

TARGET ENVIRONMENT DETAILS FOR JAVA

This section is designed to assist with the collection and review of the target environment and the components contained within. The level of details required within the section are important in ensuring that your organization's time is well spent, and that your efforts deliver measured results. Please make sure that all technical components within the target environment are accounted for.

Application Name: <i>Liquidity Risk</i>			
Application Description: <i>Our Liquidity Risk LRI AGG ENG is a multi-node distributed synchronized grid that joins quantitative risk algorithms to the bank's capital markets positions by asset class and computes highly-accurate statements of the form "position=P on Asset Class=AC has Liquidity Risk Indicator=LRI". The AGG ENG produces these statements in real time. The system renders these statements using a tiered-ply MAP-REDUCE strategy.</i>			
Application Environment (Cloud, On-Premise, Both): <i>On-Premise</i>			
POC Environment (pre-production, production): <i>pre-production</i>			
Is the application a web application, a standalone executable or a combination of the different types? Standalone Java VM executables			
No. of JVMs to be instrumented in the POC environment: <i>8 - 40</i>			
JDK Version: <i>1.7</i>			
Application Server/Java Container: <i>JBOSS Infinispan</i>		Version: <i>5.x</i>	
Operating System: <i>Linux</i>		32 or 64bit: <i>64</i>	
Application Components Used (y/n)			
Struts:	Servlets:	Spring:	RMI: <i>YES</i>
Web Services:	Messaging:	EJB:	Other:
Application Tiers (describe all tiers): <i>Master (1 JVM), Mapper (7 JVM), Reducer (???) - Dynamically configurable by symmetry. 1 Master → N Reducers → 3xN Mappers. E.g. 1 → 2 → 6 (smallest possible MR grid assignment) 1->10->30 (largest possible MR grid assignment).</i>			
Describe all communication protocols for the above tiers (inbound and outbound): <i>Infiniband (40GB/s) physical network bridging Oracle EXADATA hardware data appliance to Linux Hosted Java 1.7 VM grid using SOKETS DIRECT PROTOCOL (bridge to native Infiband VERBS).</i>			
Describe any messaging frameworks used in the application (Active MQ, Rabbit MQ, etc.): <i>None. Everything is now replaced with native Infiniband via Java 1.7 SDP.</i>			
Describe any ESBs used in the application (e.g. Mule, Tibco, etc.): <i>NONE</i>			
Logging Framework & Version (e.g. Log4J): <i>Log4j</i>			
Describe any application integration frameworks/products used in the application: <i>None. 100% pure Java. Using the Infinispan distributed Data Grid product for Map<K,V> custody and DistributedExecutor, NotifyingFuture dispatch.</i>			

Describe any databases, versions and JDBC drivers used in the application: Oracle EXADATA appliance (hardware RDBMS).	
Describe any JavaScript Frameworks used in the Application (jQuery, Moo, etc.): None	
Application started by services manager (Windows OS only)? N/A	
Firewalls or proxies between the application and location of the AppDynamics controller? No	
AppDynamics controller location (on-premise or SaaS): On-premise	AppDynamics Controller Performance Profile (s, m, l, xl, h): Small

[Supported Environments and Versions](#)