

## Error: Database connection failed.

Prof. José [Rama - jose@fq.edu.uy](mailto:jose@fq.edu.uy)

UAIQ

Facultad de Química

Resumen

**Error: Database connection failed.**, es una de las frases mas temidas de los administradores de Moodle, el cual como buen LCMS depende de una base de datos para funcionar.

En particular el Moodle es un LCMS que hace un uso muy intensivo del motor de bases de datos, y no todas sus Querys son optimizadas, una consulta sencilla puede generar cientos consultas a nivel de la base de datos.

Esta alta demanda de consultas a la base de datos, sumado al tipo de accesos ( generalmente en ráfagas de estudiantes por curso y en franjas de horarios bastantes definidas en algunos casos )

hace que los atascos de la base de datos sean frecuentes al aumentar el numero de estudiantes/cursos por encima de los recursos del servidor.

Los requerimientos de cpu/memoria crecen rápidamente al crecer el numero de estudiantes concurrentes en un instante dado.

Estos requerimientos se dan principalmente por dos lados a) el servidor web y b) el servidor de bases de datos.

Según el tipo de acciones en el moodle los problemas de saturación se pueden dar por a o por b.

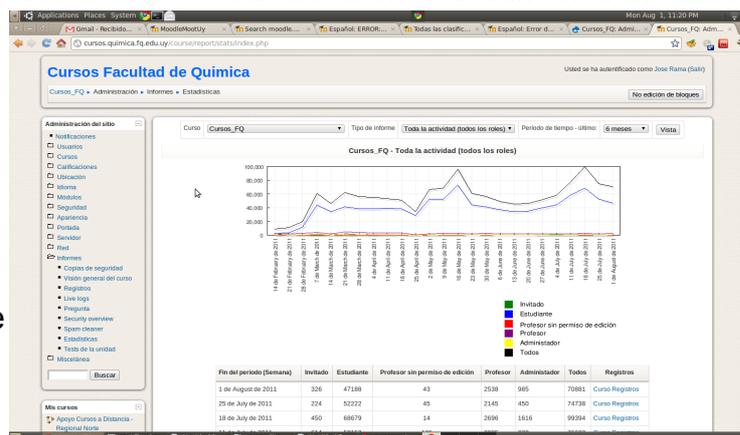
Mientras la velocidad de atención del servidor web(SW) sea mas rápida que el servidor de base de datos (SBD), se controla la fluencia de datos fácilmente por el SW y en menor grado del Sistema de Archivos.

Cuando la velocidad del SBD crece por motivos de configuración o por falla del motor de gestión de base de datos, el sistema crece muy rápidamente hacia parámetros de funcionamiento no aceptables, volviéndose inestable.

Una primera cosa importante es poder prever ese momento ( en lo posible) y en caso de llegar, ver cuales son nuestras posibilidades ( o cuando cerrar la puerta a tiempo ).

Hay varias estructuras posibles para incrementar el rendimiento de un motor de bases de datos, pero es bueno saber cuando un buen ajuste de parámetros nos permite sacar mayor rendimiento a los recursos disponibles, además que nos permiten prever posibles riesgos.

En este trabajo utilizamos dos herramientas que nos permiten estudiar las estadísticas del motor de bases de datos MYSQL, sacar conclusiones sobre el uso de estos parámetros y que cambiar en la configuración para optimizar de acuerdo a la carga histórica del mismo.



Estas herramientas son indicadoras, y se utilizan algunas reglas empíricas, otras de sentido común ( pero a veces olvidamos), pero no hay que olvidar que al final, siempre tendremos un límite de usuarios concurrentes para una determinada arquitectura de servidores.

## Problemas Detectados

Si no llevamos estadísticas sobre el uso del motor de base de datos, memoria y tiempos de accesos de páginas, igualmente nos llegarán quejas de usuarios que nos indiquen tiempos de acceso a páginas de cursos o eventuales cortes a la base de datos, el problema es que sino estamos llevando estadísticas de accesos o no tenemos el servidor controlado, no podremos extraer relaciones causa/efecto de estos problemas, por eso siempre es recomendable en un servidor Moodle, mantener algunas medidas por fuera del sistema moodle, como puede ser carga del servidor apache, número de procesos mysql, ocupación del servidor, etc, en función del tiempo, sino, estamos ciegos. Una de las características de un servidor de bases de datos no optimizado, o con parámetros por defecto ante grandes cargas de usuarios, crece el tiempo de respuesta, aumentan los procesos del servidor de web en espera y vemos que la/s CPUs del sistema no tienen una ocupación alta.

