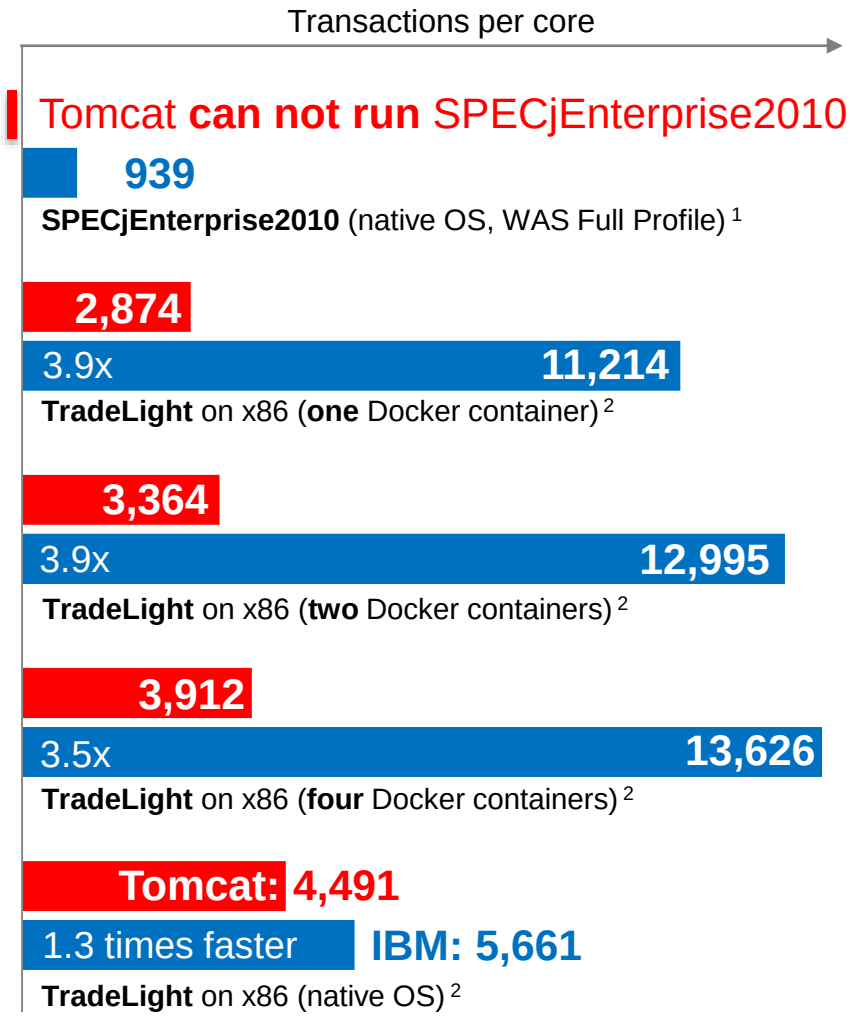
- IBM held the **most** records and was **first to publish** SPECj2001, SPECj2002, SPECj2004, SPECjEnterprise2010 while Red Hat has **never published** a single result for JBoss EAP
- IBM holds **World Record** for # of transactions per second per core with SPECjEnterprise2010 workload
- WAS is up to **2.5 times faster** than JBoss EAP (just consider license, hardware, power and cooling savings!)
- Many **independent customer** benchmarks confirm WAS performance advantage
- IBM is heavily investing in performance optimizations of WAS for **Docker Containers and Cloud Foundry**



(1) SPEC and SPECjEnterprise2010 are registered trademarks of the Standard Performance Evaluation Corporation. Results from www.spec.org as of 07/30/2015: IBM Power730 939 EjOPS/core (World Record SPECjEnterprise2010 EJOPS/core result),
(2) Since JBoss EAP 6.4 does not support Java EE 7, results for WildFly are provided

WAS performance compared to Tomcat

- IBM held the **most** records and was **first to publish** SPECj2001, SPECj2002, SPECj2004, SPECjEnterprise2010 while Red Hat has **never published** a single result for JBoss EAP
- IBM holds **World Record** for # of transactions per second per core with SPECjEnterprise2010 workload
- Liberty is up to **2.4 times faster** than Tomcat (just consider license, hardware, power and cooling savings!)
- Many **independent customer** benchmarks confirm WAS performance advantage
- IBM is heavily investing in performance optimizations of WAS for **Docker Containers and Cloud Foundry**



(1) SPEC and SPECjEnterprise2010 are registered trademarks of the Standard Performance Evaluation Corporation. Results from www.spec.org as of 07/30/2015: IBM Power730 939 EjOPS/core (World Record SPECjEnterprise2010 EJOPS/core result),

Global transactions

Enterprises often have to update data in:

- Databases (DB2, Oracle DB, MS SQL, Informix)
- TP monitors (CICS, Tuxedo)
- Messaging servers (IBM MQ, Tibco EMS, ActiveMQ)
- Caching servers (WXS), etc.

What happens if there is a failure during the transaction commit?

Tomcat and tc Server do not include a transaction manager

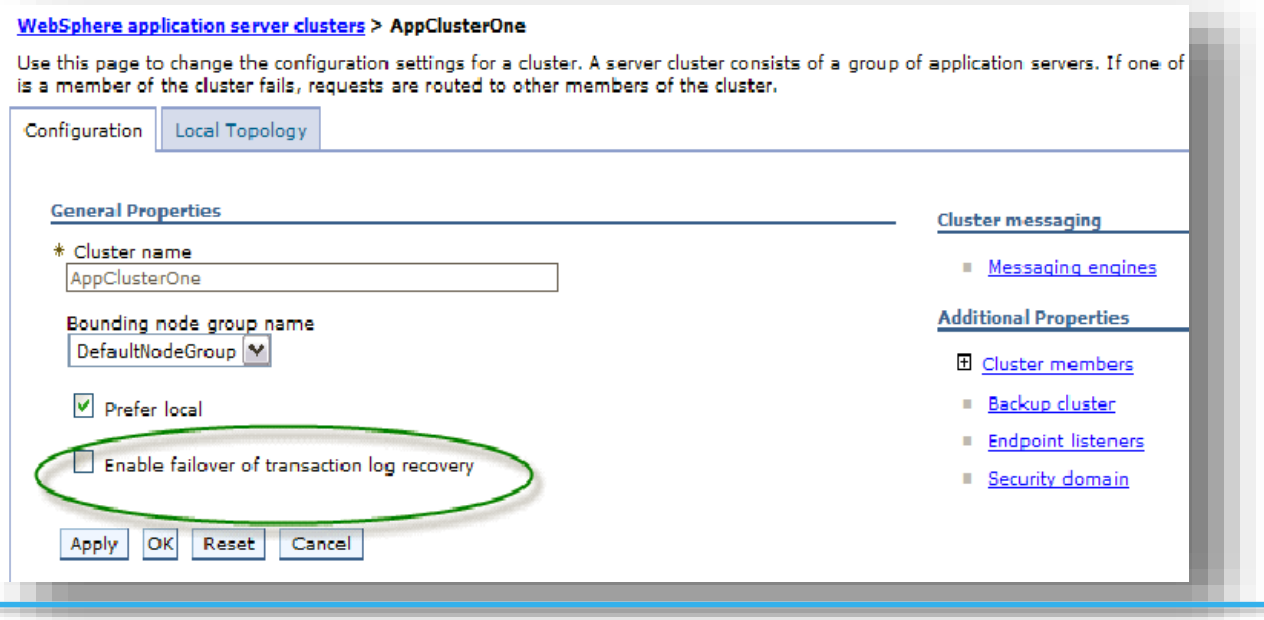
- Only support **local** transactions (with a single resource)
- Without a distributed transaction manager, there is no support for two-phase commit for global transactions, where the scope is across multiple resources
- Writing business critical code without global transaction management is possible, but is very complex (i.e. error prone and expensive)

WAS and Liberty support distributed transactions with two-phase commit and automatic data recovery in the event of a network failure, ensuring transaction integrity

Automatic transaction log failover & recovery

WAS ND Full Profile provides failover of 2PC transactions

- WAS ND can be configured to store transaction logs for each server on shared file system or in the HA RDBMS, which allows all peers to see all logs
- When a WAS ND cluster member fails, a peer is elected to recover the Transaction Log from the failed server
- In Doubt Transactions from a failed Server are recovered very quickly
- Significantly faster and cheaper than hardware clustering (seconds vs. minutes)



JBoss, Tomcat and Liberty Profile do not provide similar capabilities, which leads to longer recovery times and more admin labor

JBoss high availability issues

High availability with JBoss may be difficult to achieve without introducing significant redundancy and admin effort

JBoss App Server JVMs often need restart after configuration updates

- Examples: changing the data source pool size or other settings, changing JMS configuration, re-deploying an EAR multiple times (only WAR hot-deploy works, the EAR hot deploy only works for the first couple of times, then causes out of memory errors and JVM crash), etc.
- WAS does not need to be restarted as above updates are dynamic

JON server uses database for configuration and monitoring data

- The DB must be made HA to avoid SPOF and this requires extra license and hw costs

Ripple restart of application servers in a cluster is not provided. Administrator must manually (or via scripting) restart servers in a cluster one by one

