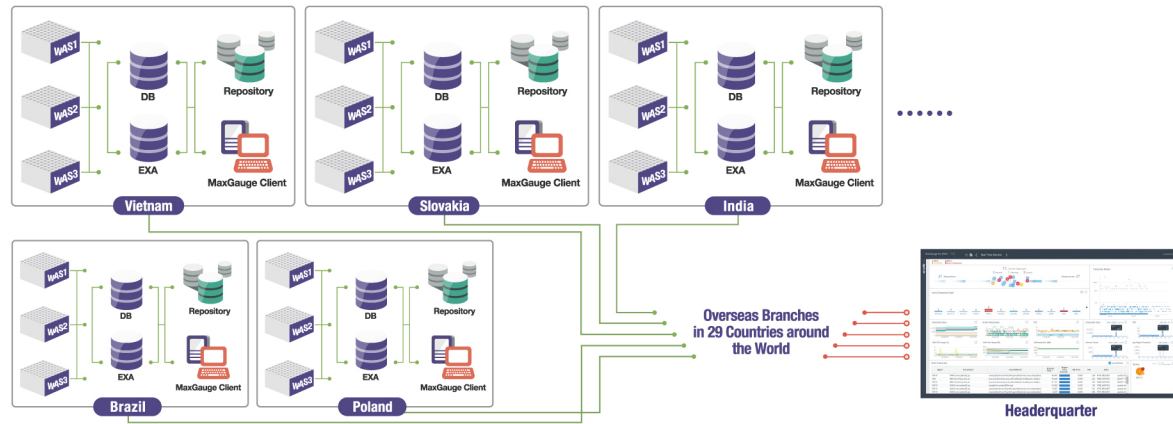


Listen to Your Application  
Any Time, Any Place

Best Practice

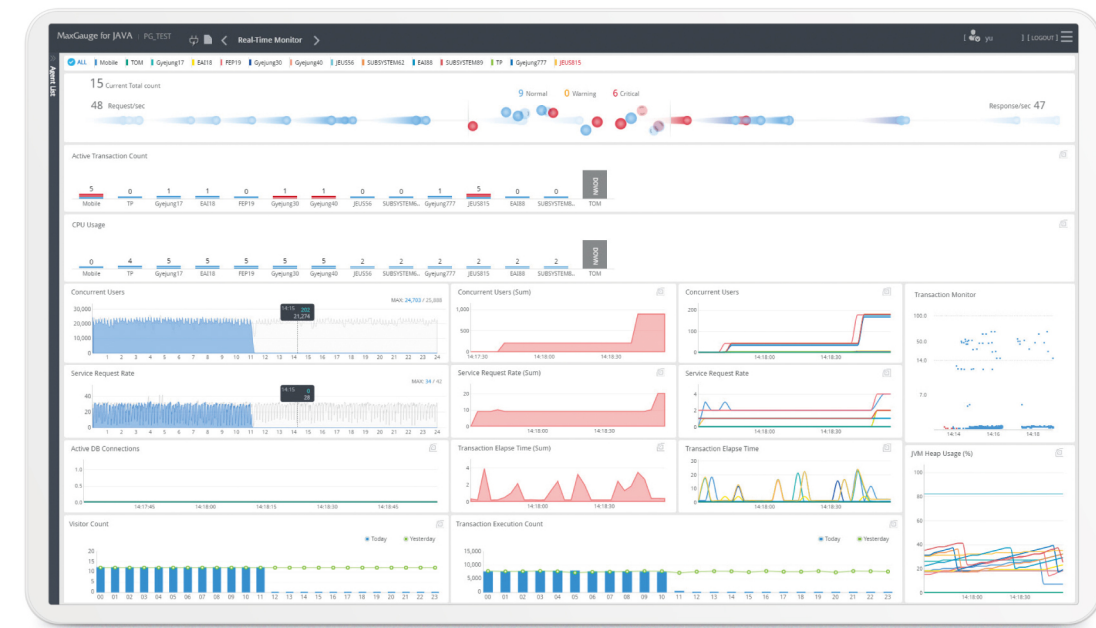
I System Setup



Integrated Solution for WAS Performance Management

MaxGauge for JAVA

MaxGauge for JAVA is a unique application performance management (APM) solution providing linkage analysis with database session at transaction level. With its powerful monitoring and analytical capabilities, MaxGauge allows you to manage applications in the best possible manner with the least waste of time and resources.



Real-Time Monitoring

Collects all performance stats at every second and provides real-time monitoring at a minimum load.