Esto se puede dar por varios motivos, uno de ellos es la demora del Gestor de base de datos en responder, tanto porque la memoria asignada a caches no sean suficiente, no existan suficientes asignaciones a archivos abiertos en el sistema de archivos, o no halla la suficiente cantidad de conexiones a la base de datos.

## Algunos Conceptos.

Usuarios concurrentes: son los usuarios que acceden dentro de la ventana de tiempo, tal que son concurrentes para el gestor de base de datos.

Acceso: el ingreso de un usuario y generación de una sesión en PHP.

Registro: toda acción que queda registrada en el log del Moodle

## Revelamientos de datos

Además de las estadísticas del Moodle, es útil mantener los siguientes datos en forma de log:

Memoria usada por el servidor.

Cantidad de procesos Apache y Mysql.

Procesos Radius ( por lo menos en nuestro caso)

También es útil crear algún script que verifique la carga y avise por correo electrónico en casos de posible saturación

## Algunos Útiles

Uno de los útiles es :

MySQL performance tuning primer script  
Written by: Matthew Montgomery <mmontgomery@mysql.com>  
Inspired by: MySQLARd (<http://gert.sos.be/demo/mysqlar/>)

Este script nos permite revisar las estadísticas del Mysql, y nos ofrece algunas sugerencias, las cuales deben ser analizadas en base a los siguientes criterios:

a.- Periodo de carga, no podemos realizar inferencias sobre periodos de carga pequeños o que no sea representativos ( baja demanda de paginas).

b.- Analizar los resultados en función de si tenemos solo funcionando Moodle o varios CMS.

Ejemplo de salida y su análisis (se han eliminado algunos test):

**-- MYSQL PERFORMANCE TUNING PRIMER --**

**- By: Matthew Montgomery -**

**MySQL Version 5.0.45-Debianlog x86\_64**

**Uptime = 0 days 2 hrs 39 min 2 sec**

**Avg. qps = 106**

**Total Questions = 1013634**

**Threads Connected = 106**

**BINARY UPDATE LOG**

**The binary update log is enabled**

**Binlog sync is not enabled, you could loose binlog records during a server crash**

**WORKER THREADS**

**Current thread\_cache\_size = 16**

**Current threads\_cached = 6**

**Current threads\_per\_sec = 0**

**Historic threads\_per\_sec = 0**

**Your thread\_cache\_size is fine**

**MAX CONNECTIONS**

**Current max\_connections = 120**

**Current threads\_connected = 109**

**Historic max\_used\_connections = 121**

**The number of used connections is 100% of the configured maximum.**

**You should raise max\_connections**

**MEMORY USAGE**

**Max Memory Ever Allocated : 363 M**

**Configured Max Per-thread Buffers : 318 M**

**Configured Max Global Buffers : 42 M**

**Configured Max Memory Limit : 360 M**

**Physical Memory : 4135079936 bytes**

**Max memory limit seem to be within acceptable norms**

**KEY BUFFER**

**Current MyISAM index space = 618 M**

**Current key\_buffer\_size = 16 M**

**Key cache miss rate is 1 : 10**

**Key buffer free ratio = 0 %**

**Using key\_buffer\_size > 4GB will cause instability in versions prior to 5.0.52**

**See Bug#5731, Bug#29419, Bug#29446**

**You could increase key\_buffer\_size**

**It is safe to raise this up to 1/4 of total system memory;**

**assuming this is a dedicated database server.**

**QUERY CACHE**

**Query cache is enabled**

**Current query\_cache\_size = 16 M**

**Current query\_cache\_used = 12 M**

**Current query\_cache\_limit = 1 M**

**Current Query cache Memory fill ratio = 76.06 %**  
**Current query\_cache\_min\_res\_unit = 4 K**  
**MySQL won't cache query results that are larger than query\_cache\_limit in size**

## **SORT OPERATIONS**

**Current sort\_buffer\_size = 2 M**  
**Current read\_rnd\_buffer\_size = 256 K**  
**Sort buffer seems to be fine**

## **JOINS**

**Current join\_buffer\_size = 132.00 K**  
**You have had 281 queries where a join could not use an index properly**  
**You should enable "log-queries-not-using-indexes"**  
**Then look for non indexed joins in the slow query log.**  
**If you are unable to optimize your queries you may want to increase your join\_buffer\_size to accommodate larger joins in one pass.**

