



Servicio Andaluz de Salud
CONSEJERÍA DE SALUD

*Oficina Técnica para la Gestión y Supervisión de
Servicios TIC
Subdirección de Tecnologías de la Información*

Métricas de referencia para Tecnologías Oracle

Referencia documento: InfV5_JASAS_AR_Monitorizacion_v920.doc

Fecha: 16 de noviembre de 2018

Versión: 9.2.0

Registro de Cambios

Fecha	Autor	Versión	Notas
12 de Enero de 2011	Pablo Fernández	3.1.0	Versión inicial
14 de Marzo de 2013	Pablo Fernández	4.1	Revisión de Marzo de 2013, contrato 2012-2014
13 de Junio de 2013	Pablo Fernández	4.2	Revisión de Junio de 2013, contrato 2012-2014
17 de Octubre de 2013	Pablo Fernández	4.3	Revisión de Octubre de 2013
16 de Julio de 2015	Pablo Fernández	6.1	Revisión de Julio de 2015
16 de Diciembre de 2015	Pablo Fernández	6.2	Revisión de Diciembre de 2015
16 de Junio de 2016	Pablo Fernández	7.1	Revisión de Junio de 2016
16 de Noviembre de 2016	Pablo Fernández	7.2	Revisión de Noviembre de 2016
16 de Junio de 2017	Pablo Fernández	8.1	Revisión de Junio de 2017
16 de Noviembre de 2017	Pablo Fernández	8.2	Revisión de Noviembre de 2017
16 de Junio de 2018	Pablo Fernández	9.1	Revisión de Junio de 2018
16 de Noviembre de 2018	Jonathan Ortiz	9.2	Revisión de Noviembre de 2018

Revisiones

Nombre	Role
Jonathan Ortiz	ACS Service Engineer
Gregorio Adame	ACS Service Engineer
José María Gómez	ACS Service TAM

Distribución

Copia	Nombre	Empresa
1	Subdirección de Tecnologías de la Información	Consejería de Salud, Junta de Andalucía
2	Servicio de Coordinación de Informática de la Consejería de Hacienda y Administración Pública	Consejería de Hacienda y Administración Pública, Junta de Andalucía

Índice de Contenidos

CONTROL DE CAMBIOS	4
INTRODUCCIÓN	5
MÉTRICAS DE NEGOCIO	6
ORACLE NET LISTENER	7
<i>Configuración monitorización</i>	<i>7</i>
<i>Recolección de métricas de producto</i>	<i>8</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>9</i>
ORACLE DATABASE SINGLE INSTANCE	10
<i>Configuración monitorización</i>	<i>10</i>
<i>Recolección de métricas de producto</i>	<i>11</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>15</i>
ORACLE REAL APPLICATION CLUSTERS	17
<i>Configuración monitorización</i>	<i>17</i>
<i>Recolección de métricas de producto</i>	<i>17</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>18</i>
ORACLE STREAMS	19
<i>Configuración monitorización</i>	<i>19</i>
<i>Recolección de métricas de producto</i>	<i>19</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>20</i>
ORACLE DATAGUARD	21
<i>Configuración monitorización</i>	<i>21</i>
<i>Recolección de métricas de producto</i>	<i>22</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>23</i>
ORACLE JVM JROCKIT Y WEBLOGIC SERVER	24
<i>Configuración monitorización</i>	<i>25</i>
<i>Recolección de métricas de producto</i>	<i>25</i>
<i>Matriz Resultados Métricas de producto / Métricas de negocio</i>	<i>29</i>
TRATAMIENTO ESTADÍSTICO DE RESULTADOS	32

Control de cambios

Cambio	Descripción	Página
	No se realizan cambios en esta versión	

Introducción

Este documento recoge un listado de las métricas a recolectar sobre cada una de las tecnologías de referencia del Servicio Andaluz de Salud (SAS) de la Junta de Andalucía durante el proceso de pruebas de sistema, carga o funcionales.

Métricas de Negocio

Una métrica de negocio es cualquier tipo de instrumento de medición usado para calcular los componentes cuantificables del rendimiento de una compañía, como el retorno de la inversión, los ratios de rotación de empleados y clientes, el beneficio antes de impuestos, etc.

Las métricas de negocio constituyen gran parte del área de business intelligence, que comprende un alto número de aplicaciones y tecnologías para recoger, almacenar, analizar y proporcionar acceso a los datos para ayudar a las empresas a realizar mejores decisiones de negocio. Existen aproximaciones sistemáticas como los cuadros de valoración que pueden ser empleados para transformar los objetivos de la organización dentro de objetivos cuantificables y así monitorizar el rendimiento de la organización en términos de la consecución de dichos objetivos.

A partir de la definición anterior y aplicada al entorno tecnológico del Servicio Andaluz de Salud, se presentan las métricas que se aplicarán como parámetros de entrada a las pruebas de carga que validan un nuevo desarrollo o una nueva funcionalidad de aplicación. Las métricas de negocio tomadas como parámetros de entrada establecerán la tipología de las pruebas de carga y serán los valores sobre los que se calculen los ratios resultantes al cruzarlos con las métricas de productos Oracle en los que se ejecutan las aplicaciones.

Oracle Soporte ha establecido las siguientes métricas de negocio.

#user – valor numérico, define el número de usuarios con los que se ejecutarán las pruebas de carga.

%concurrency_usuarios – valor numérico, define el porcentaje de usuarios sobre el número total de usuarios, que ejecutan la lógica de las pruebas de carga de forma concurrente.

#iteraciones_casouso_i_transaccional – valor numérico, define el número de iteraciones que un Caso de Uso transaccional es ejecutado por un usuario.

%ejecucion_casouso_i – valor numérico, es el porcentaje de ejecuciones del caso de uso i sobre el total de ejecuciones del resto de casos de uso de las pruebas de carga

#registros_procesados_casouso_i_masivo – valor numérico, define el número de registros que un Caso de Uso masivo tiene que procesar correctamente durante las pruebas.

%concurrency_casouso – valor numérico, dado un caso de uso i, el porcentaje de concurrencia define el número de ejecuciones concurrentes de un caso de uso i sobre el total de ejecuciones de dicho caso de uso.

tiempo_in – valor numérico, tiempo que estarán ejecutándose las pruebas.

tiempo_out – valor numérico, tiempo empleado en procesar los casos de uso establecidos en la definición de las pruebas.