- Increases complexity of administration

Transaction log requires manual failover effort in case of server failure

- WAS failover is automated and takes seconds

Application deployment causes service interruption

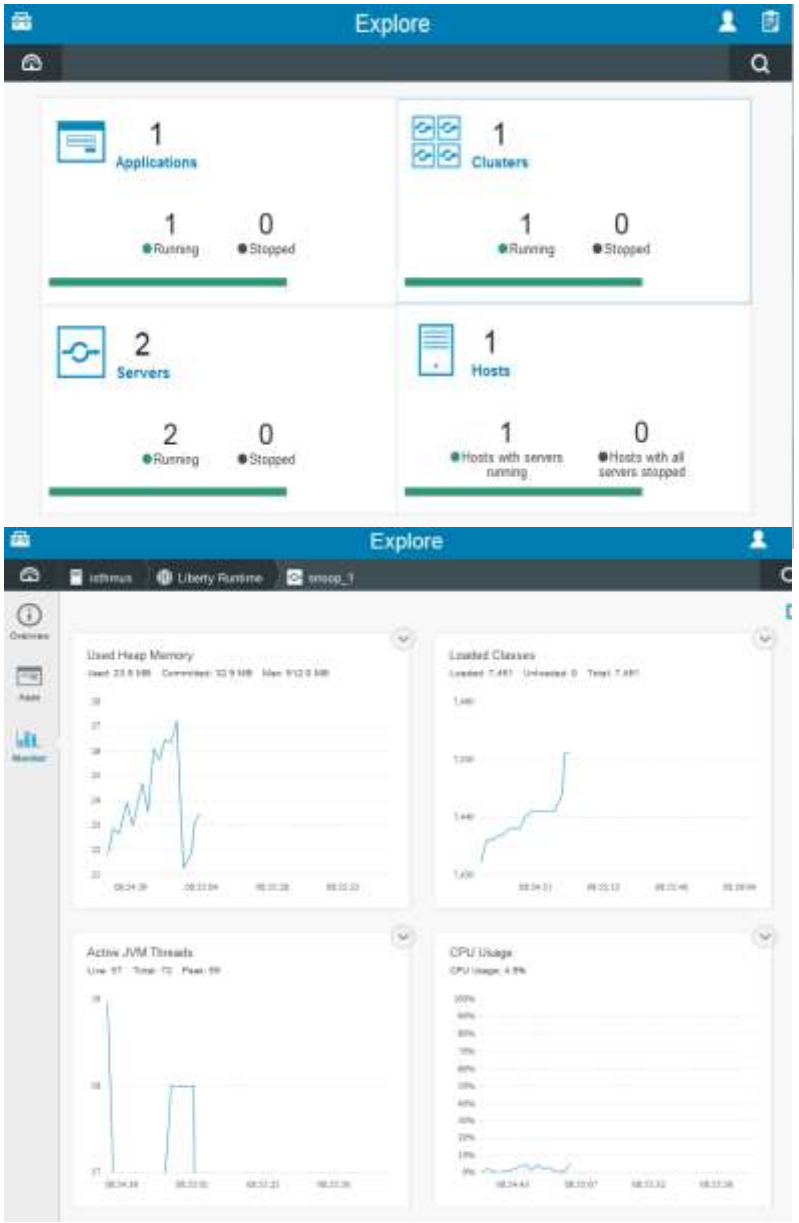
- WAS ND solves this by introducing the application versioning and graceful client transfer

Security

	WAS	WAS Liberty	Oracle WLS	JBoss EAP	Apache Tomcat
Basic web application security	Yes	Yes	Yes	Yes	Yes
Java EE Security Standards	Yes	Yes	Yes	Yes	no
Role admin security (who can do what)	Yes	no	Yes	no	no
Resource admin security (who can manage what)	Yes	no	Yes	Yes	no
Audit and track of configuration changes	Yes	no	Yes	Some	no
LDAP support and compatibility	Yes	Yes	Yes	Some	Some
Federated User Registry	Yes	Yes	Yes	no	no
SPNEGO Web Inbound	Yes	Yes	Yes	Yes	no
Kerberos	Yes	no	Yes	Yes	no
OAUTH	Yes	Yes	Yes	Yes	no
SAML	Yes	Yes	Yes	Yes	no
OpenID	Yes	Yes	no	no	no
OpenID Connect	Yes	Yes	no	no	no
Keys and Certificate Management	Yes	no	Yes	no	no
Multiple Security Configurations	Yes	no	Yes	no	no
DB2 Trusted Context for Identity propagation	Yes	Yes	no	no	no
Secure Engineering Accreditation OTT-PS	Yes	Yes	no	no	no
Encryption Standard FIPS 140-2, 800-131a	Yes	Yes	no	Some	Some

Source: http://whywebsphere.com/2015/06/26/1495/

Liberty Admin Center



A browser-based UI

- Deployment, monitoring and managing Liberty single
  servers and collectives

Deploy

- Server package (runtime + server + apps)

Monitor

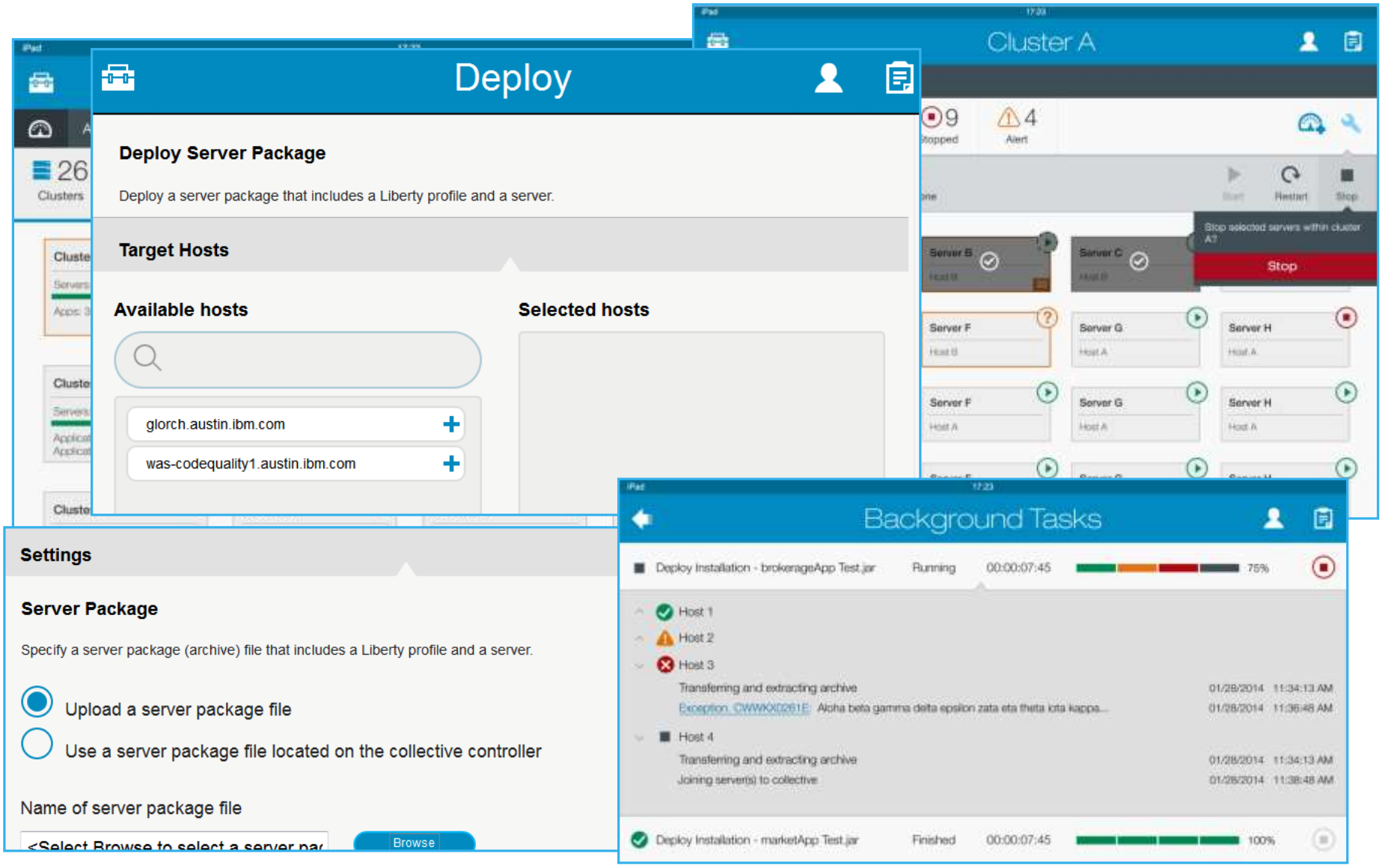
- Performance and Health metrics
- Log Analytics [beta]
- Dashboard, Alerts and Notifications
  [under investigation]

Manage

- Browse, search, filter
- Tags and metadata
- Start / Stop / Restart
- Auto scaling (demo)
- Server Configuration [beta]
- Health Management [under investigation]

Tomcat does not provide similar capabilities, which leads to more admin labor

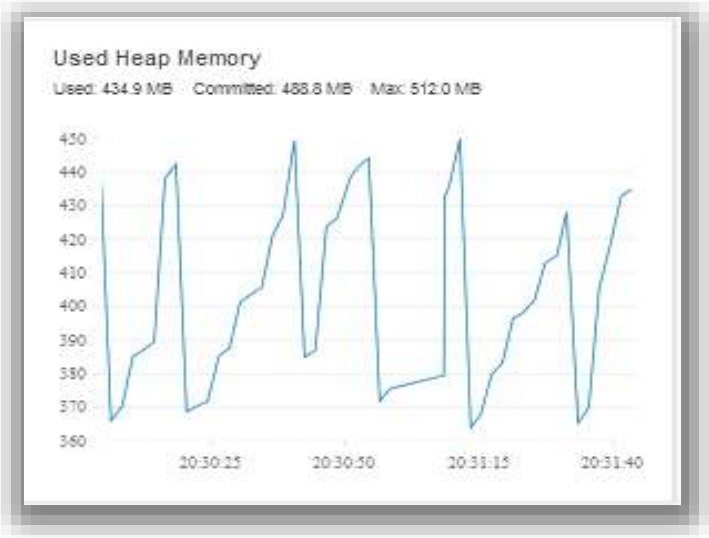
Resource details, multiple target actions, asynchronous Liberty server deployment