**Note! This script will still suggest raising the join\_buffer\_size when ANY joins not using indexes are found.**

## **OPEN FILES LIMIT**

**Current open\_files\_limit = 3000 files**  
**The open\_files\_limit should typically be set to at least 2x-3x that of table\_cache if you have heavy MyISAM usage.**  
**Your open\_files\_limit value seems to be fine**

## **TABLE CACHE**

**Current table\_cache value = 1000 tables**  
**You have a total of 2409 tables**  
**You have 1000 open tables.**  
**Current table\_cache hit rate is 1%**  
**, while 100% of your table cache is in use**  
**You should probably increase your table\_cache**

Estos datos deben ser recogidos frecuentemente para ver si estamos dentro de lo que el autor considera adecuado, además pueden cambiar significativamente el numero de usuarios en periodos de pruebas,etc.

Otra herramienta muy útil, por la cantidad de recomendaciones y análisis estadísticos es el PHPMyAdmin, que además de hacer un exhaustivo análisis de las estadísticas, nos despliega la información con tips en el caso de que los valores sean altos y nos subraya lo que considera valores altos.

Applications Places System Mon Aug 1, 10:57 PM

courses.quimica.fq.edu... x

courses.quimica.fq.edu uy/admin/mysql/server\_status.php?token=XWyU5xhW0H

Servidor: localhost

Bases de datos Estado actual Variables Juegos de caracteres Motores Procesos Exportar Importar

### Información acerca del tiempo de ejecución del proceso principal

[Actualizar] [Resetear] [⌂]

Este servidor MySQL ha estado activo durante 0 días, 2 horas, 52 minutos y 35 segundos. Se inició en 01-08-2011 a las 20:03:13.

[consulta SQL] [InnoDB] [NDB] [SSL] [Gestor] [Cache de consultas] [Procesos] [Log binario] [Datos temporales] [Delayed inserts] [Cache principal] [Vínculos (Joins)] [Replicación] [Organizando (sorting)] [Tablas] [Coordinador de transacción]

Tráfico del servidor: Estas tablas muestran las estadísticas de tráfico en la red de este servidor MySQL desde su inicio.

Tráfico	por hora	por minuto	por segundo	Conexiones	por hora	%
Recibido	311 MB	108 MB	107,96	Número máx. de conexiones concurrentes	121	---
Enviado	4,365 MB	1,518 MB	1,518	Abortados	1,337	464,62 3,82%
Total	4,677 MB	1,626 MB	1,626	Total	578	200,95 1,65%
					35 k	12,18 k 100,00%

Estadísticas de consulta: Desde su inicio, 1,118,165 consultas han sido enviadas al servidor.

Total	por hora	por minuto	por segundo	Tipo de consulta	por hora	%
1,118 k	388,74 k	6,48 k	107,96	admin commands	29 k	9,985,44 2,65%
				alter db	0	0,00 0,00%
				alter table	0	0,00 0,00%
				analyze	0	0,00 0,00%
				backup table	0	0,00 0,00%
				begin	0	0,00 0,00%
				call procedure	0	0,00 0,00%
				change db	59 k	20,64 k 5,48%
				change master	0	0,00 0,00%
				check	2,410	837,86 0,22%
				checksum	0	0,00 0,00%
				commit	0	0,00 0,00%
				create db	0	0,00 0,00%
				create function	0	0,00 0,00%
				create index	0	0,00 0,00%
				create table	0	0,00 0,00%
				create user	0	0,00 0,00%
				delete	253	87,96 0,02%
				rollback	0	0,00 0,00%
				savepoint	0	0,00 0,00%
				select	298 k	72,29 k 19,18%
				set option	33 k	11,36 k 3,02%
				show binlog events	0	0,00 0,00%
				show binlogs	6	2,09 0,00%
				show charsets	15	5,21 0,00%
				show collations	15	5,21 0,00%
				show column types	0	0,00 0,00%
				show create db	0	0,00 0,00%
				show create table	0	0,00 0,00%
				show databases	16	5,56 0,00%
				show errors	0	0,00 0,00%
				show fields	1,959	684,54 0,18%
				show grants	6	2,09 0,00%
				show innodb status	0	0,00 0,00%
				show keys	250	86,91 0,02%
				show logs	0	0,00 0,00%

Si bien tiene el inconveniente de funcionar vía PHP (Seguridad:cuidado con la configuración), nos aporta mucha información muy importante tal como Máximo numero de conexiones concurrentes, consultas por hora,etc.