Oracle Net Listener

Configuración monitorización

Cada componente de *Oracle Net Services* produce su propio fichero de traza, la siguiente tabla muestra los nombres de los ficheros de traza por defecto y el componente que genera dicho fichero de traza.

Trace File	Component
instance-name_pid.trc	Oracle Connection Manager listener
instance-name_cmgtw_pid.trc	Oracle Connection Manager CMGW (Connection Manager gateway) process
instance-name_cmadmin_pid.trc	Oracle Connection Manager CMADMIN (Connection Manager Administration) process
listener.trc	Listener
sqlnet.trc	Client
svr_pid.trc	Database Server
tnsping.trc	TNSPING Utility

Los parámetros que controlan la configuración del trazo incluyendo el tipo y la cantidad de información de traza, son configurados por los ficheros de configuración de cada uno de los componentes, listados en la siguiente tabla. Para el caso que nos ocupa sólo nos centraremos en el fichero *listener.ora*.

Componente	Fichero de Configuración
Oracle Connection Manager Processes	cman.ora
Listener	listener.ora
Client	sqlnet.ora
Database Server	sqlnet.ora
TNSPING Utility	sqlnet.ora

Los parámetros válidos para la configuración de la traza del *listener* son los siguientes:

Parámetros listener.ora	Descripción
TRACE_LEVEL_listener_name	Especifica el nivel de detalle de traza que se almacena para el listener. El valor del trace leve puede ser un valor dentro del rango de 0 a 16 (0, el listener no se está traceando, 16 máxima cantidad de traceo del listener) • off (equivalente a 0) no proporciona información de traceo • user (equivalente to 4) traza para identificar condiciones de error inducidas por el cliente. • admin (equivalente to 6) tracea para identificar problemas específicos de la instalación. • support (equivalente to 16) mayor nivel de traza para proporcionar información a Oracle Support Services.
TRACE_DIRECTORY_listener_name TRACE_FILE_listener_name	Directorio destino para el fichero de traza.
TRACE_FILELEN_listener_name	Especifica el tamaño del fichero de traza del listener en KB.
TRACE_FILENO_listener_name	Especifica el número de ficheros de trazas permitidos para el traceo. Cuando este parámetro se configure más allá del TRACE_FILELEN_listener_name , los ficheros de traza se van rotando.
TRACE_TIMESTAMP_listener_name	Añade un timestamp con el format dd-mon-yyyy hh:mi:ss:mil a todo fichero de traza.

Recolección de métricas de producto

En la siguiente tabla se listan las métricas a recoger para *Oracle Net Listener* junto con su periodicidad.

Métrica	Descripción	Origen Datos	Periodicidad
Conexiones establecidas por minuto	Número medio de conexiones por minuto que fueron establecidas por el listener	lsnrctl services <listener_name>	Cada hora
Conexiones rechazadas	Número de conexiones al listener que fueron rechazadas.	lsnrctl services <listener_name>	Cada hora
Tiempo de respuesta (msec)	Tiempo en milisegundos que tarda un listener en responder a un tns ping.	tnsping <listener_name>	Cada hora

Matriz Resultados Métricas de producto / Métricas de negocio

Tomando las métricas de negocio definidas en la sección correspondiente y las métricas específicas para *Oracle Net Listener*. Oracle Soporte recomienda la recogida de resultados en la siguiente matriz que cruza las métricas negocio con las métricas de *Oracle Net Listener*.

Prueba i	Resultados	#user	%concurrentia_usuarios	#iteraciones_casouso_i_transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrentia_casouso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Conexiones establecidas por minuto									
Conexiones rechazadas									
Tiempo de respuesta (msec)									

Oracle Database Single Instance

Configuración monitorización

Oracle proporciona el parámetro de inicialización *STATISTICS_LEVEL*, que controla la configuración de recolección de estadísticas. Este parámetro configura el nivel de recolección de las estadísticas de la base de datos.

Dependiendo de la configuración de *STATISTICS_LEVEL*, los asesores y estadísticas son recolectados como sigue:

- *BASIC*: No se recogerán estadísticas ni asesores. Monitorización y muchas características automáticas estarán deshabilitadas. Oracle no recomienda esta configuración puesto que deshabilita importantes características de Oracle.
- *TYPICAL*: Es la configuración por defecto y asegura la recolección de la mayoría de estadísticas mientras se proporciona el mejor rendimiento de la base de datos.
- *ALL*: Se ejecutarán todos los asesores y se recolectarán las mismas estadísticas que con la configuración *TYPICAL*, además de las estadísticas del sistema operativo y estadísticas de ejecución.

El parámetro *TIMED_STATISTICS* especifica si las estadísticas relativas al tiempo son recogidas.

- *True*, las estadísticas son recogidas y almacenadas en ficheros de trazas o mostradas en las vistas dinámicas *v\$sysstats* y *v\$sesstats*.
- *False*, el valor referente al tiempo de las estadísticas es cero. Esta configuración evita a Oracle el *overhead* de pedir al sistema operativo información de tiempo.

A partir de la versión 11.1.0.7, el valor *TIMED_STATISTICS* no puede configurarse a *false* si el valor de *STATISTICS_LEVEL* está configurado a *TYPICAL* o *ALL*.

Tras la configuración de los parámetros anteriores, se recomienda la configuración de la recogida de estadísticas de AWR cada 20 minutos.

(Nota: Caso de no disponer de Licencias de Diagnostic&Tuning Pack se configurará con la misma parametrización el paquete estadístico STATSPACK)

Recomendaciones configuración

STATISTIC_LEVEL	TYPICAL
TIMED_STATISTICS	True
TIMED_OS_STATISTICS	0
Intervalo recogida AWR a 20 minutos	exec dbms_workload_repository.modify_snapshot_settings (interval => 20)

Recolección de métricas de producto

Wait Events

Los eventos de espera son estadísticas que son incrementadas por un proceso servidor para indicar que se tuvo que esperar para que un evento se complete antes de poder continuar. Los eventos de espera revelan síntomas de problemas que pueden impactar el rendimiento, por ejemplo *latch contention*, *buffer contention*, y contención de I/O.

Los eventos de espera se agrupan en clases. Las clases de eventos de espera incluyen:

- Administrative,
- Application,
- Cluster,
- Commit,
- Concurrency,
- Configuration,
- Idle,
- Network,
- Other,
- Scheduler,
- System I/O
- User I/O

Las estadísticas de eventos de espera incluyen el número de veces que un evento ha estado esperando y el tiempo esperado para que un evento termine.