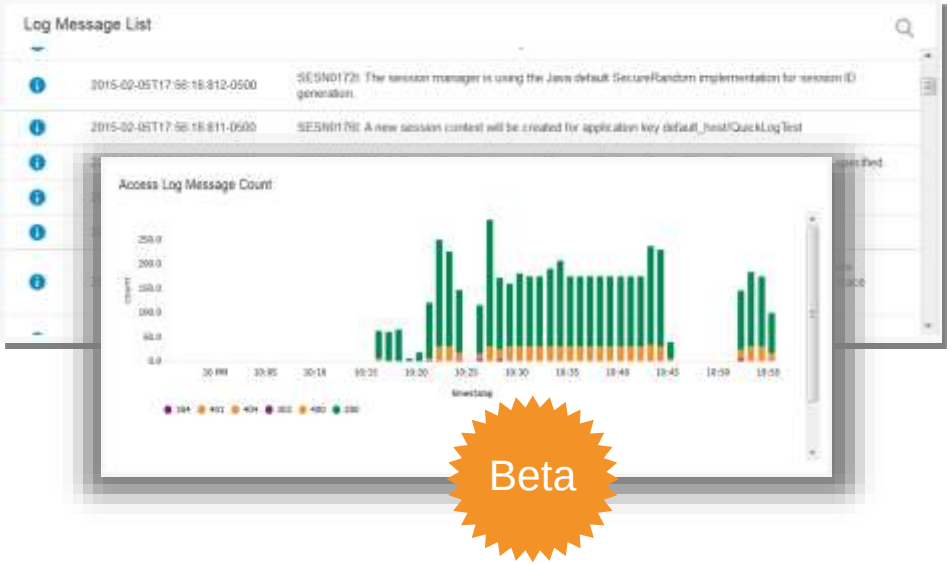


Liberty Monitoring and Log Analytics

Out of the box capabilities:

- Monitor your Application Infrastructure
- Identify the root cause of a problem
- Ensures High Availability of your application
- Plus wide set of additional Java monitoring tools available in the IBM Java Health Center





Monitor:

- Used Heap Memory
- Loaded Classes
- Active JVM Threads
- CPU Usage
- Liberty MXBeans

Analyze data from:

- Access Log, Log Messages, FFDC & Trace Messages

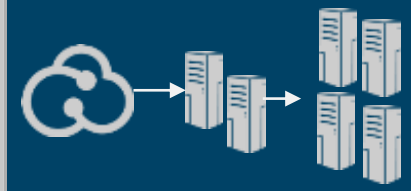
WebSphere Intelligent Management

Application Edition Management

Server Health Management

SLA based Dynamic Clustering

Intelligent Routing and SLA Enforcement



Better TCO through management efficiency and performance, Intelligent Management delivers the ability to sense and respond quickly to changes

Up to **45%** **less hardware**

Up to **60%** **less administration**

Up to **45%** **less software**

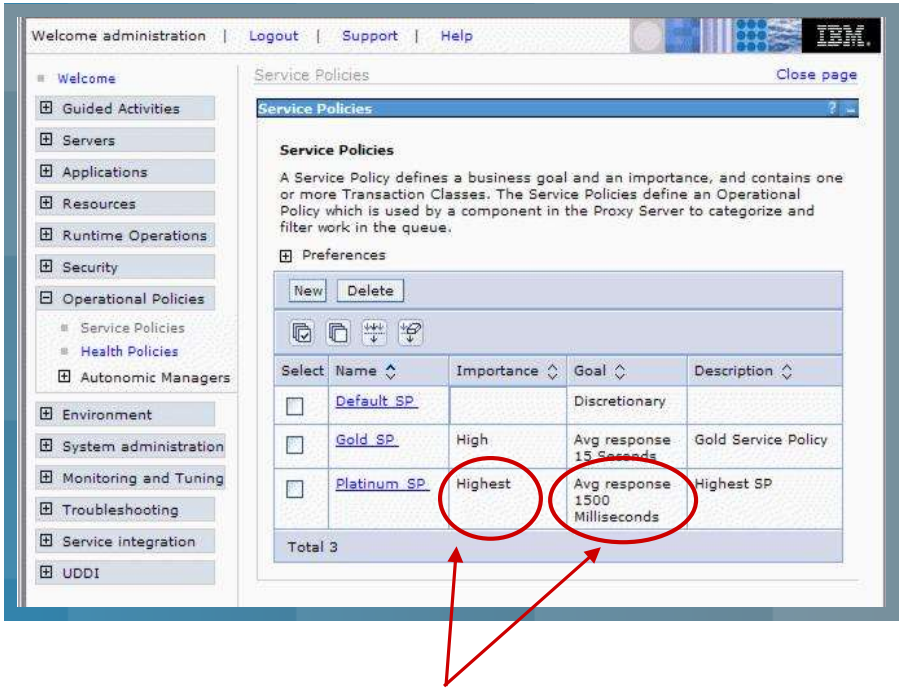
Up to **90%** **fewer outages**

Source: Based on 60+ Operations Optimization Value Assessments done to date by IBM for real customers
Cost reductions are compared to traditional WAS ND deployment

WAS ND traditional - Service Policies

Easily allows an administrator to specify the relative importance of applications and optionally a response time goal. WebSphere then manages your applications according to this policy.

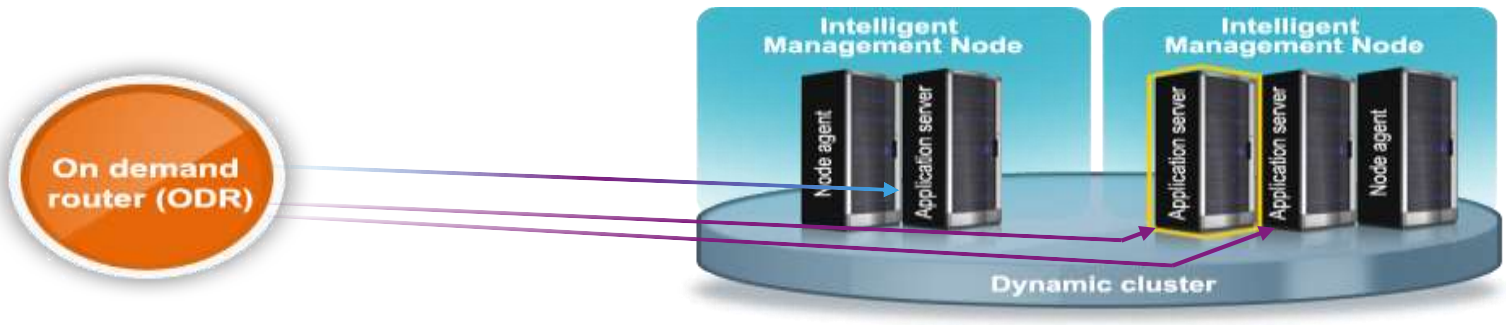
- Service policies are used to define application service level goals
- Allow workloads to be classified, prioritized and intelligently routed
- Enables application performance monitoring
- Resource adjustments are made if needed to consistently achieve service policies



Service Policies define the relative importance and response time goals of application services; defined in terms the end user result the customer wishes to achieve

Dynamic Clustering

- Proactively provision and start or stop application servers based on workload demands to meet Service Level Agreements
- Associate service policies with your applications
  - Let WebSphere manage to the service goals
- Programmatically respond to spikes in demand
  - Add or reduce application server instances as appropriate
- Automatically recover from infrastructure problems
- Includes automatic start and stop of cluster members based on load for MQ-driven applications
- Decrease administrative overhead required to monitor and diagnose performance issues

