

JXINSIGHT PROBES

JXInsight Probes is an extensible resource metering framework, with an open API, that allows developers and performance engineering staff extend the profiling and metric monitoring capabilities of JXInsight to custom components and frameworks.

This extremely lightweight, fast and extensible runtime has explicit support for aggregation of probe resource consumption across custom hierarchical groups with out of the box support for metering of numerous resource counters including high resolution clock time, thread cpu time, thread monitor waiting, thread monitoring blocking, gc times and counts.

Probes Meters

JXInsight's resource metering framework, *Probes*, is significantly different than any other typical code profiler or business transaction (request) path tracer in that it can be configured to collect data from multiple resource meters per probe measurement interval.

Instead of collecting either the clock time or cpu time, the standard option with other tools, the probes runtime can record meter readings from a number of sources such as JXInsight's native profiling agent as well as the Java 5 Management and Monitoring API.

Metering Strategies

JXInsight provides an innovative approach to metering that allows for multiple profiling strategies to be applied during a measurement window, such as sampling every Nth probe or spacing the time interval between each probe metered.

Truly unique is the ability to configure each profiling strategy separately and chain them together.

JXInsight includes the following metering strategies: Hotspot, Random, Exclude, Include, Frequency, Interval, Warmup, Burst, Sample, and Delay.

On-Demand

JXInsight's Probes can be dynamically enabled or disabled from within its application performance management and monitoring console as well as via its command line interface allowing production systems to run at full speed.

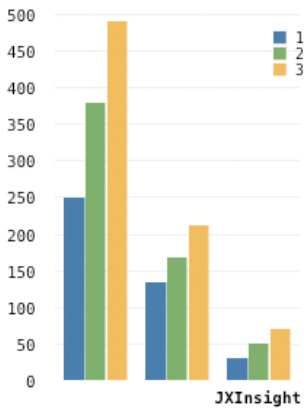
Probe					
Metering Table					
	Probe	Meter	Count	Total	Total (I)
	<>	clock.time	*754	*25,618,726	*25,618,726
	org	clock.time	539	25,152,077	17,128,015
	gridgain	clock.time	539	25,152,077	17,128,015
	grid	clock.time	536	25,147,408	17,123,346
	examples	clock.time	3	4,669	4,669
	com	clock.time	215	8,490,711	8,490,711
	tangosol	clock.time	215	8,490,711	8,490,711
	coherence	clock.time	195	8,490,226	8,490,226
	net	clock.time	20	485	485
	<>	cpu.time	*754	9,260,689	9,260,689
	org	cpu.time	539	9,218,584	6,779,130
	gridgain	cpu.time	539	9,218,584	6,779,130
	grid	cpu.time	536	9,217,316	6,777,862
	examples	cpu.time	3	1,268	1,268
	com	cpu.time	215	2,481,559	2,481,559
	tangosol	cpu.time	215	2,481,559	2,481,559
	coherence	cpu.time	195	2,481,063	2,481,063
	net	cpu.time	20	496	496
	<>	waiting.time	*754	5,865,017	5,865,017
	org	waiting.time	539	5,634,368	1,881,421
	gridgain	waiting.time	539	5,634,368	1,881,421
	grid	waiting.time	536	5,634,368	1,881,421
	examples	waiting.time	3		
	com	waiting.time	215	3,983,596	3,983,596
	tangosol	waiting.time	215	3,983,596	3,983,596
	coherence	waiting.time	195	3,983,596	3,983,596
	net	waiting.time	20		

Metering Strategy Scopes

Metering strategies can be classified by the scope at which the determination to meter a fired probe is made. Name based strategies (frequency, interval, warmup) decide base on data pertaining to an actual probe name (Group).

The decision of global based strategies (burst, sample, delay) does not necessarily pertain to the firing of a particular probe name.

Profiling Benchmark By Call Depth



Production Ready

In benchmarks with other competitive solutions JXInsight's Probes outperforms all others by a significant margin with the nearest solution being between 3-4 times slower than JXInsight and the second nearest being between 8-10 times slower.

JXInsight Probes is the first truly production ready enterprise Java performance management solution that combines the extremely low overhead of sampling with high precision measurement accuracy across multiple resource meters.

Grid & Cloud Computing Ready

JXInsight's Probes resource metering analysis can span multiple hosts, processes and threads - context. Within each metering context (cluster, host, process, and thread) resource usage statistics including count, total, maximum, and minimums are listed per probe group (i.e. package, class, or method).