Como estadísticas de eventos de espera se recogerán los porcentajes de las clases de los eventos de espera.

Análisis de la carga

Normalmente, los eventos de espera son los primeros datos a analizar. Sin embargo, si se dispone de un informe de línea base se deberá revisar en primer lugar si la carga ha cambiado. A pesar de tener una línea base, es útil mirar si el uso de recursos es alto.

Las estadísticas relativas a la carga incluyen:

- redo size,
- session logical reads,
- db block changes,
- physical reads,
- physical read total bytes,
- physical writes,
- physical write total bytes,
- parse count (total),
- parse count (hard)
- user calls.

Por ello, además de los porcentajes de las clases de eventos de espera se recogerán las métricas de la base de datos relativas a la carga.

En el informe de *Automatic Workload Repository*, mirar la sección *Load Profile*, puesto que los datos han sido normalizados sobre transacciones y sobre segundos.

Métrica	Descripción	Origen Datos	Periodicidad
Clases de eventos de espera			
Administrative	Esperas resultantes de comandos DBA que causan que el usuario espere (por ejemplo, un index rebuild)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Application	Esperas resultantes del código de la aplicación de usuario (por ejemplo esperas por bloqueos causadas por bloqueos a nivel de fila o comandos explícitos de bloqueo)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Cluster	Esperas relativas a recursos de Real Application Cluster	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Commit	Esta clase de evento de espera solamente se compone de un evento de espera, la espera a la confirmación de escritura en un redo log después de un commit, esto es, log file sync	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Concurrency	Esperas por recursos internos de la base de datos (por ejemplo latches)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Configuration	Esperas causadas por una configuración inadecuada de los recursos de la base de datos o de la instancia (por ejemplo, tamaño de redo log no adecuado, tamaño de shared pool size)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Idle	Esperas que significan que la sesión está inactiva, esperando por trabajo (por ejemplo, SQL*Net message from client)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Network	Esperas relativas a mensajes de red	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Other	Esperas que de forma usual no ocurren en un sistema de base de datos	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Scheduler	Esperas relativas a Resource Manager	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
System I/O	Esperas por background process de entrada/salida (ejemplo db_file_parallel_write)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
User I/O	Esperas por E/S de usuario (ejemplo db_file_sequential_read)	Sección Wait Class informe AWR, solamente aparecerán las clases de eventos de espera sobre los que se ha esperado	20'
Métricas de carga			
redo size	Tamaño de redo generado desde la última toma de datos	Informe AWR -> Sección Load Profile -> Redo Size	20'
session logical reads	Lecturas lógicas de bloques de base de datos bien del buffer cache, bien del area privada del usuario.	Informe AWR -> Sección Load Profile -> Logical Reads	20'
db block changes	Esta estadística determina el ratio al cual los buffers están siendo marcados como sucios, cuenta el número total de cambios que generan entradas en los redo log y llegan a ser permanentes si la transacción es confirmada	Informe AWR -> Sección Load Profile -> Block change	20'
physical reads	Número total de bloques de datos leídos de disco	Informe AWR -> Sección Load Profile -> Physical Reads	20'
physical writes	Número total de bloques de datos escritos a disco	Informe AWR -> Sección Load Profile -> Physical Writes	20'
parse count (total)	Número total de parses (hard y soft) un soft parse es un chequeo de que un objeto existe en la shared pool, para verificar que los permisos del objeto no han cambiado.	Informe AWR -> Sección Load Profile -> Parses	20'

parse count (hard)	Número total de parses (hard parses), un hard parse es una operación muy costosa en términos de uso de memoria, puesto que requiere que Oracle reserve una estructura de memoria para construir un árbol de parseo.	Informe AWR -> Sección Load Profile -> Hard Parses	20'
Execution	Número total de llamadas (de usuario y recursivas) que ejecutan sentencias SQL	Informe AWR -> Sección Load Profile -> Executes	20'
Sorts	Número total de filas que han sido ordenadas	Informe AWR -> Sección Load Profile -> Sorts	20'
user calls	Número de llamadas de usuario, como por ejemplo login, parse, fetch o ejecución.	Informe AWR -> Sección Load Profile -> User Calls	20'

Matriz Resultados Métricas de producto / Métricas de negocio

Prueba i	Resultados	#user	%concurrency_usuarios	#iteraciones_caso uso_i transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrency_casouso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Clases de eventos de espera									
Administrative									
Application									
Cluster									
Commit									
Concurrency									
Configuration									
Idle									
Network									
Other									
Scheduler									
System I/O									
User I/O									

Load Profile									
redo size									
session logical reads									
db block changes									
physical reads									
physical writes									
parse count (total)									
parse count (hard)									
Execution									
Sorts									
user calls									

Oracle Real Application Clusters

Configuración monitorización

La configuración para la monitorización de métricas para *Real Application Cluster* (RAC) es similar a la configuración de monitorización para *Oracle Database*, por lo tanto aplicarán las mismas recomendaciones de configuración de base de datos a cada una de las instancias que forman el RAC. Las estadísticas se recogerán en todas las instancias que formen parte del RAC.

Recolección de métricas de producto

Para cada una de las instancias que formen el RAC, se recogerán las mismas métricas que se listan en el presente informe para la base de datos *single instance*. En dichas recomendaciones se recogen las clases de eventos de espera y las métricas de carga para la instancia de base de datos.

Para el caso particular de base de datos en RAC, además de las métricas para base de datos se recogerán las siguientes métricas de carga aplicables a RAC.

Métrica	Descripción	Origen Datos	Periodicidad
Global Cache blocks received	Número de bloques recibidos desde la instancia remota por el Interconnect	AWR. Sección Global Cache Load Profile	Cada 20'
Global Cache blocks served	Número de bloques enviados a la instancia remota por el Interconnect	AWR. Sección Global Cache Load Profile	Cada 20'
GCS/GES messages received	Número de mensajes recibidos desde la instancia remota por el Interconnect	AWR. Sección Global Cache Load Profile	Cada 20'
GCS/GES messages sent	Número de mensajes enviados a la instancia remota por el Interconnect	AWR. Sección Global Cache Load Profile	Cada 20'
Estd Interconnect traffic (KB)	Tráfico estimado del Interconnect en KB	AWR. Sección Global Cache Load Profile	Cada 20'

Notas: Las estadísticas se recogerán en todas las instancias que formen parte del RAC.