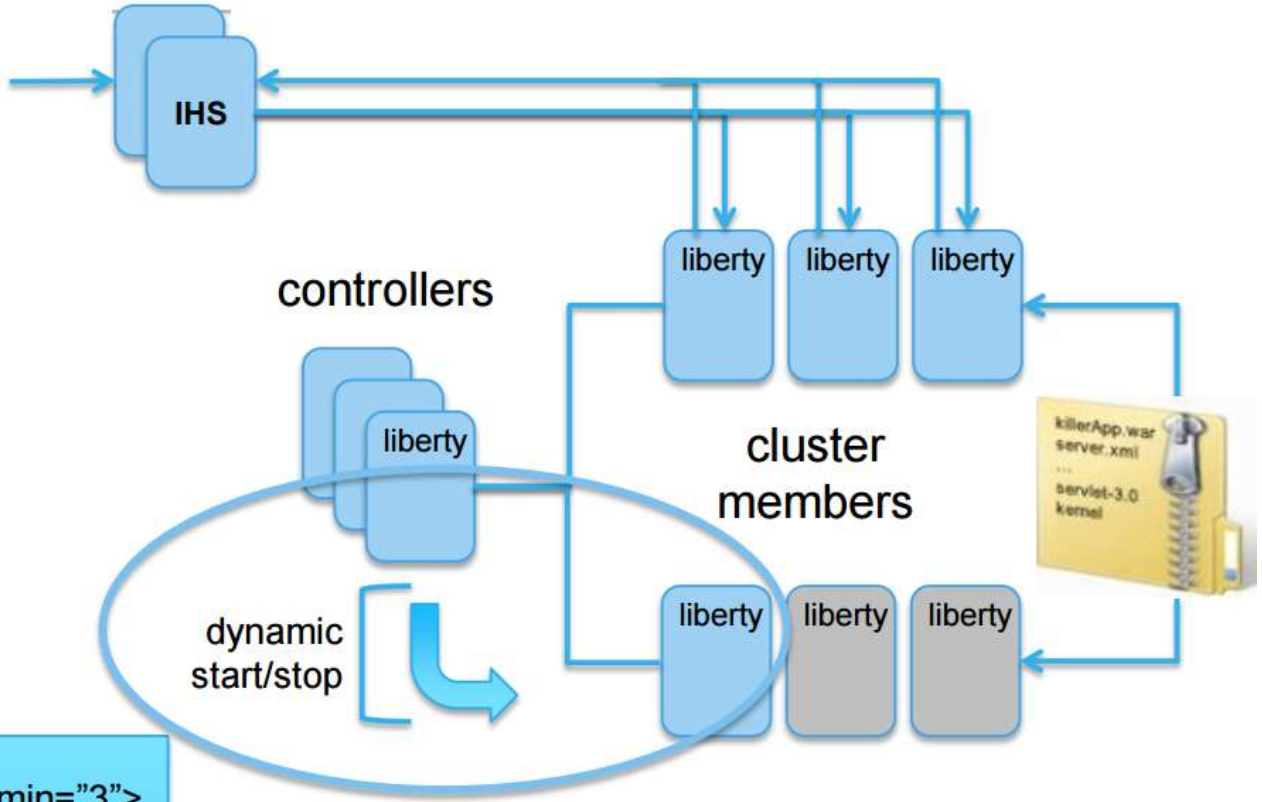


Auto-scaling of Liberty Collective

- Policy bound to app clusters
- Automatic start/stop of JVMs to scale up/down capacity
- Dynamic routing

Administratively defined policy

```
<scalingPolicy name="policy1"  min="3">
    <metric name="heap" max="80"/>
    <bind clusters="cluster1"/>
</scalingPolicy>
```



Tomcat and JBoss do not offer comparable capabilities => more admin labor, less responsive applications

Dynamic clustering (auto scaling)

	WAS ND	WAS ND Liberty	Oracle WLS EE	JBoss EAP	Apache Tomcat
Static clusters (pre-provisioned)	Yes	Yes	Yes	Yes	Yes
Manually add or remove servers to/from a running cluster	Yes	Yes	Yes	Yes	Yes
Centralized management of cluster members	Yes	Yes	Yes	Yes	no
Dynamically starts & stops servers when load changes	Yes	Yes	Yes	no	no
Provisions new servers to hosts when workload increases	Yes	Yes	no	no	no
Scaling policy allows for min and max number of servers	Yes	Yes	Yes	no	no
Scaling policy based on CPU, heap or memory use	Yes	Yes	Yes	no	no
Scaling based on service policies (URL, response time, etc.)	Yes	no	Yes	no	no
Apps have relative priorities when servers are allocated	Yes	no	no	no	no
Auto vertical stacking on a node	Yes	no	Yes	no	no
Cluster isolation groups	Yes	no	no	no	no
Lazy application start	Yes	no	Yes	no	no

Source: http://whywebsphere.com/2015/12/14/dynamic-clustering-auto-scaling-in-websphere-weblogic-jboss-and-tomcat/

Intelligent routing

	WAS ND[1]	WAS ND Liberty[2]	WAS ND + Data Power AO[8]	Oracle WLS EE	Traffic Director & Exalogic[7]	JBoss EAP + EWP	Apache HTTP + Tomcat
High performance	some	some	Yes	some	some	some	some
WLM across static app server clusters	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SSL termination and HTTP compression	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Auto updates of configuration when cluster or app change	Yes	Yes	Yes	Yes	Yes	Yes	no[9]
JVM maintenance mode	Yes	Yes	Yes	Yes	Yes	no	no
Node (or host) maintenance mode	Yes	Yes	Yes	Yes	Yes	no	no
Application edition-aware routing	Yes	no	Yes	Yes	Yes	no	no
Dynamic clusters auto-grow or shrink based on workload	Yes	Yes	Yes	no	no	no	no
Health policy support	Yes	Yes	Yes	no	no	no	no
Auto-adjusts server weights based on resource use	Yes	no	Yes	no	Yes	no	no
Traffic shaping and SLA enforcement for HTTP	Yes	no	Yes	no	Yes	no	no
CPU and heap overload protection	Yes	no	Yes	no	no	no	no
Support traffic shaping for 3rd party servers	Yes	no	Yes	no	no	no	no
Custom rules for request routing (URI, IP address, etc.)	Yes	no	Yes	no	Yes	no	no
Request rate limiting	no	no	Yes	no	some	no	no
Number of client connections limiting	no	no	Yes	no	Yes	no	no
Content based routing	no	no	Yes	no	some	no	no
Protect against XML and SQL injection attacks	no	no	Yes	no	no	no	no
XML processing (parsing, transformation, validation, etc.)	no	no	Yes	no	no	no	no
Custom advisors	no	no	Yes	no	no	no	no
DMZ ready	Yes[5]	Yes	Yes	no	no	Yes	Yes
Static file serving and in-memory and disk page caching	Yes[3]	Yes[3]	no	no[4]	Yes	Yes	Yes
Replace hardware based load balancer(s)	Yes[6]	Yes[6]	Yes	no	some	no	no
Traffic shaping and SLA enforcement for IIOP and JMS	Yes	no	no	no	no	no	no
HTTP session rebalancing	Yes	no	no	no	no	no	no

Source: http://whywebsphere.com/2015/12/08/traffic_shaping/

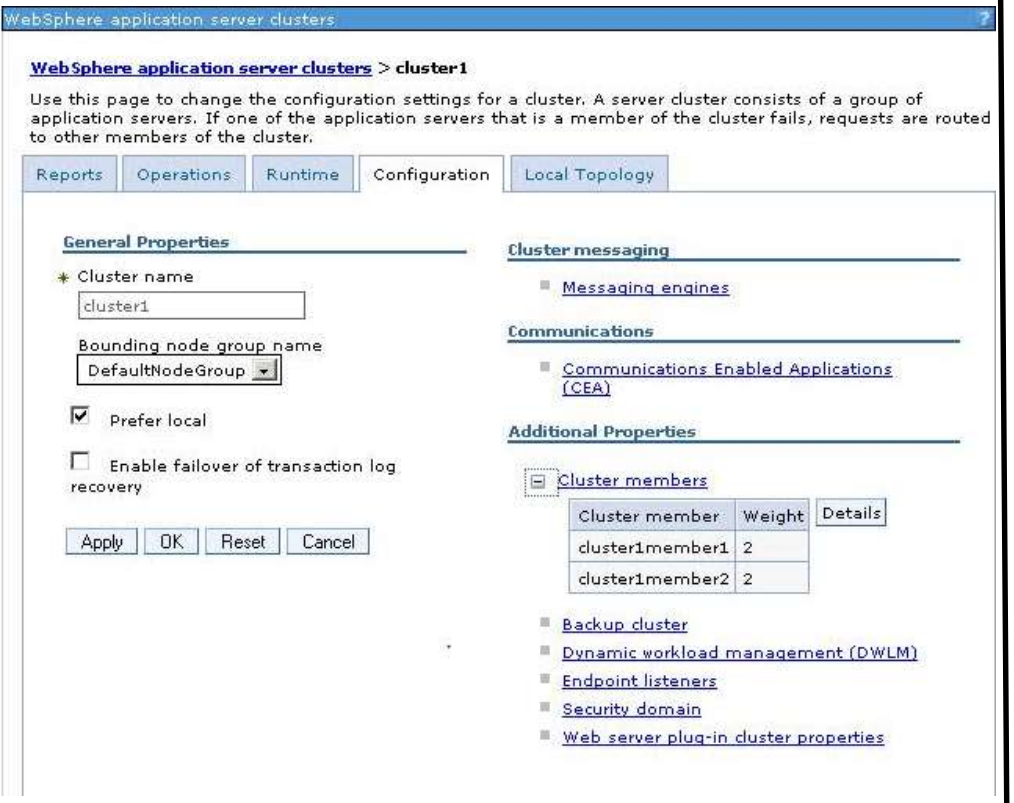
Dynamic clustering

	WAS ND traditional	JBoss EAP
New node is added to a cell	If node meets the dynamic selection criteria, it is automatically added to the dynamic cluster as potential host for the JVM	Static cluster member must be manually defined for each participating node and manually added to the static cluster.
Vertical stacking (VS)	If VS is allowed, JVM process definitions are automatically created for each node	Cluster members must be manually created and port conflict resolution must be manually done for each new JVM
Cluster isolation	Dynamic cluster can belong to different isolation groups and conflicts are automatically resolved	Manual work is required to prevent conflicts between JVMs that must be isolated from each other
Workload increase	If workload increases for the application, new members of dynamic cluster are started to accommodate such increased workload	Manual start of cluster members is required to accommodate increase in workload
Workload decrease	When workload drops off, members of dynamic clusters may be stopped if CPU or memory are required for other workloads. Lazy application start can be configured	Manual stop of instances is required to free up resources for other workloads. Application must always be up and running to accept workload
Critical load and resource shortage	When overall workload is greater than the system can handle, service policies are enforced such that more important applications get priority over less important ones and SLA policies for response times are met. SLAs can be defined based on a rule set based on URI, time, user properties, IP, etc.	No provision for prioritization of workload, no SLAs for applications. Typical solution is to create duplication by using dedicated hosts (physical or virtual) for each workload, which increases admin complexity, hardware and software cost
Server properties	Server template can be updated and changes are reflected on all members of dynamic cluster automatically	Properties must be updated on each member of the static cluster manually