Hay que tener en cuenta que este tipo de herramientas pueden ser muy cuestionadas en el área de seguridad, lo que es cierto que no podemos estar sin usarlas, ya que el riesgo de que se nos caiga un sitio en medio de periodo de exámenes por falta de control es tan malo como tener un sitio con una baja seguridad, para ello es importante realizar testeos de carga antes de tener un sitio activo, y poder realizar, una vez que modelizamos nuestros parámetros de ajustes y de verificación, realizar estas consultas directamente sin herramientas y guardar logs de las mismas.

## Ajustes de parámetros

Para el ajuste de parámetros es recomendable realizarlo en etapas, por lo menos 2, un asuste grueso, que corresponde a los siguientes parámetros :

Current max\_connections  
Max Memory Ever Allocated  
Configured Max Per-thread Buffers  
Configured Max Global Buffers  
Configured Max Memory Limit  
Current MyISAM index space  
Current key\_buffer\_size  
Current query\_cache\_size  
Current query\_cache\_used  
Current query\_cache\_limit  
Current open\_files\_limit  
Current table\_cache

Esas variables de configuración nos darán un ajuste grueso y de alto impacto en muchos casos, luego en un segundo análisis buscaremos ajustar variables como ,

Key\_read\_requests

Key\_reads.

Hay que realizar el ajuste con método, se deben ir guardando los conjuntos de parámetros, junto con algunas medidas de eficiencia, de manera de optimizar los resultados.

## Mas halla y hacia el infinito

Vamos a llegar a un punto en que el limite de memoria física, o cantidad de procesos nos impida un mejor ajuste, en estos casos es necesario un cambio de arquitectura del Moodle.

Existen muchas posibilidades, entre ellas tenemos :

- a.- Un servidor dedicado para la base de datos en conexión interna con el servidor web
- b.- Mas de un servidor de bases de datos ( en particular es interesante uno de lectura y otro de escritura, ya que las escrituras de logs en el Moodle llevan mucho tiempo)
- c.-Realizar análisis de Sentencias SQL del Moodle en busca de operaciones muy lentas y/o costosas.

## Conclusiones

No podemos iniciar un servidor de Moodle sin tener dimensionado el conjunto de cursos y

usuarios del sistema, en lo inmediato y deberemos tratar de modelizar los requerimientos del sistema de base de datos y recursos del servidor, una mala configuración del servidor de Bases de datos dejara inservible el sistema en los momentos que mayor cantidad de usuarios los necesiten y en ese momento poca cosa podremos hacer para salvar el desastre.

Se debe realizar una minuciosa medición de requerimientos antes de que el servidor este en producción y mantener medidas periódicas con alguna herramienta, para ver la evolución del mismo.

También se debe esta atento a las actualizaciones, tanto del Moodle como del Mysql.

Para el primero, realizar pruebas fuera del servidor de producción y analizar el comportamiento de la Base de datos, ya que en al comienzo de una nueva versión hay muchas SQL no optimizadas y pueden llegar a afectar mucho el rendimiento ( recordar versión 1.82 del Moodle).

Con respecto a la versión del Mysql también corren las mismas advertencias, pero hay que revisar muchos los foros de usuarios y del proveedor.

## Bibliografía

- <http://dev.mysql.com/doc/refman/5.0/es/server-status-variables.html>
- <http://dev.mysql.com/doc/refman/5.0>
- Mastering phpMyAdmin 3.3.x for Effective MySQL Management, ISBN : 1849513546
- Capacidad de servidor MoodleDocs, [http://docs.moodle.org/20/en/User\\_site\\_capacities](http://docs.moodle.org/20/en/User_site_capacities)
- Configuración Moodle de alta disponibilidad, Moodlemoot09, Felipe Retortillo, <http://www.slideshare.net/felipe.retortillo/configuracin-moodle-alta-disponibilidad>
- Buen inicio para ajustar Mysql y herramientas de [http://gert.sos.be/nl/articles/mysql\\_tuning/](http://gert.sos.be/nl/articles/mysql_tuning/)
- Script para ajuste de Mysql, <http://forge.mysql.com/projects/project.php?id=44>
- Forecasting MySQL Scalability with the Universal Scalability Law ,Baron Schwartz and Ewen Fortune , <http://www.percona.com/files/white-papers/forecasting-mysql-scalability.pdf>
- Causes of Downtime in Production MySQL Servers ,Baron Schwartz, <http://www.percona.com/files/white-papers/causes-of-downtime-in-mysql.pdf>
- Preventing MySQL Emergencies ,Baron Schwartz, <http://www.percona.com/files/white-papers/preventing-mysql-emergencies.pdf>