Matriz Resultados Métricas de producto / Métricas de negocio

Para RAC aplica la misma matriz de recogida de datos que para base de datos, recogiendo los datos para cada instancia de base de datos. Además, se añadirá a dicha matriz la recogida de las métricas de RAC indicadas en el epígrafe de métricas de producto.

Prueba i	Resultados	#user	%concurrentia_usuarios	#iteraciones_caso_uso_i_transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrentia_casouso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Global Cache blocks received									
Global Cache blocks served									
GCS/GES messages received									
GCS/GES messages sent									
Estad Interconnect traffic (KB)									

Oracle Streams

Configuración monitorización

Tras la configuración de un entorno de *Streams* sobre una base de datos Oracle, las métricas de rendimiento se irán acumulando en las vistas dinámicas que almacenan los metadatos de *Streams*, por ello para la configuración aplican la misma parametrización de recogida de una instancia de base de datos Oracle.

Recolección de métricas de producto

Métrica	Descripción	Origen Datos	Periodi cidad
Latencia de escaneo de redo para cada proceso de captura	Especifica el numero de segundos entre la fecha de creación de la entrada de redolog mas reciente escaneado por un proceso de captura y el tiempo actual.	<pre> COLUMN CAPTURE_NAME HEADING 'Capture Process Name' FORMAT A25 COLUMN LATENCY_SECONDS HEADING 'Latency in Seconds' FORMAT 9999999999 COLUMN LAST_STATUS HEADING 'Seconds Since Last Status' FORMAT 999999 COLUMN CAPTURE_TIME HEADING 'Current Process Time' COLUMN CREATE_TIME HEADING 'Message Creation Time' FORMAT 999999 SELECT CAPTURE_NAME, ((SYSDATE - CAPTURE_MESSAGE_CREATE_TIME)*86400) LATENCY_SECONDS, ((SYSDATE - CAPTURE_TIME)*86400) LAST_STATUS, TO_CHAR(CAPTURE_TIME, 'HH24:MI:SS MM/DD/YY') CAPTURE_TIME, TO_CHAR(CAPTURE_MESSAGE_CREATE_TIME, 'HH24:MI:SS MM/DD/YY') CREATE_TIME FROM V\$STREAMS_CAPTURE; </pre>	Cada minuto
Latencia del job de propagación	Es el tiempo de espera máximo para propagación de un nuevo mensaje durante la propagación	<pre> COLUMN START_DATE HEADING 'Start Date' COLUMN PROPAGATION_WINDOW HEADING 'Duration in Seconds' FORMAT 99999 COLUMN NEXT_TIME HEADING 'Next Time' FORMAT A8 COLUMN LATENCY HEADING 'Latency in Seconds' FORMAT 99999 COLUMN SCHEDULE_DISABLED HEADING 'Status' FORMAT A8 COLUMN PROCESS_NAME HEADING 'Process' FORMAT A8 COLUMN FAILURES HEADING 'Number of Failures' FORMAT 99 SELECT DISTINCT TO_CHAR(s.START_DATE, 'HH24:MI:SS MM/DD/YY') START_DATE, s.PROPAGATION_WINDOW, s.NEXT_TIME, s.LATENCY, DECODE(s.SCHEDULE_DISABLED, 'Y', 'Disabled', 'N', 'Enabled') SCHEDULE_DISABLED, s.PROCESS_NAME, s.FAILURES FROM DBA_QUEUE_SCHEDULES s, DBA_PROPAGATION p WHERE p.PROPAGATION_NAME = '<NAME_PROPAGATION>' AND p.DESTINATION_DBLINK = s.DESTINATION AND s.SCHEMA = p.SOURCE_QUEUE_OWNER AND s.QNAME = p.SOURCE_QUEUE_NAME; </pre>	Cada minuto
Latencia de desencolación de un mensaje	Es la cantidad de tiempo entre cuando el mensaje fue creado en la base de datos fuente y cuando el mensaje fue desencolado por el proceso de apply	<pre> COLUMN APPLY_NAME HEADING 'Apply Process Name' FORMAT A17 COLUMN LATENCY HEADING 'Latency in Seconds' FORMAT 9999 COLUMN CREATION HEADING 'Message Creation' FORMAT A17 COLUMN LAST_DEQUEUE HEADING 'Last Dequeue Time' FORMAT A20 COLUMN DEQUEUED_MESSAGE_NUMBER HEADING 'Dequeued Message Number' FORMAT 999999 SELECT APPLY_NAME, (DEQUEUE_TIME-DEQUEUED_MESSAGE_CREATE_TIME)*86400 LATENCY, TO_CHAR(DEQUEUED_MESSAGE_CREATE_TIME, 'HH24:MI:SS MM/DD/YY') CREATION, TO_CHAR(DEQUEUE_TIME, 'HH24:MI:SS MM/DD/YY') LAST_DEQUEUE, DEQUEUED_MESSAGE_NUMBER FROM V\$STREAMS_APPLY_READER; </pre>	Cada minuto

Matriz Resultados Métricas de producto / Métricas de negocio

Prueba i	Resultados	#user	%concurrentia_usuarios	#iteraciones_caso uso_i_transaccional	%ejecucion_casouso_i	#registros_p rocesados_c asouso_i_m asivo	%concurrentia_caso uso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Latencia de escaneo de redo para cada proceso de captura									
Latencia del job de propagación									
Latencia de descolación de un mensaje									

Oracle Dataguard

Configuración monitorización

Configurar *TIMED_STATISTICS=TRUE* tanto en la base de datos primaria como en la base de datos *standby*.