- Active Transaction Monitor
- Transaction Response Time Monitor
- Performance Stat
- Service Stat
- Active Transaction List
- DB Monitor
- Connection Pool Monitor
- GC Stat



- System : Global MES (Manufacturing Execution System) in 29 Countries
- Average Transactions
  - Manufacturing Part : 700~1,000TPS
  - Mobile Part : 1,500~2,000 TPS

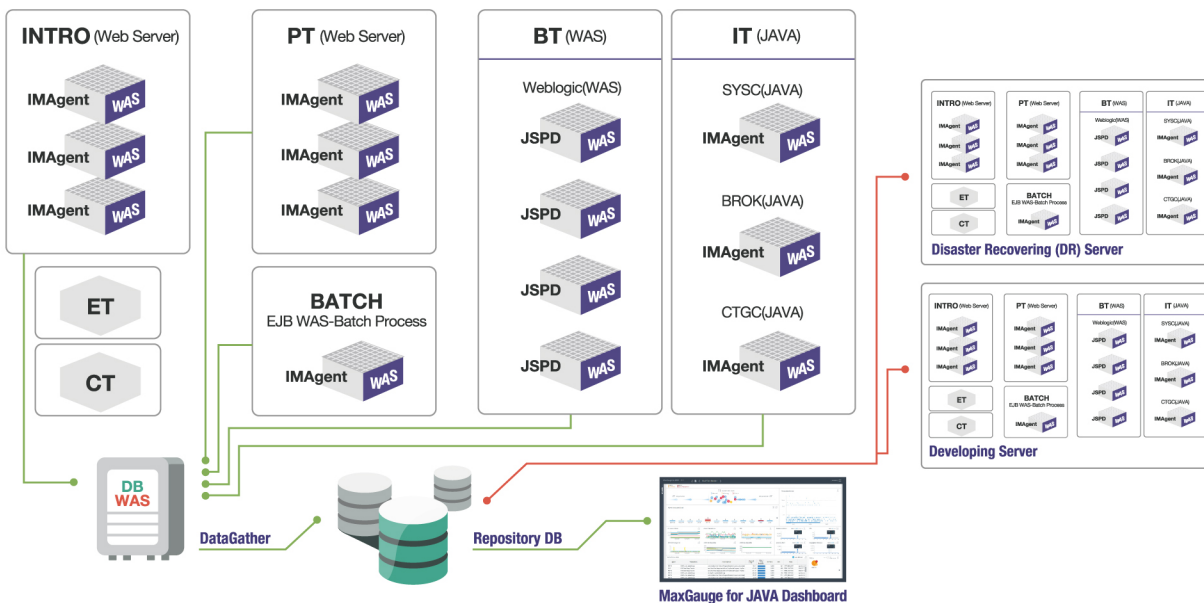
MaxGauge Products

- MaxGauge for Oracle
- MaxGauge for SQL Server
- MaxGauge for JAVA

Remarkable Effect

- Integrated monitoring on the huge number of systems in many countries
- Immediately pinpointing any performance issue of the production systems
- Monitoring load balance among servers under multi-tier environment

I System Setup



- System : Online Banking System
- Peak Time Transactions : 2,100 TPS
- Average Transactions : 1,000 TPS

MaxGauge Products

- MaxGauge for Oracle
- MaxGauge for Java
- MaxGauge for E2E

Remarkable Effect

- Immediately pinpointing the root cause of any performance issue during developing and operational tasks of online banking
- Reducing the troubleshooting time and preventing repeated issues
- Enabling the detailed management at service level
- Improving the quality of customer services

# Performance Analyzer

## Transaction Path View

The path flow of all transactions running in the application is automatically displayed on the screen. The user can identify the delay period based on the elapsed time and analyze the root cause in detail with just a few clicks.

Transaction Response Time: 36 seconds out of 39 seconds were spent in J772 Web Apps (Java)

Class	Method	Execute Count	Elapse Time	Exception C...	Elapse Ratio	Method Type	Method SEQ
sample/EchoServerThread	StreamProc()	1	36.83	0	100%	loop.io.net	1
sample/EchoCommon	sleep(int)	1	27.892	0	76%		2
sample/weave/LoopDepth	run()	10	0.001	0	0%		3
sample/weave/LoopDepth	depth001()	10	0.001	0	0%		
sample/weave/LoopDepth	depth002()	10	0.001	0	0%		
sample/weave/LoopDepth	depth003()	10	0.001	0	0%		
sample/weave/LoopDepth	depth004()	10	0.001	0	0%		
sample/weave/LoopDepth	depth005()	10	0.001	0	0%		
sample/weave/LoopDepth	depth006()	10	0.001	0	0%		
sample/weave/LoopDepth	depth007()	10	0.001	0	0%		
sample/weave/LoopDepth	depth008()	10	0.001	0	0%		
sample/weave/LoopDepth	depth009()	10	0.001	0	0%		
sample/EchoServerThread	clientSocket(EchoCli...	2	8.936	0	24%	io.net	15
sample/Common	close(InputStream, O...	4	0.001	0	0%	io.net	16

If you look at the corresponding Web Apps Call Tree, you can identify that 27 seconds out of 36 seconds were generated in Echo Common Class' sleep() function.

