



Pruebas de performance en una plataforma para exámenes online utilizando moodle.

Emilio Penna

Servicio Central de Informática

Universidad de la República - Uruguay

emilio.penna@seciu.edu.uy

WOPR Latam 2023

Universidad de la República (Udelar)

- Principal universidad en Uruguay
- 100,000 estudiantes activos
- 11,500 docentes
- 450 carreras
- 20 facultades
- 25 instancias de moodle
- Autenticación centralizada con proveedor de identidad SAML (shibboleth)

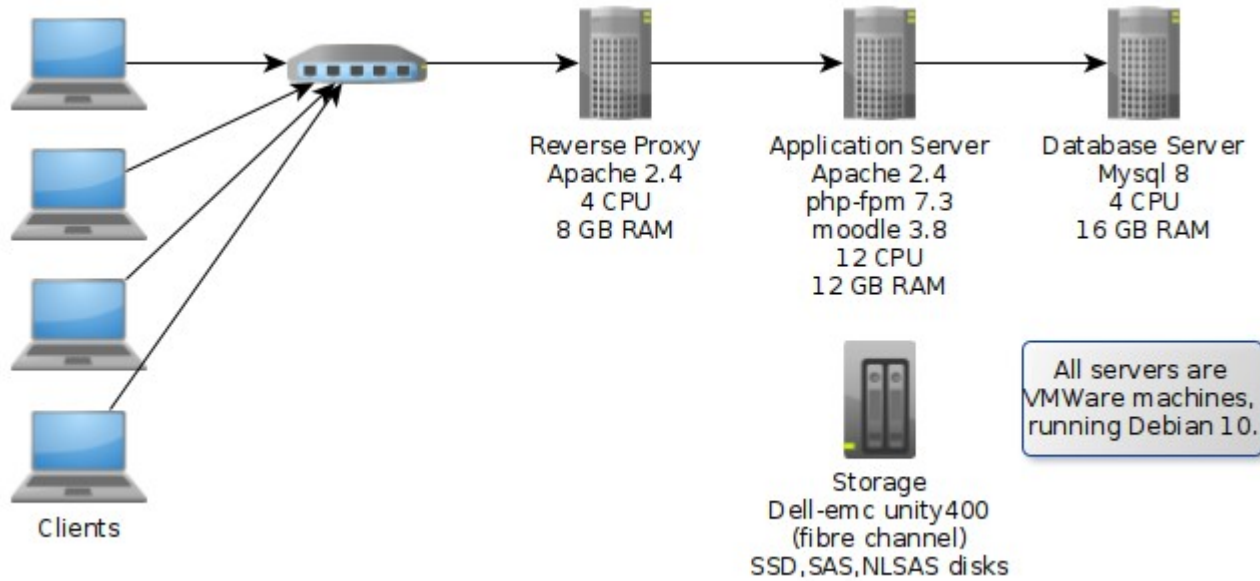


UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Exámenes online

- 2020: Necesidad de dictar clases y tomar exámenes online
- Varias facultades suelen tener exámenes con más de 1000 estudiantes
- La universidad usa moodle (Learning Management System en PHP) para enseñanza online.
- Para poder dar un mejor servicio en los exámenes con gran cantidad de estudiantes, hicimos una instalación de moodle de uso exclusivo para exámenes y evaluaciones (mediante una agenda se controla que no haya más de una evaluación al mismo tiempo).

Arquitectura (on premise)



Las preguntas

- ¿Cuántos estudiantes soporta en forma simultánea en un examen online?
- ¿Que mejoras podemos realizar (en un corto plazo y con la infraestructura existente) para aumentar la capacidad?
- ¿Cual es el máximo de usuarios soportados con un solo servidor de aplicaciones?

Antecedentes



https://docs.moodle.org/403/en/JMeter_test_plan_generator



JMeter test plan generator

[Main page](#) ► [Managing a Moodle site](#) ► [Developer tools](#) ► **JMeter test plan generator**

The JMeter test plan generator allows you to run performance tests over a Moodle course, it works along with <https://github.com/moodlehq/moodle> you can find more info about how to use it in <https://github.com/moodlehq/moodle-performance-comparison/blob/master/README.md>

Contents [\[hide\]](#)

- 1 [Requirements](#)
- 2 [Finding the generator tool](#)
- 3 [Usage](#)
 - 3.1 [Options](#)

Requirements

This tool is for developer use only and should not be used on a live site. It is disabled unless you set the debugging option to DEVELOPER.