Además se recomienda la planificación de la recogida de los siguientes datos listados a continuación:

- Salida del comando *sar* del sistema operativo, el cual debe invocarse cada 60 segundos en la instancia primaria y en la *standby*.
- Salida del comando del OS *vmstat* durante el periodo de recuperación de la *standby*.
- Informes *AWR* de la base de datos primaria
- Salida de la siguiente *query* cada 60 segundos en la base de datos *standby*:

```
select name, value, to_char(sysdate, 'DD-MON-YYYY  
HH:MI:SS') from v$sysstat where value > 0 order by name;  
spool off
```

- Salida de la siguiente *query* cada 60 segundos en la *standby* database.

```
set echo off  
set feedback off  
set termout off  
set pagesize 100  
set numwidth 10  
column event format a40 truncate  
column name format a35 truncate  
column opname format a35 truncate  
column value format 9999999999999999  
select event, total_waits, time_waited, average_wait*10  
from v$system_event where time_waited > 100 and event not  
like 'rdbms ipc %' and event not like '%timer%' and  
lower(event) not like '%idle%' and lower(event) not like  
'sql%net%' and event not like 'ges%' order by time_waited;
```

- Salida de la siguiente consulta de la vista *V\$RECOVERY_PROGRESS* una vez que termine la operación de *media recovery* en la base de datos *standby*:

```
set echo off  
set feedback off  
set termout off  
set pagesize 100  
set numwidth 15  
  
column type format a20 truncate  
column item format a25 truncate  
column units format a10 truncate  
column name format a30 truncate  
select to_char(start_time, 'dd-mon-yyyy HH:MI:SS')  
start_time, type, item, units, sofar, total,  
to_char(timestamp, 'dd-mon-yyyy HH:MI:SS') timestamp from  
v$recovery_progress;
```

Recolección de métricas de producto

En la siguiente tabla se listan las métricas a recoger para *Dataguard* junto con su periodicidad. Se incluye de forma específica si la recolección debe ser en la primaria o la *standby*.

Métrica	Descripción	Origen Datos	Periodicidad
Conexiones establecidas por minuto	Número medio de conexiones por minuto que fueron establecidas por el listener	lsnrctl services <listener_name>	Cada 24 horas
Sincronización entre la primaria y la standby	Tiempo de retardo del sincronismo de la <i>standby</i> con la primaria del entorno Dataguard.	v\$dataguard_stats v\$archive_dest. select value,unit, time_computed from v\$dataguard_stats where name='apply lag';	Cada hora
Average Apply Rate	Redo Applied / Elapsed	Ver notas Punto 4	Cada hora
Active Apply Rate	Redo Applied / Active	Ver notas Punto 4	Cada hora

Notas: Revisión en entornos Dataguard.

1. *Broker status* (si aplicará por que estuviera configurado)
2. Se revisan los *listener*
3. Sincronización
 - Para poder chequear el estado entre la primaria y la *standby* en entornos *dataguards* se lanzará la siguiente consulta periódicamente para saber el tiempo exacto de retardo de sincronismo que lleva la *standby* con respecto a la primaria.

```
select value,unit, time_computed from
v$dataguard_stats where name='apply lag';
```

También se pueden revisar los archivos recibidos en la *standby* usando la vistas *v\$archive_dest*.

4. Rendimiento sincronización
 - Para obtener un histórico de los valores “*Average Apply Rate*” y “*Active Apply Rate*” y saber si existe una pérdida de rendimiento en la aplicación de cambios en la *standby*.

```
column type format a20 truncate
column item format a25 truncate
column units format a10 truncate
column name format a30 truncate

select to_char(start_time,'dd-mon-yyyy HH:MI:SS')
start_time, type, item, units, sofar, total,
to_char(timestamp,'dd-mon-yyyy HH:MI:SS') timestamp
```

```
from v$recovery_progress;
```

5. Eventos de espera

Se pueden obtener los eventos de sistemas y de sesión consultando las vistas *V\$SYSTEM_EVENTS* y *V\$SESSION_WAITS* y extraer los más representativos teniendo en cuenta el valor de *"TIME_WAITED"*.

```
select event, total_waits,
round(time_waited/100) "TIME(s)",
average_wait*10 "AVG(ms)",
TO_CHAR(SYSDATE, 'DD-MON-YYYY HH:MI:SS') time
from v$system_event where time_waited > 100 and
event not in ('rdbms ipc message','pmon timer','control
file heartbeat','smon timer')
order by time_waited;
```

Matriz Resultados Métricas de producto / Métricas de negocio

Prueba i	Resultados	#user	%concurrency_ usuarios	#iteraciones_casouso_i_transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrency_a_casouso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Conexiones establecidas por minuto									
Sincronización entre la primaria y la standby									
Average Apply Rate									
Active Apply Rate									

Oracle JVM JRockit y Weblogic Server

La capa del servidor de aplicaciones comprende tanto el entorno de los servidores *Weblogic* en los que están desplegadas las aplicaciones, así como la máquina virtual *Java JRockit* de cada uno de los servidores. Puesto que ambos componentes tienen el mismo método de acceso a las métricas en tiempo de ejecución, la configuración y recogida de métricas para *Weblogic Server* y *JRockit* serán mostradas en el mismo epígrafe.

Weblogic Server proporciona su propio conjunto de *MBeans* que puede usarse para configurar, monitorizar y gestionar recursos *Weblogic Servers*.

Existen diferentes dos tipos de *MBean*, para monitorizar y configurar. Todos los *MBeans* de *Weblogic Server* pueden ser organizados dentro de los siguientes tipos dependiendo si el *MBean* monitoriza o configura un recurso:

- *Runtime MBeans*, contienen información sobre el estado del servidor en tiempo de ejecución y sus recursos. Generalmente contienen solamente datos sobre el estado actual del servidor o los recursos y no tienen persistencia de datos. Cuando se cierra la instancia del servidor, todas las estadísticas y métricas del *MBean* son destruidas.
- *Configuration MBeans*, contienen información sobre la configuración del servidor y sus recursos. Representan la información que se almacena en los ficheros de configuración XML del dominio.

Oracle Soporte utilizará los *MBean* para la monitorización, el servidor de *MBean* es el *core* de cualquier agente *JMX*, y actúa como contenedor de *MBeans*. Por ejemplo la *JVM* del *AdminServer* mantiene tres *MBeans* *server* y opcionalmente mantienen el servidor *MBean* de la plataforma (*platform MBean*), que es proporcionado por el *JDK*. La *JVM* de un *ManagedServer* mantendrá solamente un *MBean Server* y el opcional *platform MBean*.

La siguiente tabla describe cada uno de *MBean Server* que se encuentran en los servidores *Weblogic*.