From within the JXInsight application performance management and monitoring console a single view of an entire distributed code base and execution flow can be viewed, filtered, sorted, analyzed, annotated, archived, and exported.

The ability to reset and mark the metering for all probe groups within all runtimes part of a grid makes JXInsight the ideal software execution analysis solution during development, test, pre-production and production. Dispatch a service request to a grid, enterprise service bus or similar middleware technology and watch the execution flow via resource metering changes per runtime and by package, class, method or similar execution abstraction (url, service endpoint, or business process workflow context).

With JXInsight's Probes cluster wide marking capability it is even possible to reverse engineer the execution dependency across components of complex distributed system.

Metering Table						
	Probe	Context	Meter	Count	Total	Total (I)
	<>		clock.time	*7,098	*168,100,209	*168,100,209
	<>	mac.local	clock.time	*7,098	*168,100,209	*168,100,209
	com		clock.time	3,883	36,952,251	36,952,251
	org		clock.time	3,236	158,908,024	131,168,465
	<>		cpu.time	*7,098	41,521,543	41,521,543
	<>	mac.local	cpu.time	*7,098	41,521,543	41,521,543
	<>	1517	cpu.time	2,259	14,023,535	14,023,535
	<>	1516	cpu.time	2,206	13,776,574	13,776,574
	<>	1515	cpu.time	2,633	13,721,434	13,721,434
	com	mac.local	cpu.time	3,883	9,565,506	9,565,506
	org	mac.local	cpu.time	3,236	41,014,435	31,958,108
	com		cpu.time	3,883	9,565,506	9,565,506
	org		cpu.time	3,236	41,014,435	31,958,108
	<>		waiting.time	*7,098	40,425,063	40,425,063
	<>	mac.local	waiting.time	*7,098	40,425,063	40,425,063
	<>	1517	waiting.time	2,259	30,491,167	30,491,167
	<>	1515	waiting.time	2,633	6,766,642	6,766,642
	<>	1516	waiting.time	2,206	3,167,254	3,167,254
	com	mac.local	waiting.time	3,883	10,952,655	10,952,655
	org	mac.local	waiting.time	3,236	35,107,976	29,482,690
	com		waiting.time	3,883	10,952,655	10,952,655
	org		waiting.time	3,236	35,107,976	29,482,690
	<>		jmgmt.gc.time	*7,098	14,445,000	14,445,000
	<>	mac.local	jmgmt.gc.time	*7,098	14,445,000	14,445,000
	<>	1517	jmgmt.gc.time	2,259	6,515,000	6,515,000
	<>	1515	jmgmt.gc.time	2,633	4,197,000	4,197,000
	<>	1516	jmgmt.gc.time	2,206	3,733,000	3,733,000
	com	mac.local	jmgmt.gc.time	3,883	7,312,000	7,312,000
	org	mac.local	jmgmt.gc.time	3,236	13,892,000	7,133,000
	com		jmgmt.gc.time	3,883	7,312,000	7,312,000
	org		jmgmt.gc.time	3,236	13,892,000	7,133,000
	<>		blocking.time	*7,098	1,032,302	1,032,302
	<>	mac.local	blocking.time	*7,098	1,032,302	1,032,302
	<>	1517	blocking.time	2,259	490,476	490,476
	<>	1516	blocking.time	2,206	488,138	488,138
	<>	1515	blocking.time	2,633	53,688	53,688
	com	mac.local	blocking.time	3,883	641,474	641,474
	org	mac.local	blocking.time	3,236	441,944	390,828
	com		blocking.time	3,883	641,474	641,474
	org		blocking.time	3,236	441,944	390,828

Extension Packs

JXInsight comes with an incredible number of extension packs for all major technology standards, platforms, frameworks and products including: JMS, JDBC, JCR, JTS, JTA, JSP, Java Servlet, JSF, JCA, JAX-RPC, JAX-WS, JNDI, EJB, JPA, JMX, JavaSpaces, JINI, RMI, CORBA, JBoss Remoting, WebLogic RMI, Oracle Coherence, GridGain, IBM ObjectGrid, JBoss Seam, Google Guice, Spring, OSGi, WebLogic EventServer, Jetty (Comet), Apache Tomcat, Apache Struts, Apache Lucene, CommonJ, and IBM MQ.