Finding the generator tool

The tool can be found in:

- Web interface
- The Site Administration block -> Development -> Make JMeter test plan

Planificando la prueba

- Decidimos realizar pruebas de carga con jmeter, desarrollamos un script que simula un estudiante realizando el examen.
- El examen lo modelamos como un moodle quiz con varias páginas, distintos tipos de preguntas y algunas imágenes. Moodle posee un generador de script jmeter, pero no hace pruebas sobre quiz.
- Instalamos un ambiente igual al de producción, instalamos monitores (nmon, nagios), scripts para extraer logs y reiniciar servicios luego de cada prueba, respaldo inicial de la base para partir de los mismos datos iniciales.
- Ejecución usando jmeter en 2 PC en la misma red (generadoras)
Llegamos a 1500 threads por PC

Planificando la prueba

- Línea base de 200 usuarios virtuales

A	B	C	D	E	F	G	H	I	J	K
Id	Ejecución	Comentario del ESCENARIO	Cant UV	Cant iter	Cant evaluaciones	Ejecutado por	Fecha ejecución	Hora Inicio	Hora Fin	Duración
1	200uv – 1 iter – solo evaluación	200uv	200	1	200	vparula	13/10/2020	17:01:04	17:08:05	0:07:01
2	250uv – 1 iter – solo evaluación	250uv	250	1	250	vparula	13/10/2020	17:29:08	17:36:30	0:07:22
3	250uv – 1 iter – solo evaluación	250uv	250	1	250	vparula	21/10/2020	15:58:42	16:08:25	0:09:43
4	500uv – 1 iter – solo evaluación	500uv	500	1	500	vparula	21/10/2020	16:23:30	16:35:03	0:11:33
5	1000uv – 1 iter – solo evaluación	1000uv	1,000	1	1,000	vparula	21/10/2020	17:00:23	17:10:48	0:10:25

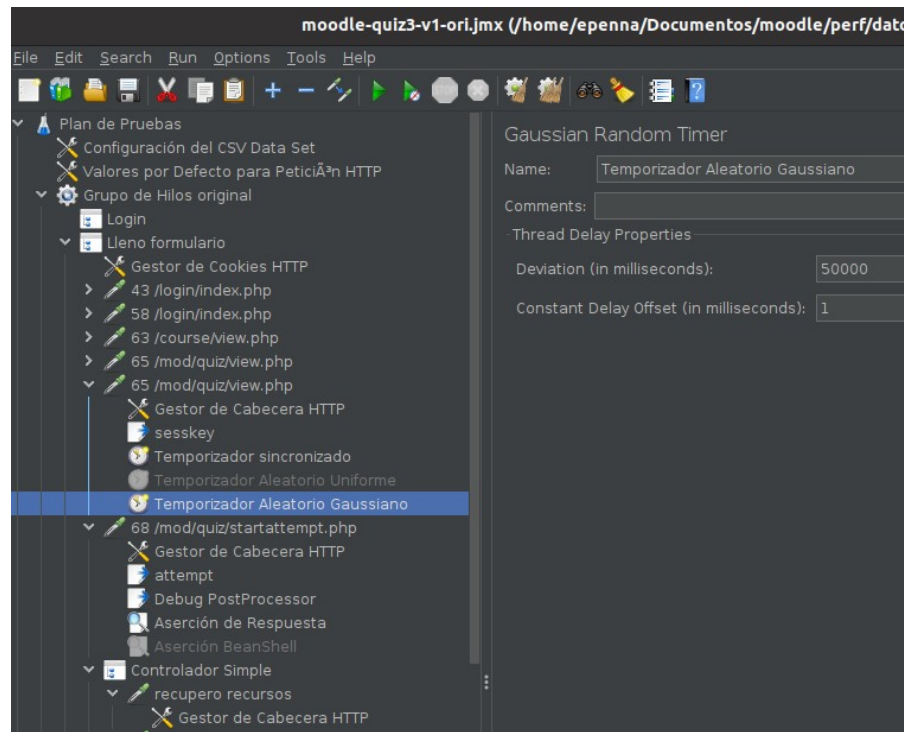
Configuración de la prueba

- Momento crítico: inicio del examen (cuando se habilita el ingreso al examen, entran todos los estudiantes “al mismo tiempo”)
- Hicimos un análisis de logs de producción para ver la distribución de tiempos de ingreso de los estudiantes al examen (startattempt.php)

Segundo	Cant Usuarios	Porcentaje
0	0	0 %
10	334	13 %
20	731	29 %
30	1100	44 %
40	1416	57 %
50	1685	67 %
60	1835	73 %
70	2020	81 %
80	2187	87 %
90	2295	92 %
100	2365	95 %
110	2418	97 %
120	2451	98 %