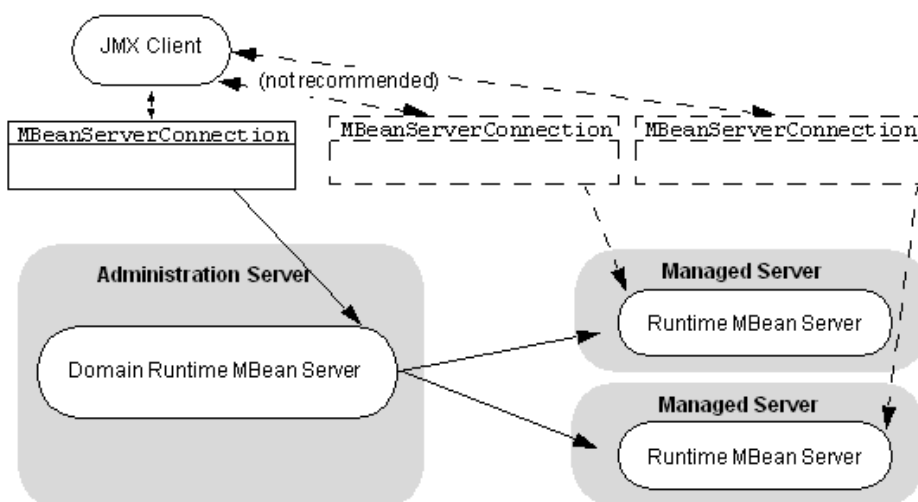
Tipo *MBean* Server

Crea, registra y proporciona acceso a...

Domain Runtime MBean Server	<i>MBeans</i> para el ámbito del dominio. Este <i>MBean</i> también actúa como punto de acceso para los <i>MBeans</i> que residen en los <i>Managed Sever</i> . Puede registrarse un <i>MBean</i> propio en este <i>MBean</i> .
Runtime MBean Server	<i>MBeans</i> que expone monitorización, control de ejecución y la configuración activa de una instancia específica de <i>Weblogic Server</i> . Los <i>MBeans</i> que sean desarrollados a medida pueden registrarse también en este <i>MBean server</i> .
Edit MBean Server	Es el <i>MBean</i> que mantiene las configuraciones pendientes y las operaciones que controlan la configuración de un dominio de <i>Weblogic</i> . Publica un <i>ConfigurationManagerMBean</i> para bloquear, salvar y activar cambios. Solo el <i>AdminServer</i> tiene una instancia de este <i>MBean</i> .

JVM platform	<i>MBeans</i> proporcionado por el JDK que contiene información de
MBean server	monitorización de la máquina virtual Java.

Definidos los tipos de *MBean* Server, se usará el *platform MBean* para recoger las métricas de monitorización de la máquina virtual *Java JRockit*, para la recogida de las métricas del servidor *Weblogic* se accederá al *Runtime MBean* de cada uno de los servidores del dominio, a través del *Domain Runtime MBean*, puesto que es el método recomendado.



Configuración monitorización

- Configurar el *Weblogic Server Runtime MBean* para ser el *Platform MBean*.
 - Activar el *Platform MBean* Server para que sea accedido a través del Run-time *MBean*. A través de la consola de administración de *Weblogic*, navegar a Dominio > Configuración > General y en opciones avanzadas seleccionar el *check box Platform MBean* Server.
- El acceso remoto al *Platform MBean* no está habilitado por defecto.
 - Para configurar la instancia de *JVM*, tiene que especificarse la información de conexión para decirle al agente como obtener la información de gestión de la instancia *JVM* en ejecución. Por ello es necesario conocer el puerto al cual se conecta el agente para consultar la información del *Platform MBean*, el puerto se encuentra definido en el fichero *config.xml* del dominio de *Weblogic*, también puede indicarse el puerto de conexión para el agente *JMX* durante el arranque de *JRockit*.

```
-Djrockit.managementserver.port=<port-number>
-Xmanagement:ssl=false,authenticate=false
```

- Los clientes *JMX* remotos acceden al *platform MBeans* conectándose al *Runtime MBean*.

Recolección de métricas de producto

Usaremos *WLST* para el acceso al *Platform MBean* y al *Domain MBean* y recoger los datos tanto para la *JVM* como el servidor *Weblogic*.

Para la recogida de las propiedades para *JVM JRockit* usaremos el siguiente *WLST*, sustituyendo los métodos *java*, por los listados en la tabla de métricas.

Consulta métrica AdminServer

```
connect()
domainCustom()
cd ("java.lang")

# Admin Server, consulta de uso de heap y thread

cd ("java.lang:Location=AdminServer,type=Memory")
huAdmin = get("HeapMemoryUsage")
cd ("..")

cd ("java.lang:Location=AdminServer,type=Threading")
numThreadsAdmin = get("ThreadCount")

print "memory usage=", huAdmin.get("max"), "number threads:", numThreadsAdmin
cd ("..")
```

Consulta métrica ManagedServer

```
connect()
domainCustom()
cd ("java.lang")

# El ManagedServer se denomina m1, consulta uso heap y thread

cd ("java.lang:Location=m1,type=Memory")
huM1 = get("HeapMemoryUsage")
cd ("..")

cd ("java.lang:Location=m1,type=Threading")
numThreadM1 = get("ThreadCount")
cd ("..")

print "memory usage = ", huM1.get("max"), " number threads: ", numThreadM1
```

Para la recogida de datos de *Weblogic Server*, usaremos *WLST*, accediendo en primer lugar al *MBean* que contiene los atributos y posteriormente a la propiedad