Source: http://whywebsphere.com/2015/12/14/dynamic-clustering-auto-scaling-in-websphere-weblogic-jboss-and-tomcat/

Administration with WebSphere is easier

WAS ND 8.5 has a UI



Cluster Configuration

tc Server 2.8.1: Hand-edited XML

```
<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"
         channelSendOptions="8">
  <Manager className="org.apache.catalina.ha.session.DeltaManager"
           expireSessionsOnShutdown="false"
           notifyListenersOnReplication="true"/>
  <Channel className="org.apache.catalina.tribes.group.GroupChannel
    <Membership className="org.apache.catalina.tribes.membership.Mc
                address="228.0.0.4"
                port="45564"
                frequency="500"
                dropTime="3000"/>
    <Receiver className="org.apache.catalina.tribes.transport.nio.N
              address="auto"
              port="4000"
              autoBind="100"
              selectorTimeout="5000"
              maxThreads="6"/>
    <Sender className="org.apache.catalina.tribes.transport.Replica
      <Transport className="org.apache.catalina.tribes.transport.ni
    </Sender>
    <Interceptor className="org.apache.catalina.tribes.group.interc
    <Interceptor className="org.apache.catalina.tribes.group.interc
  </Channel>

  <Valve className="org.apache.catalina.ha.tcp.ReplicationValve"
         filter=""/>
  <Valve className="org.apache.catalina.ha.session.JvmRouteBinderVa

  <Deployer className="org.apache.catalina.ha.deploy.FarmWarDeploye
            tempDir="/tmp/war-temp/"
            deployDir="/tmp/war-deploy/"
            watchDir="/tmp/war-listen/"
            watchEnabled="false"/>

  <ClusterListener className="org.apache.catalina.ha.session.JvmRou
  <ClusterListener className="org.apache.catalina.ha.session.Cluste
</Cluster>
```

Hand-editing XML files leaves no audit trail for changes!

How do customers really use JBoss EAP?

> *"One minute of system downtime can cost an organization anywhere from $2,500 to $10,000 per minute. Using that metric, even 99.9% availability can cost a company $5 million a year"* - The Standish Group

Vast majority of JBoss EAP and WildFly customers are not using clustering

- Must tolerate lower quality of services ($$$)

Red Hat customers are forced to purchase 3rd party management tools, monitoring tools, configuration management tools, performance profilers, etc.

- 3rd party tools require license and support payments ($$$)
- 3rd party tools are not always in synch with the desired version of JBoss ($$$)
- 3rd party vendor viability poses risks ($$$)

Most JBoss customers invest significant staff time to build home grown scripting frameworks for JBoss management (a combination of shell scripting and generation of JBoss XML files using XSLT, Java or other template mechanism)

- Cost to develop, debug, maintain such scripts can be significant ($$$)
- New major versions of JBoss (major or minor) are not backwards compatible, causing significant rework of home grown scripts and tools ($$$)

WAS provides needed administrative tools out of the box at no extra cost

WAS ND traditional - App Edition Management

Applications can be upgraded or downgraded without incurring outages or requiring additional hardware and license costs

- Upgrade Applications without interruption to end users
- Concurrently run multiple editions of an application
- Automatically route users to a specific application
- Two editions can be activated for extended periods of time (one production and one validation)
- Rollout policies to switch from one edition to another without service loss
- Easy-to-use edition control center in admin console
- Full scripting support

Validation Mode

Rollout Policies

Concurrent Activation

App deployment in a clustered environment

	WAS ND Liberty	WAS ND	Oracle WLS EE	JBoss EAP	Apache Tomcat
Cluster-wide hot application update with interruption	Yes	Yes	Yes	Yes	no
Sequential interruption free update of a compatible app	no	Yes	Yes	no	no
Interruption free update of an incompatible app	no	no	no	no	no
Atomic update of an app	no	Yes	no	no	no
Previous app editions can be activated from history	no	Yes	no	no	no
Run 2 editions of the same app concurrently	no	Yes	Yes	no	no
Run app in validation mode w/ the other edition running	no	Yes	no	no	no
Run 3 or more editions of the same app concurrently	no	no	no	no	no

Automatic Health Management

Sense and respond to problems before end users suffer an outage

Fully automatic or manual mode

- Supervised mode generates recommendations, but actions are approved by the user
- Automatic mode handles health problems without human intervention
- Designed to maintain continuous availability

Health policies

- Consist of a (1) condition, (2) actions, and (3) target set of processes

Health conditions

- User defined rules with logical expression builder

Corrective actions

- Notify administrator
- Capture diagnostics
- Restart server or put it in or out of maintenance mode
- Custom actions (shell scripts or Java)

Health conditions

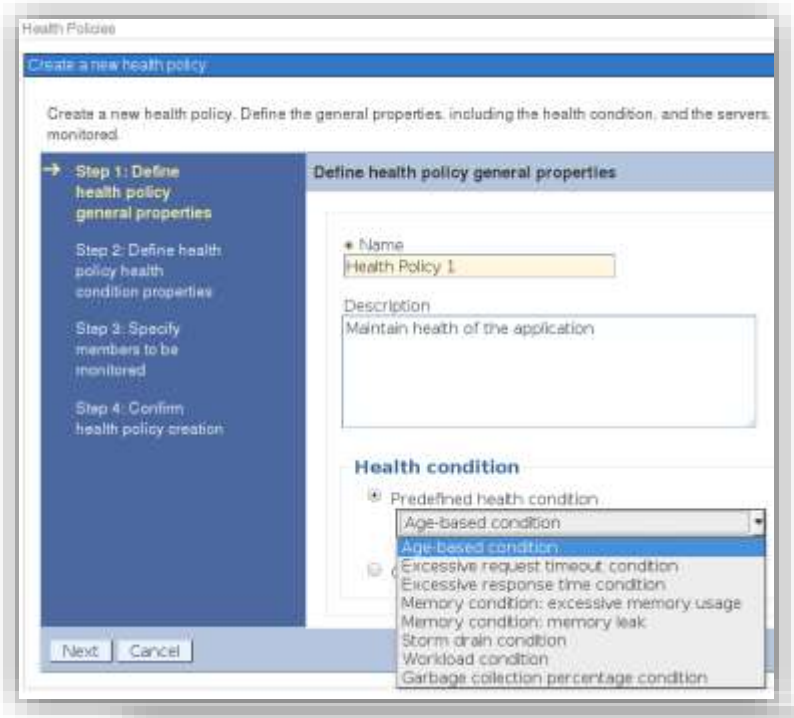
WAS traditional and Liberty

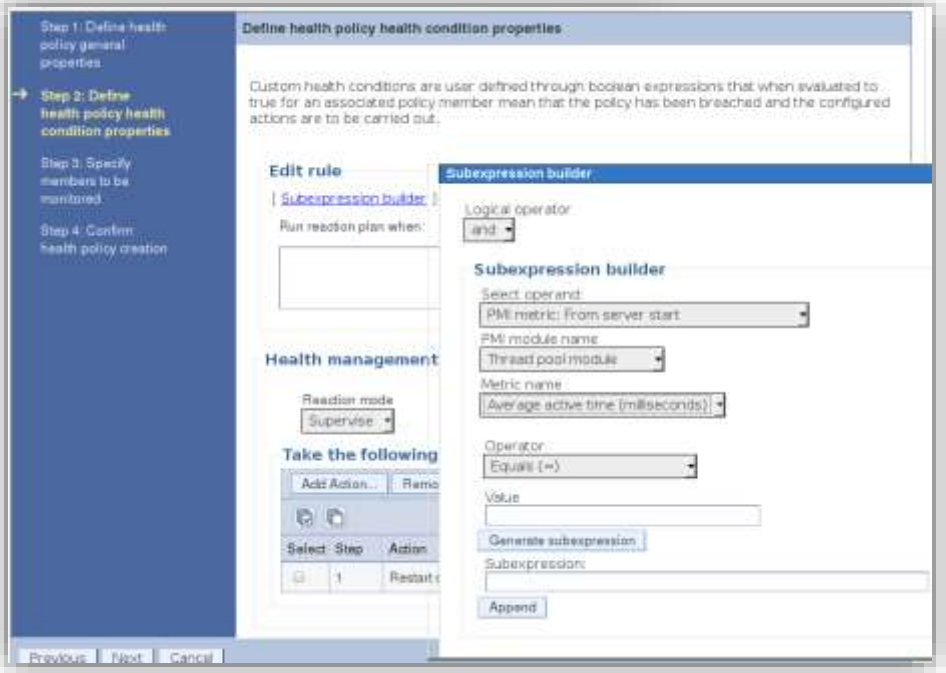
Predefined health conditions

- Excessive request timeouts
  - % of timed out requests
- Excessive response time
  - average response time
- Excessive memory
  - % of maximum JVM heap size
- Memory leak
  - JVM heap size after garbage collection
- Age-based*
  - amount of time server has been running
- Storm drain*
  - significant drop in response time
- Workload*
  - total number of requests
- Garbage collection*
  - percentage of time spent by JVM on GC

Custom health conditions

- WAS traditional (at this time)
- Conditions based on any PMI metric, MBean ops and attributes, or combination using complex Boolean expressions





* - currently available only in WAS traditional

Health actions

Available in WAS traditional and Liberty

Predefined health actions

- Commonly used actions are provided, such as thread dumps, heap dumps, server restart, put in and out of maintenance mode, etc.