Configuración de la prueba

- Login: se realiza en los 10 minutos previos a la prueba (10 min ramp up)
- Synchronized timer para esperar que todos los usuarios virtuales (threads) hayan entrado
- Vimos que podíamos tener una aproximación muy cercana a la distribución real de tiempos del inicio usando un “Gaussian random timer” (desviación 50s)



Ejecución

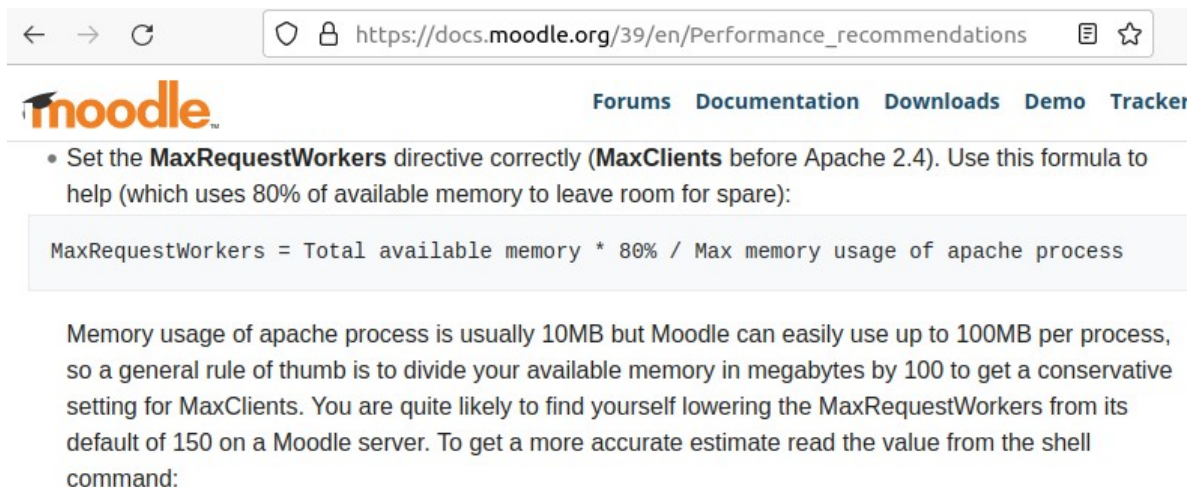
- Hicimos ejecuciones incrementales hasta la falla. Luego revisión de logs, análisis de recursos limitantes, ajustes, reiteración de prueba.
- Al llegar a resultados relevantes, repetimos para verificar (partiendo de los datos iniciales)
- Llegamos a ejecutar pruebas con 2500 usuarios virtuales (jmeter threads) sin errores y con buenos tiempos de respuesta (560ms avg, 800ms std dev)
- Con 3000 usuarios virtuales, los tiempos de respuesta se volvieron inaceptables. Carga de CPU muy alta (>200%) durante casi toda la prueba

Ejecución – primer obstáculo

- Primeras pruebas: 100-200 UV (usuarios virtuales, jmeter threads)
- Ya comienza a fallar:
- Log Jmeter: org.apache.http.NoHttpResponseException: eva-perf.seciu.edu.uy:443 failed to respond
- Log Apache: server reached MaxRequestWorkers setting, consider raising the MaxRequestWorkers setting.
- Se estaba llegando al máximo de procesos apache (usábamos prefork mpm) y apache dejaba de responder.
- Se había configurado de acuerdo a la documentación de moodle, según la RAM disponible y memoria consumida por cada proceso.

Ejecución – primer obstáculo

- Configuración según la documentación de moodle:
- $\text{MaxRequestWorkers } 12000 * 0.8 / 100 = 96$
- Considerando memoria compartida: $12000 * 0.8 / 20 = 480$



The screenshot shows a web browser window with the URL https://docs.moodle.org/39/en/Performance_recommendations. The Moodle logo is visible in the top left, and navigation links for Forums, Documentation, Downloads, Demo, and Tracker are in the top right. A bullet point states: "Set the **MaxRequestWorkers** directive correctly (**MaxClients** before Apache 2.4). Use this formula to help (which uses 80% of available memory to leave room for spare):"

```
MaxRequestWorkers = Total available memory * 80% / Max memory usage of apache process
```

Memory usage of apache process is usually 10MB but Moodle can easily use up to 100MB per process, so a general rule of thumb is to divide your available memory in megabytes by 100 to get a conservative setting for MaxClients. You are quite likely to find yourself lowering the MaxRequestWorkers from its default of 150 on a Moodle server. To get a more accurate estimate read the value from the shell command:

Ejecución – primer ajuste

- Solución: cambiamos el módulo de multiprocesamiento de apache, pasamos a usar event mpm (multihilo)
- También cambiamos a php-fpm, por conocer antecedentes donde se mencionaba un mejor rendimiento y menor consumo de RAM.
- Aumentamos máximos de procesos apache y php para ver hasta donde llegaba

Ejecución – segundo obstáculo

Error: Database connection failed

It is possible that the database is overloaded or otherwise not running properly.