Ejemplo de recogida de métricas para el JDBC

```
connect('Weblogic','Weblogic','t3://localhost:7001')
allServers=domainRuntimeService.getServerRuntimes();
if (len(allServers) > 0):
    for tempServer in allServers:
        jdbcServiceRT = tempServer.getJDBCServiceRuntime();
        dataSources = jdbcServiceRT.getJDBCDataSourceRuntimeMBeans();
        if (len(dataSources) > 0):
            for dataSource in dataSources:
                print 'ActiveConnectionsAverageCount' ,
                dataSource.getActiveConnectionsAverageCount()
                print 'ActiveConnectionsCurrentCount' ,
                dataSource.getActiveConnectionsCurrentCount()
                print 'ActiveConnectionsHighCount' ,
                dataSource.getActiveConnectionsHighCount()
                print 'ConnectionDelayTime' ,
                dataSource.getConnectionDelayTime()
                print 'ConnectionsTotalCount' ,
                dataSource.getConnectionsTotalCount()
                print 'CurrCapacity' ,
                dataSource.getCurrCapacity()
                print 'CurrCapacityHighCount' ,
                dataSource.getCurrCapacityHighCount()
                print 'DeploymentState' ,
                dataSource.getDeploymentState()
                print 'FailedReserveRequestCount' ,
                dataSource.getFailedReserveRequestCount()
```

```

        print 'FailuresToReconnectCount'      ' ' ,
dataSource.getFailuresToReconnectCount()
        print 'HighestNumAvailable'           ' ' ,
dataSource.getHighestNumAvailable()
        print 'HighestNumUnavailable'          ' ' ,
dataSource.getHighestNumUnavailable()
        print 'LeakedConnectionCount'          ' ' ,
dataSource.getLeakedConnectionCount()
        print 'ModuleId'                       ' ' ,
dataSource.getModuleId()
        print 'Name'                           ' ' ,
dataSource.getName()
        print 'NumAvailable'                   ' ' ,
dataSource.getNumAvailable()
        print 'NumUnavailable'                 ' ' ,
dataSource.getNumUnavailable()
        print 'Parent'                        ' ' ,
dataSource.getParent()
        print 'PrepStmtCacheAccessCount'       ' ' ,
dataSource.getPrepStmtCacheAccessCount()
        print 'PrepStmtCacheAddCount'          ' ' ,
dataSource.getPrepStmtCacheAddCount()
        print 'PrepStmtCacheCurrentSize'       ' ' ,
dataSource.getPrepStmtCacheCurrentSize()
        print 'PrepStmtCacheDeleteCount'       ' ' ,
dataSource.getPrepStmtCacheDeleteCount()
        print 'PrepStmtCacheHitCount'          ' ' ,
dataSource.getPrepStmtCacheHitCount()
        print 'PrepStmtCacheMissCount'         ' ' ,
dataSource.getPrepStmtCacheMissCount()
        print 'Properties'                     ' ' ,
dataSource.getProperties()
        print 'ReserveRequestCount'            ' ' ,
dataSource.getReserveRequestCount()
        print 'State'                         ' ' ,
dataSource.getState()
        print 'Type'                          ' ' ,
dataSource.getType()
        print 'VersionJDBCdriver'              ' ' ,
dataSource.getVersionJDBCdriver()
        print 'WaitingForConnectionCurrentCount' ,
dataSource.getWaitingForConnectionCurrentCount()
        print 'WaitingForConnectionFailureTotal' ' ' ,
dataSource.getWaitingForConnectionFailureTotal()
        print 'WaitingForConnectionHighCount'   ' ' ,
dataSource.getWaitingForConnectionHighCount()
        print 'WaitingForConnectionSuccessTotal' ' ' ,
dataSource.getWaitingForConnectionSuccessTotal()
        print 'WaitingForConnectionTotal'       ' ' ,
dataSource.getWaitingForConnectionTotal()
        print 'WaitSecondsHighCount'           ' ' ,
dataSource.getWaitSecondsHighCount()

```

En la siguiente tabla se listan las métricas a recoger para *JVM JRockit* y *Oracle Weblogic Server* junto con su periodicidad.

Métrica	Descripción	Origen Datos (MBean)	Periodicidad
Métricas Oracle JRockit			
Classes Loaded (since startup)	Número total de clases que han sido cargadas dentro de la JVM desde el arranque .	java.lang:type=ClassLoading - getTotalLoadedClassCount()	20'
Classes Unloaded (since startup)	Número total de clases que han sido descargadas de la JVM desde el arranque.	java.lang:type=ClassLoading - getUnloadedClassCount()	20'
Compilation Time (since startup) (ms)	Tiempo total empleado por la JVM en compilación.	java.lang:type=Compilation - getTotalCompilationTime()	20'
Garbage Collection Invocations (since startup)	Número de veces que se ha invocado al Garbage collector desde el arranque de JVM	java.lang:type=GarbageCollector,name=collector's name - getCollectionCount()	20'
Garbage Collector - Execution Time (since startup) (ms)	Tiempo total empleado por el Gargage collector desde el inicio de la JVM	java.lang:type=GarbageCollector,name=collector's name - getCollectionTime()	20'
Memory Pool - Current Usage (KB)	Memoria actual usada por la JVM	java.lang:type=MemoryPool,name=pool's name - getUsage()	20'
Memory Pool - Peak Usage (KB)	Máxima memoria usada por la JVM desde el inicio de la JVM	java.lang:type=MemoryPool,name=pool's name - getPeakUsage()	20'
Heap Memory Usage (KB)	Heap Memory actualmente en uso	java.lang:type=Memory - getHeapMemoryUsage()	20'
Non-Heap Memory Usage (KB)	NonHeap Memory actualmente en uso	java.lang:type=Memory - getNonHeapMemoryUsage()	20'
Objects Pending Finalization	Número de objetos pendientes de finalización en la JVM	java.lang:type=Memory - getObjectPendingFinalizationCount()	20'
Deadlocked Threads	Número de ciclos de thread en deadlocked	java.lang:type=Threading - findMonitorDeadlockedThreads()	20'
Peak Threads	Máximo número de threads iniciados en la JVM desde el inicio	java.lang:type=Threading - getPeakThreadCount()	20'
Active Threads	Número actual de Active Threads	java.lang:type=Threading - getThreadCount()	20'
Threads Started (since startup)	Threads iniciados desde el inicio de la JVM	java.lang:type=Threading - getTotalStartedThreadCount()	20'
Métricas Oracle Weblogic Server			
Server Datasource - ActiveConnectionsCurrentCount	Número de conexiones activas en un JDBC connection pool	getJDBCServiceRuntime(); getJDBCDataSourceRuntimeMBeans(); getActiveConnectionsCurrentCount()	20'
Server Datasource - ActiveConnectionsHighCount	El mayor número de conexiones activas en un JDBC connection pool	getJDBCServiceRuntime(); getJDBCDataSourceRuntimeMBeans(); getActiveConnectionsHighCount()	20'
Server Datasource - NumAvailable	Número actual de conexiones a base de datos disponibles	getJDBCServiceRuntime(); getJDBCDataSourceRuntimeMBeans(); getNumAvailable()	20'
Server Datasource - WaitingForConnectionCurrentCount	Número de peticiones que están esperando por una conexión del JDBC connection pool	getJDBCServiceRuntime(); getJDBCDataSourceRuntimeMBeans(); getWaitingForConnectionCurrentCount()	20'
OpenSocketsCurrentCount	Número medio de llamadas a servlet o JSP en los últimos 5 minutos	java Weblogic.Admin -username user -password pass GET -pretty -type ServerRuntime grep OpenSocketsCurrentCount	20'
AcceptBacklog	Número de peticiones que están esperando por una conexión TCP	java Weblogic.Admin -username user -password pass GET -pretty -type Server grep AcceptBacklog	20'
ExecuteThreadCurrentIdleCount	Número de thread en la cola de ejecución del servidor que están en estado idle y no están procesando nada.	java Weblogic.Admin -username user -password pass GET -pretty -type ExecuteQueueRuntime grep ExecuteThreadCurrentIdleCount	20'