Custom health actions*

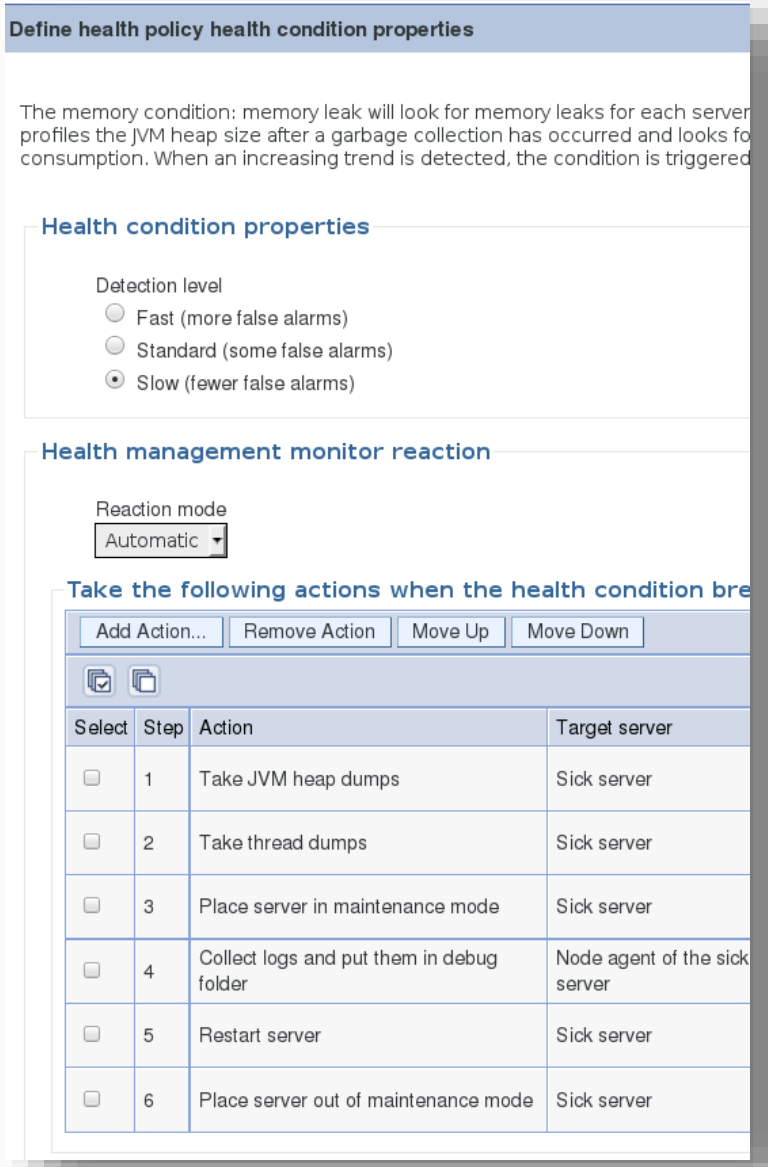
- Any Java class or executable OS program can be invoked with custom arguments

Sequences of actions

- Reusable library of actions at the cell level
- Actions can be executed in user defined sequence

Targets

- Actions can be applied to any server in a cell and any node agent in a cell (not only sick servers)
- Actions can be applied to WAS and other managed servers (Tomcat, JBoss, etc.)



* - currently available only in WAS traditional

Memory leak, slow and hung threads detection

Diagnose issues and increase QoS for “broken” apps in WAS traditional and Liberty

Detection & warnings

- Class loader memory leaks, leaks triggered by JRE, application triggered leaks
- Hung threads (infinite loops, resource deadlocks, etc.)

Prevention actions

- Clear resource caches, stop new timer threads, renew threads in a thread pool, etc.

Recovery actions

- Stop “sick” apps, interrupt threads, unregister JDBC drivers, get heap and system dumps, etc.

Liberty: Request Timing feature

- Which request is slow? Which is hung? Why?
- Slow servlet requests are detected and a full diagnostic of the request is dumped to the log
- Hung servlet requests are detected, triggering creation of a set of javacores
- Use in your production environments to catch issues the first time they occur

Dramatically reduce the time it takes to diagnose the source of slow requests

```
TRAS0112W: Request websphere.servlet.service|DayTrader Web | TradeScenarioServlet(AAC9KLwFFXT_AAAAAAAAAAN) has been
running on thread 0000006b for 1549.460ms. The following stack trace shows what this thread is currently doing.
        <stack trace>
The following table shows the events that have run during this request.
Duration     Operation
1552.012ms + websphere.servlet.service | DayTrader Web | TradeScenarioServlet
   0.014ms       websphere.session.getAttribute | R-ObCtcDfR8Zd9riQEMCh6R | uidBean
  30.714ms       websphere.servlet.service | DayTrader Web | TradeAppServlet
   0.010ms           websphere.session.getAttribute | R-ObCtcDfR8Zd9riQEMCh6R | uid
  30.456ms           websphere.servlet.service | DayTrader Web | /quote.jsp
  28.903ms               websphere.servlet.service | DayTrader Web | /displayQuote.jsp
   0.194ms                   websphere.datasource.psExecuteQuery | jdbc/TradeDataSource | SELECT t0.CHANGE1, t0.COMPANYNAME..
1520.695ms +     websphere.servlet.service | DayTrader Web | TradeAppServlet
   0.013ms           websphere.session.getAttribute | R-ObCtcDfR8Zd9riQEMCh6R | uidBean
   0.190ms           websphere.datasource.psExecuteQuery | jdbc/TradeDataSource | SELECT t0.ADDRESS, t0.CREDITCARD, ...
   0.135ms           websphere.datasource.psExecuteQuery | jdbc/TradeDataSource | SELECT t0.ACCOUNTID, t0.BALANCE, ...
...
```

```
<requestTiming
    slowRequestThreshold="10s"
    hungRequestThreshold="600s"
    includeContextInfo="true"
    sampleRate="1"
/>
```

JBoss and Tomcat do not provide similar capability => increased admin labor for troubleshooting

Liberty: Event Logging feature

Track events running in your Liberty applications

- How do I know what events are happening in the server and how long do they take?
- Create log entries for any Servlet or JDBC request, or HTTP get/set attribute operation
- Use with minimum duration setting to watch for slow events in production environments
- Use with all events enabled to show what apps are doing in development and test
- For best performance use binary logging of Liberty and increase sample rate to >1

Know what's happening in your applications

Servlet events include
servlet name, path info, query string

Servlets:

```
[6/18/14 16:21:35:761 IST] 0000002a EventLoggingProbeExtension.class                    I BEGIN
requestID=AAADvUHkFwy-AAAAAAAAAAD # type=websphere.servlet.service # contextInfo=com.ibm.ws.request.timing.TestJDBC

[6/18/14 16:22:04:643 IST] 0000002a EventLoggingProbeExtension.class                    I END
requestID=AAADvUHkFwy-AAAAAAAAAAD # type=websphere.servlet.service # contextInfo=com.ibm.ws.request.timing.TestJDBC
# duration=2.614ms
```

JDBC events include
datasource and SQL

JDBC requests:

```
[6/18/14 16:21:43:727 IST] 0000002a EventLoggingProbeExtension.class                    I BEGIN
requestID=AAADvUHkFwy-AAAAAAAAAAD # type=websphere.datasource.executeUpdate # contextInfo=jdbc/exampleDS | create
table cities (name varchar(50) not null, population int, county varchar(30))

[6/18/14 16:21:44:200 IST] 0000002a EventLoggingProbeExtension.class                    I END
requestID=AAADvUHkFwy-AAAAAAAAAAD # type=websphere.datasource.executeUpdate # contextInfo=jdbc/exampleDS | create
table cities (name varchar(50) not null, population int, county varchar(30)) # duration=0.231ms
```

JBoss and Tomcat do not provide similar capability > increased admin labor for troubleshooting

IBM WebSphere Performance Tuning Toolkit

Find potential performance problems

- PTT shows detailed status of system with easily understood charts and forms. Users can analyze the performance data from various perspectives.
- PTT helps to find an error as soon as it occurs - monitor the servlet errors, transaction rollback, transaction timeout, JDBC connection timeout, thread hung, etc.

Accelerate performance tuning process

- User can tune many servers in one step in a centralized view by running tuning scripts within the workbench, download or upload performance related settings manually or via script

Health Check

- PTT can detect the performance decline and take actions automatically based on predefined rules. Rule engine detects the abnormal symptoms according to user defined rules (with ability to create and edit existing rules)