The site administrator should also check that the database details have been correctly specified in config.php

Ejecución – segundo obstáculo

- Se alcanzaba rápidamente la max DB connections configurada (estaba por defecto en 150).



- Set the **maximum number of connections** so that your users will not see a "Too many connections" message. Be careful that this may have an impact on the total memory used. MySQL connections usually last for milliseconds, so it is unusual even for a heavily loaded server for this value to be over 200.

Ejecución – segundo obstáculo

- Se aumentó la cantidad de conexiones permitidas en la configuración de MySQL: `max_connections=500` (en el monitoreo no se ve problemas en el uso de recursos)
- Variable interesante: `max_used_connections` en las pruebas llegó a 338
- Setting de MySQL 8, aprovechando que es un servidor dedicado:
`innodb_dedicated_server = ON`

Ejecución – tercer obstáculo y ajustes

- Se observaba un uso muy alto de CPU en el application server, coincidiendo con empeoramiento de tiempos de respuesta y fallas. Entendimos que lo que estaba limitando la cantidad de usuarios soportados era la CPU.
- Pasamos de 8 a 12 cores en el servidor de aplicaciones
- Logramos un número mayor de usuarios soportados.
- Otros ajustes: configuración de opcache

Resultados (2500 UV)

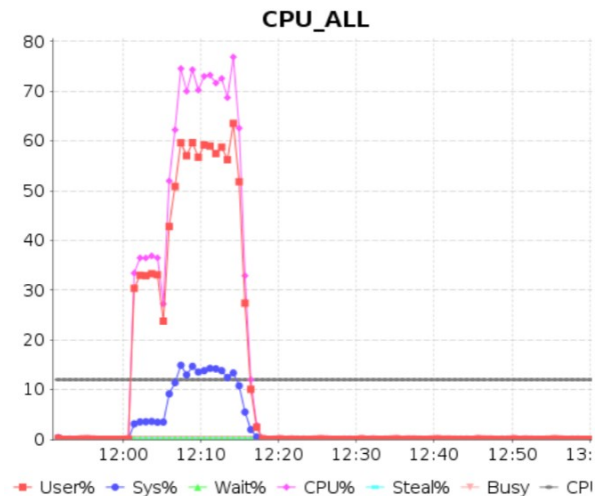
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento
43 /login/index....	2500	105	82	686	49,20	0,00%	8,9/sec
58 /login/index....	2500	400	230	1677	136,67	0,00%	8,9/sec
63 /course/vie...	2500	145	110	650	42,20	0,00%	9,0/sec
65 /mod/quiz/vi...	5000	506	109	7031	698,58	0,00%	11,0/sec
68 /mod/quiz/st...	2500	1716	224	10051	1782,19	0,00%	14,2/sec
recupero recur...	15000	493	6	6894	639,17	0,00%	85,6/sec
Lleno primer pr...	2500	1837	139	11793	2219,19	0,00%	12,6/sec
Lleno segunda...	2500	918	135	8063	1450,42	0,00%	11,6/sec
paso de pagina	17500	635	121	15167	968,29	0,00%	28,8/sec
Lleno tercer pr...	2500	430	138	6898	562,52	0,00%	11,0/sec
Lleno cuarta pr...	2500	513	134	3841	560,09	0,00%	9,9/sec
Pregunta 5	2500	481	139	4727	543,74	0,00%	9,1/sec
Pregunta 6	2500	510	140	4576	559,08	0,00%	9,0/sec
Pregunta 7	2500	469	137	4122	524,95	0,00%	8,4/sec
Pregunta 8	2500	426	143	3486	448,65	0,00%	8,1/sec
Pregunta 9	2500	380	144	4089	378,03	0,00%	8,1/sec
Pregunta 10	2500	372	139	7591	382,06	0,00%	7,3/sec
Pregunta 11	2500	409	137	5200	467,71	0,00%	7,2/sec
Pregunta 12	2500	464	132	6170	601,96	0,00%	7,1/sec
Pregunta 13	2500	518	127	6135	707,93	0,00%	7,0/sec
Pregunta 14	2500	552	128	6061	757,94	0,00%	6,9/sec
Pregunta 15	2500	464	128	6053	673,44	0,00%	6,6/sec
Pregunta 16	2500	363	125	3347	543,40	0,00%	6,9/sec
finish	5000	443	147	6183	549,26	0,00%	13,8/sec
Total	92500	561	6	15167	885,85	0,00%	92,9/sec