Source View

```

package ltk.mca.res;
import ltk.mca.res.*;
import ltk.res.*;

public class MCA_SVC {

    public String request(final byte index, final String service) throws Exception {
        final String url = "http://192.168.123.241:6000/QA/IN/grejeong:jsp?SERVICE=GYE_";
        final String gid = "MCD:0410101010101";
        final StringBuffer result = new StringBuffer();
        String message = "";
        try {
            final MCA mca = new MCA();
            mca.process(service);
            switch (index) {
                case 1: {
                    message = "SVC0001 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 2: {
                    message = "SVC0002 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 3: {
                    message = "SVC0003 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 4: {
                    message = "SVC0004 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 5: {
                    message = "SVC0005 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 6: {
                    message = "SVC0006 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 7: {
                    message = "SVC0007 FROM VIRTUAL MCA(normal)";
                    break;
                }
                case 8: {
                    message = "SVC0008 FROM VIRTUAL MCA(normal)";
                    break;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return result.toString();
    }
}
    
```

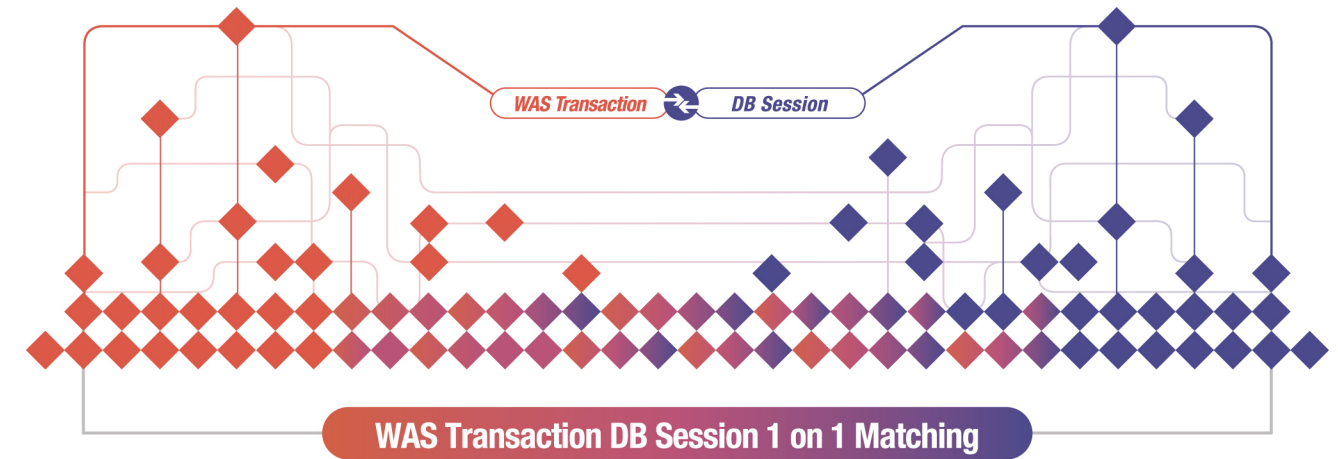
Source Diff View

```

diff --git a/MCA_SVC.java b/MCA_SVC.java
index 1234567..8901234
--- a/MCA_SVC.java
+++ b/MCA_SVC.java
@@ -1,4 +1,4 @@
 package ltk.mca.res;
 import ltk.mca.res.*;
 import ltk.res.*;
    
```

## Web Apps(Java) Transaction DB Session 1:1 Match

You can easily identify the root cause of performance issues by 1:1 matching of the application transaction with the DB session information generated by the transaction.



Database session information is provided from WAS active transaction list simultaneously

Main performance indicators are provided such as active session and lock tree

Recognize the DB performance time, work amount and wait events at once

Database is traced to identify which session performs an execution thread of a particular transaction

## DB Monitoring

8 Current Total count  
33 Requested  
Response: 33

Active Transaction Count: 5 Normal, 0 Warning, 6 Critical

CPU Usage(%)

Concurrent Users

JVM CPU Usage (%)

JVM Free Heap (MB)

JVM Heap Size (MB)

Transaction Monitor

Concurrent Users

Execute Count

Avg Elapse Time(sec)

DB Lock Holder and Waiter information caused by Transaction

SID	Hold Lock Type	Wait Lock Type	Request Mode	Agent	Transaction	Elapsed Time	Instance	SQL 1	Logical Reads	Physical Reads	Wait Info
37	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	14.993	ORCL	update xm_customer set customer_name='abc'	2101333	199	eng: CR - block range reuse (dga - 6762) [Exclusive 305569 (R)   WAITING (1323) (0)]
43	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	7.965	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
31	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	9.974	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
44	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	8.716	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
29	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	4.224	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
37	CR	Exclusive	CR	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	14.993	ORCL	update xm_customer set customer_name='abc'	2101333	199	eng: CR - block range reuse (dga - 6762) [Exclusive 305569 (R)   WAITING (1323) (0)]
49	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	5.716	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
46	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	12.47	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
41	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	11.715	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
55	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	14.347	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]
42	TX	Exclusive	TX	TOMC_LOCK	/xeem7/jpp/customer/wait_1.jsp	11.485	ORCL	select * from xm_customer where customer_name='BATKOREA' for update	0	0	eng: TX - row lock contention - 8488 [Exclusive user=16   slot524290 sequence0007...]

Web Apps(Java) Transaction

DB Session Information