PendingRequestCurrentCount	Número de peticiones pendientes que están esperando en la cola de ejecución.	java <i>Weblogic</i> .Admin -username user -password pass GET -pretty -type <i>ExecuteQueueRuntime</i> grep <i>PendingRequestCurrentCount</i>	20'
TransactionCommittedTotalCount	Número total de transacciones que han sido procesadas por el <i>Weblogic</i> Server	java <i>Weblogic</i> .Admin -username user -password pass GET -pretty -type <i>TransactionResourceRuntime</i> grep <i>TransactionCommittedTotalCount</i>	20'
TransactionRolledBackTotalCount	Número total de transacciones sobre las que se ha hecho rollback	java <i>Weblogic</i> .Admin -username user -password pass GET -pretty -type <i>TransactionResourceRuntime</i> grep <i>TransactionRolledBackTotalCount</i>	20'
InvocationTotalCount	Número total de veces que un servlet en ejecución en el servidor <i>Weblogic</i> fue invocado.	java <i>Weblogic</i> .Admin -username user -password pass GET -pretty -type <i>ServletRuntime</i> grep <i>InvocationTotalCount</i>	20'

Matriz Resultados Métricas de producto / Métricas de negocio

Tomando las métricas de negocio definidas en la sección correspondiente y las métricas específicas para *Oracle JRockit* y *Oracle Weblogic Server*. Oracle Soporte recomienda que los resultados sean tomados en la siguiente matriz que cruza las métricas negocio con las métricas de *Oracle JRockit* y *Oracle Weblogic Server*.

Prueba i	Resultados	#user	%concurrency_ usuarios	#iteraciones_caso uso_i_transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrency_casouso	tiempo_in	tiempo_out
Valores de Entrada		S	T	U	V	W	X	Y	Z
Métricas <i>Oracle JRockit</i>									
Classes Loaded (since startup)									
Classes Unloaded (since startup)									
Compilation Time (since startup) (ms)									
Garbage Collection Invocations (since startup)									
Garbage Collector - Execution Time (since startup) (ms)									
Memory Pool - Current Usage (KB)									

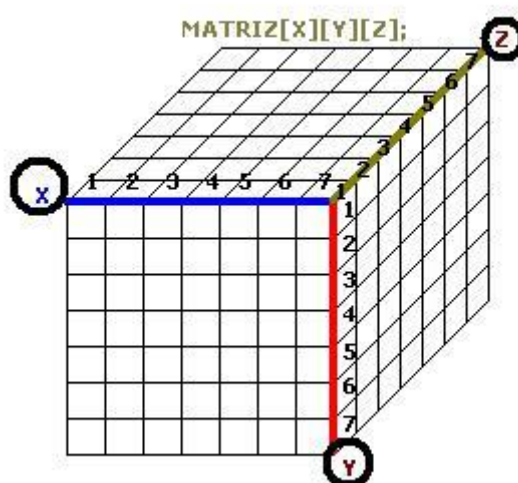
Memory Pool - Peak Usage (KB)									
Heap Memory Usage (KB)									
Non-Heap Memory Usage (KB)									
Objects Pending Finalization									
Deadlocked Threads									
Peak Threads									
Active Threads									
Threads Started (since startup)									
Métricas Oracle Weblogic Server									
Server Datasource - ActiveConnectionsCurrentCount									
Server Datasource - ActiveConnectionsHighCount									
Server Datasource - NumAvailable									
Server Datasource - WaitingForConnectionCurrentCount									
OpenSocketsCurrentCount									
AcceptBacklog									
ExecuteThreadCurrentIdleCount									
PendingRequestCurrentCount									
TransactionCommittedTotalCount									
TransactionRolledBackTotalCount									



InvocationTotalCount									
----------------------	--	--	--	--	--	--	--	--	--

Tratamiento estadístico de resultados

A lo largo del presente documento se han definido las matrices de valores que deben recogerse para cada producto *Oracle*. Es por ello, que la ejecución de unas pruebas de carga sobre una aplicación en un intervalo de tiempo determinado, resultará en un número N de matrices de valores para cada producto, siendo N el número de muestras recogidas para cada métrica a lo largo de la duración de las pruebas de carga.



X – variables de negocio.

Y – variables de referencia producto *Oracle*.

Z – número de muestras recogidas durante la duración de las pruebas.

Puesto que los valores resultantes estarán distribuidos en el tiempo, se necesitarán de herramientas estadísticas para conocer mejor la dispersión de los valores, y la frecuencia de ellos. Dichas herramientas estadísticas nos proporcionarán que valores no se encuentran dentro de la normalidad de la muestra.

Se recomienda el cálculo de las siguientes variables estadísticas

- **Media aritmética**, cantidad total de la variable distribuida a partes iguales entre cada observación.
- **Varianza**, medida de dispersión de los valores.
- **Moda**, será el valor con una mayor frecuencia en la distribución de datos.
- **Desviación estándar**, medida de dispersión de valores, conoceremos la desviación que presentan los datos respecto de la media aritmética de dicha distribución.

Las variables estadísticas serán calculadas para cada métrica de producto *Oracle* y para cada métrica de negocio, siendo una posible hoja de resultados igual a la siguiente:

Prueba i	#user	%concurrency_usuarios	#iteraciones_casouso_i_transaccional	%ejecucion_casouso_i	#registros_procesados_casouso_i_masivo	%concurrency_a_casouso	tiempo_in	tiempo_out
Valores de Entrada	S	T	U	V	W	X	Y	Z
Metrica Producto 1		Valor de Media Valor de Varianza Valor de Moda Valor de Desviación estándar						
...								
Metrica Producto N								

A partir de la hoja de resultados, se observarán las métricas de producto y las desviaciones que pudieran existir en comparación con los valores normales.

Para aquellos valores fuera de la normalidad, pasarán a analizarse los valores de entrada de las métricas de negocio y todas las matrices de valores de métricas de producto *Oracle* a lo largo del tiempo, para inferir cuales son las causas de un posible mal resultado de las pruebas de ejecución.