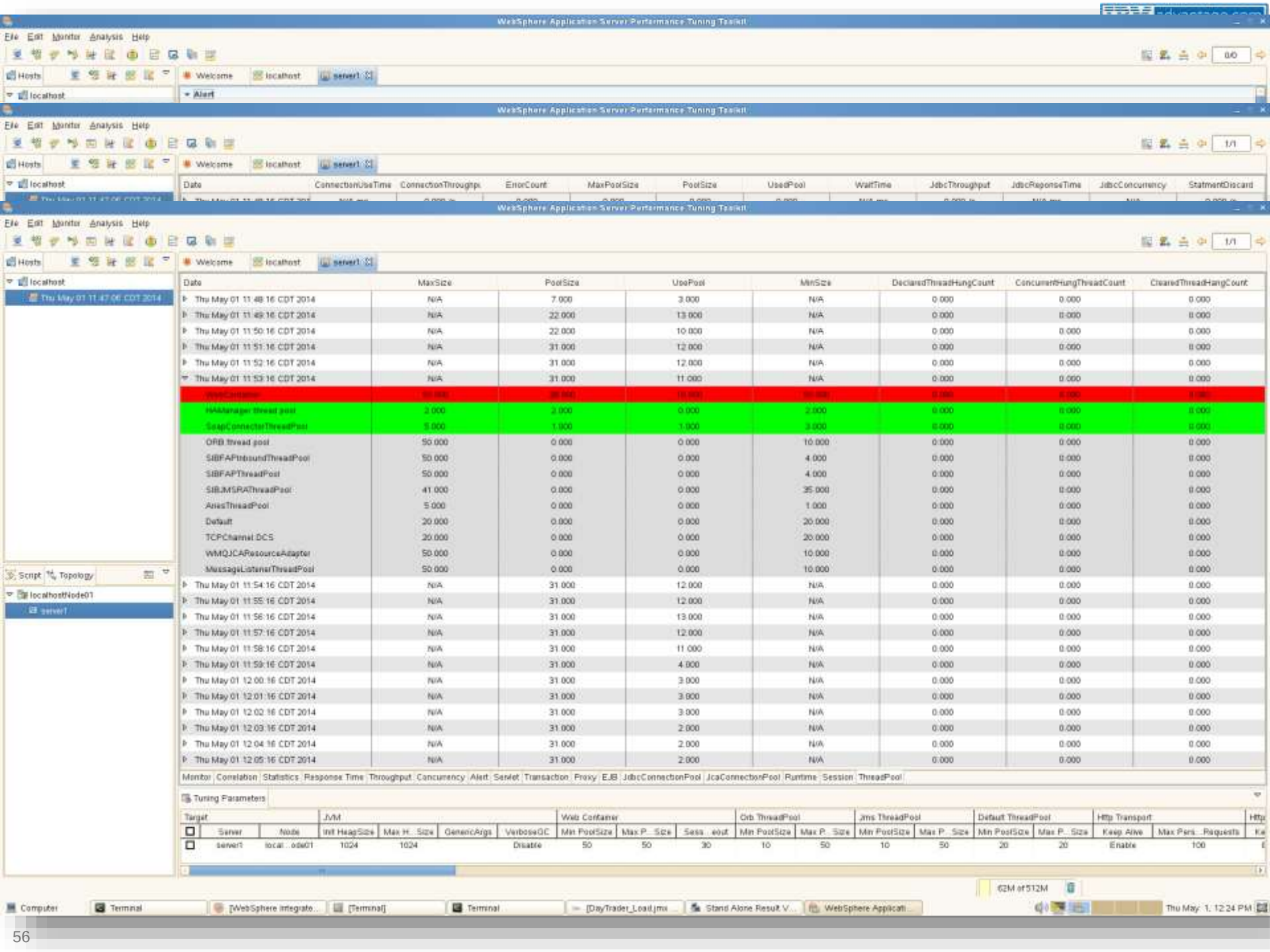
Operations to facilitate problems determination

- PTT can generate thread dump and heap dump for the JVM, enable trace settings, extract the connection pool contents

Report engine

- Online and offline analysis and reporting (generate, export and print report)

JBoss and Tomcat users spend considerably more effort finding all the right tuning variables. Monitoring data is scattered across multiple locations in the GUIs and custom JMX programs


WebSphere Application Server Performance Tuning Toolkit
File Edit Monitor Analysis Help
Hosts
localhost
Welcome
localhost
server1
Alert
Date
ConnectionUseTime
ConnectionThroughput
ErrorCount
MaxPoolSize
PoolSize
UsedPool
WaitTime
JdbcThroughput
JdbcResponseTime
JdbcConcurrency
StatmentDiscard
Thu May 01 11:48:16 CDT 2014
Thu May 01 11:49:16 CDT 2014
Thu May 01 11:50:16 CDT 2014
Thu May 01 11:51:16 CDT 2014
Thu May 01 11:52:16 CDT 2014
Thu May 01 11:53:16 CDT 2014
WebContainer
HAManager thread pool
SoapConnectorThreadPool
ORB.thread.pool
SIBFAPInboundThreadPool
SIBFAPThreadPool
SIBJMSRAThreadPool
AriesThreadPool
Default
TCPChannel.DCS
WMQJCAResourceAdapter
MessageListenerThreadPool
Thu May 01 11:54:16 CDT 2014
Thu May 01 11:55:16 CDT 2014
Thu May 01 11:56:16 CDT 2014
Thu May 01 11:57:16 CDT 2014
Thu May 01 11:58:16 CDT 2014
Thu May 01 11:59:16 CDT 2014
Thu May 01 12:00:16 CDT 2014
Thu May 01 12:01:16 CDT 2014
Thu May 01 12:02:16 CDT 2014
Thu May 01 12:03:16 CDT 2014
Thu May 01 12:04:16 CDT 2014
Thu May 01 12:05:16 CDT 2014
MaxSize
PoolSize
UsePool
MinSize
DeclaredThreadHungCount
ConcurrentHungThreadCount
ClearedThreadHangCount
Monitor Correlation Statistics Response Time Throughput Concurrency Alert Servlet Transaction Proxy EJB JdbcConnectionPool JcaConnectionPool Runtime Session ThreadPool
Tuning Parameters
Target
JVM
Web Container
Orb ThreadPool
Jms ThreadPool
Default ThreadPool
Http Transport
Server
Node
Init HeapSize
Max H... Size
GenericArgs
VerboseGC
Min PoolSize
Max P... Size
Sess...eout
Keep Alive
Max Pers...Requests
server1
Disable
Enable
62M of 512M
Computer
Terminal
[WebSphere Integrato...
[Terminal]
Terminal
[DayTrader_Load.jmx ...
Stand Alone Result V...
WebSphere Applicati...
Thu May 1, 12:24 PM

Problem determination and health management

	WAS	WAS Liberty	Oracle WLS	JBoss EAP	Apache Tomcat
Hang, crash, memory management and dump analysis	Yes	Yes	Yes	no	no
GC and Memory monitoring and analysis	Yes	Yes	Yes	no	no
Stuck thread detection and reporting	Yes	Yes	Yes	?	no
Troubleshoot memory leaks and excessive heap consumption	Yes	Yes	Yes	?	no
Deep real time monitoring of running virtual machines	Yes	Yes	Yes	?	no
Analyzer for Java heap dump	Yes	Yes	Yes	?	no
Trace and request analyzer for HTTP and app server traces	Yes	Yes	Yes	?	no
Detect improper web plug-in configurations	Yes	Yes	no	no	no
Troubleshoot JDBC connection pools	Yes	Yes	Yes	?	no
Correlate logs from different products, get fix recommendations	Yes	Yes	no	no	no
Guided troubleshooter - Guides you through solving problems	Yes	Yes	no	no	no
Find Java threads that are excessively consuming processor resources	Yes	Yes	Yes	?	no
Request timing – detect slow or hung requests the first time they happen	Yes	Yes	Yes	?	no
App class data sharing for smaller memory footprint and faster startup	Yes	Yes	no	no	no
System class data sharing for reduced memory and faster startup	Yes	Yes	Client	Client	Client
Server maintenance mode – allow only affinity traffic	Yes	no	Yes	?	no
Server maintenance mode – do not allow any traffic	Yes	Yes	Yes	?	no

Liberty support for z/OS

Security

- zosSecurity feature - integrates with SAF based security (RACF, ACF2, Top Secret)

WLM

- zosWLM feature - integrates with z/OS WLM - for workload classification and reporting -- can use RMF-based tools to monitor/track/report on performance

Transactions

- zosTransactions feature -integrates with z/OS RRS - for consolidated resource recovery between WAS and DB2 (2PC between subsystems)

Adapters

- zosLocalAdapters feature- this is WOLA - so you can tightly integrate Liberty apps with existing CICS and batch applications on z/OS (high performance local cross memory connectivity)

REST

- zosConnect feature -REST/JSON gateway for interacting with CICS/IMS/batch apps on z/OS from API callers

QoS

- Liberty is optimized and made to work with more qualities of service on z/OS -- to make the production environment more capable

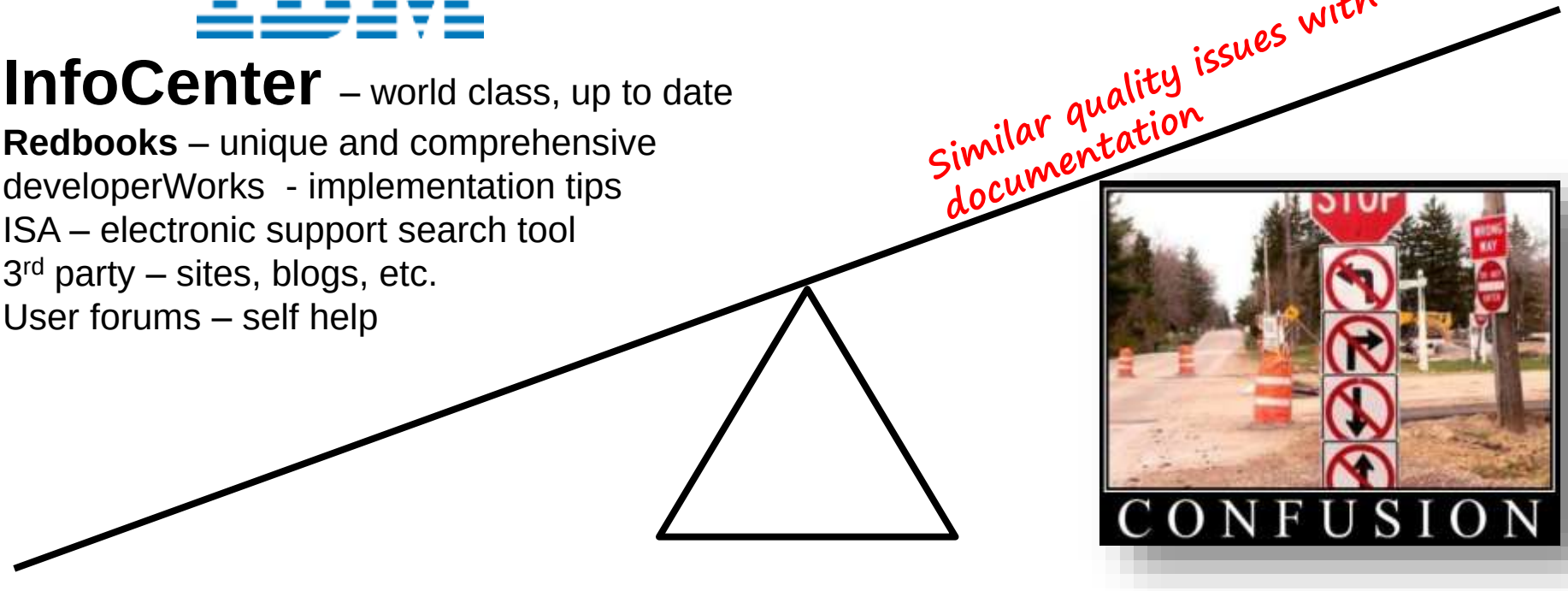
None of these capabilities are available with Tomcat, JBoss or WebLogic

Documentation

Order of magnitude difference in quality

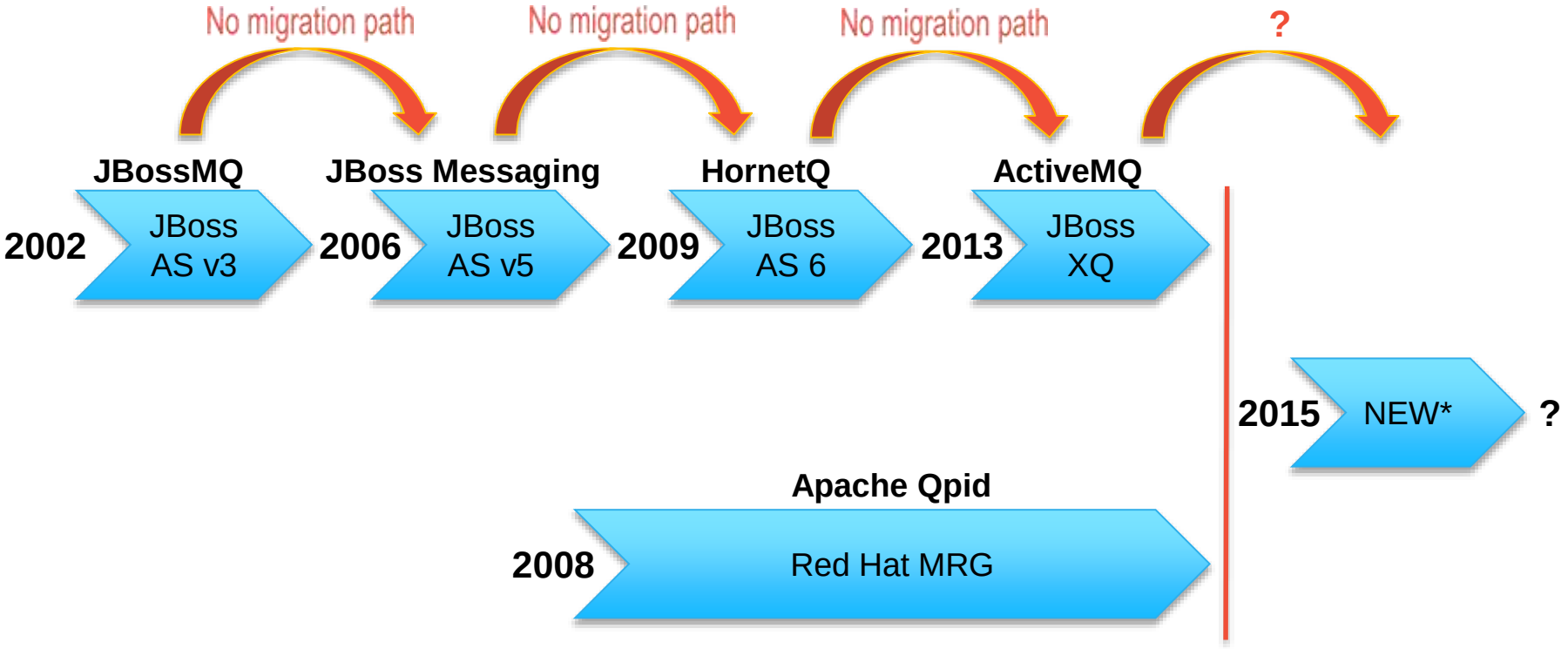
JBoss docs – limited and inconsistent, lags in time
JBoss wikis – lots of old confusing info
User forums – no longer actively monitored by developers

InfoCenter – world class, up to date
Redbooks – unique and comprehensive
developerWorks - implementation tips
ISA – electronic support search tool
3rd party – sites, blogs, etc.
User forums – self help



The history of Red Hat and JBoss messaging

Where change is the only constant



* - New Red Hat “strategic” messaging is described to be a REWRITE and a combination of “best ideas” from Apache Qpid + Red Hat HornetQ + Apache ActiveMQ

Platforms support

		WAS 8.5.5.7	WebLogic 12.2.1	JBoss EAP 6.4	tc Server
X86	Red Hat Enterprise Linux	5, 6, 7	6, 7	6, 7	6, 7
	Asianux	3	no	no	no
	Ubuntu	12, 14	no	no	Dev
	Oracle Linux	no	6, 7	no	no
	Mac OS	Liberty	Dev.	no	Dev
	SuSe Linux ES	10, 11, 12	11	no	V11
	Windows	7+, 2008+	2012	2008+	2008
	Solaris	10, 11	11	11	no
RISC	Red Hat Enterprise Linux	5, 6, 7	6	7	no
	SuSe Linux ES	10, 11, 12	11	no	no
	Ubuntu	14	no	no	no
	AIX	6, 7	7	no	no
	IBM I	6, 7	no	no	no
	HP-UX	11i v2, v3	11i v3	11i v3	no
	Inspur K-UX (Itanium)	2.1	no	no	no
	Solaris (SPARC)	10, 11	11	11	no
z/Series	Red Hat Enterprise Linux	5, 6, 7	6	no	no
	SuSe Linux ES	10, 11, 12	11	no	no
	z/OS	1, 2	no	no	no

WebSphere Application Server Migration Toolkit

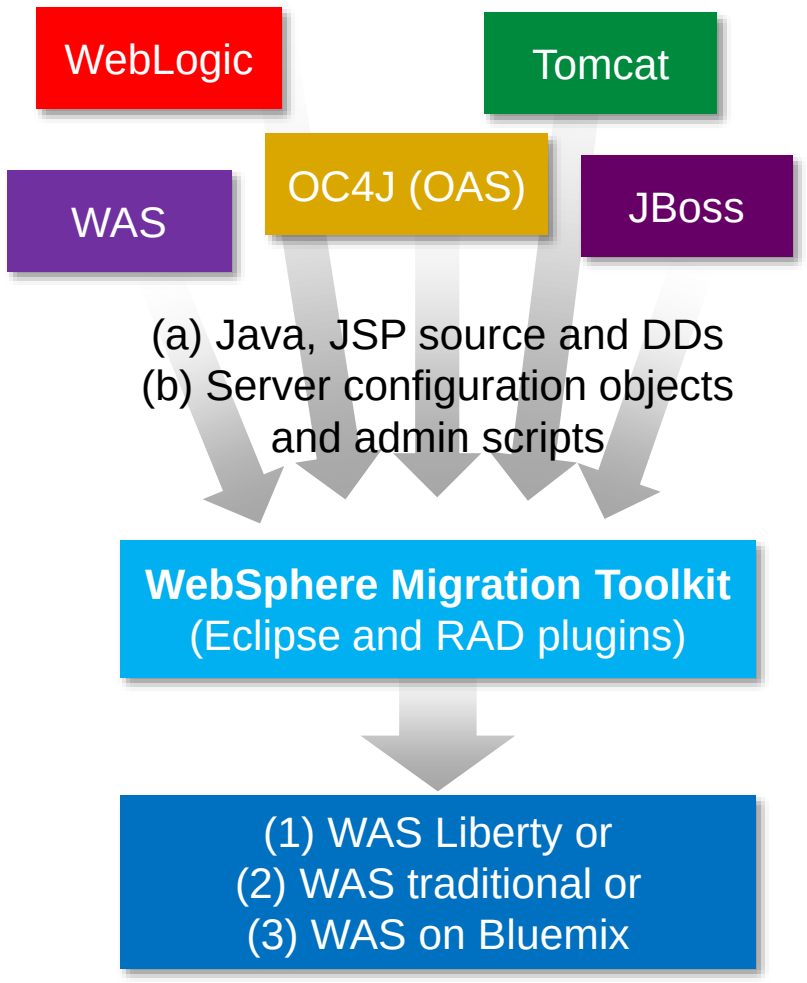
Now easier then ever before to migrate your applications to WAS

No Charge plugins for Eclipse and RAD

- Rule sets for multiple source / destination combinations (e.g. WLS->WAS, etc.):
- (a) The tool scans Java source code, JSP files and deployment descriptors and identifies the changes required (allows for Java upgrade also).
- (b) The tool scans server configuration files (looking for Datasources, servers, JMS settings, etc.) and generates appropriate Liberty or WAS configuration.
- In most cases the toolkit is capable of making the application changes itself. After the “scan” and “conversion” are done the toolkit generates report on the results of the migration and any manual migration tasks (if required).

Free migration RedBook and developerWorks articles on migration

No Charge Migration Assessment Workshop for qualified customers



IBM migration tools and offerings: http://whywebsphere.com/?s=migration

Online WebSphere Migration Discovery Tool

http://ibm.biz/migration-discovery

Subscribe to our **external** competitive blog