App server performance (2500 UV)

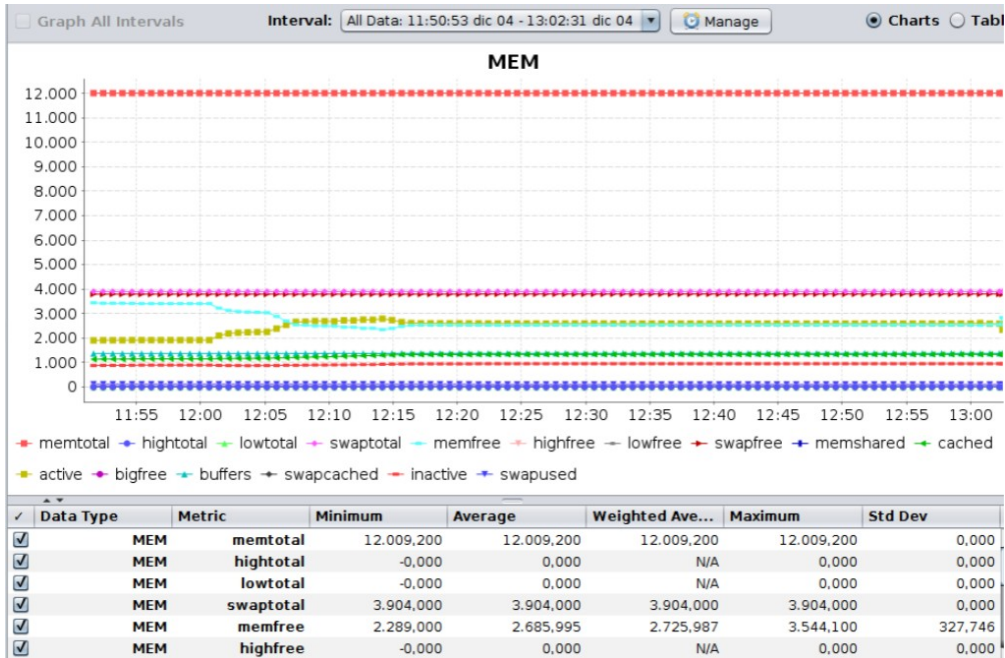
- Máximo uso observado de CPU: load average entre 12 y 13 (110%) durante los 3 minutos de mayor intensidad (12 cores)
- Max procesos Apache: 14 (150 threads each)
- Max procesos php-fpm: 28
- Memoria en el momento de mayor carga (no usó swap):

Total	used	free	shared	buff/ cache	available
11Gi	1,7Gi	2,4 Gi	111Mi	7,7Gi	9,9Gi

Monitoraggio con nmon



✓	Data T...	Metric	Minimum	Average	Weigh...	Maximum
✓	CPU_ALL	User%	0,000	10,058	51,978	76,100
✓	CPU_ALL	Sys%	0,000	2,088	12,207	21,200
✓	CPU_ALL	Wait%	0,000	0,007	0,440	1,400
✓	CPU_ALL	CPU%	0,000	12,146	63,750	93,100



✓	Data Type	Metric	Minimum	Average	Weighted Ave...	Maximum	Std Dev
✓	MEM	memtotal	12.009,200	12.009,200	12.009,200	12.009,200	0,000
✓	MEM	hightotal	-0,000	0,000	N/A	0,000	0,000
✓	MEM	lowtotal	-0,000	0,000	N/A	0,000	0,000
✓	MEM	swaptotal	3.904,000	3.904,000	3.904,000	3.904,000	0,000
✓	MEM	memfree	2.289,000	2.685,995	2.725,987	3.544,100	327,746
✓	MEM	highfree	-0,000	0,000	N/A	0,000	0,000

¿Que dice la documentación?



[Forums](#) [Documentation](#) [Downloads](#) [Demo](#) [Tracker](#)

Hardware configuration

Note: The fastest and most effective change that you can make to improve performance is to **increase the amount of RAM on your web server** - get as much as possible (e.g. 4GB or more). Increasing primary memory will reduce the need for processes to swap to disk and will enable your server to handle more users.

Otras pruebas de performance realizadas en moodle

- Comparación de caches: ejecución de la prueba de examen online con file cache vs memcached (sorprendentemente nuestro file cache dio mejor resultado)
- Util: Moodle Site administration -> Plugins -> Caching -> Test performance
- Pruebas de carga con tareas donde se suben archivos
- Pruebas de carga usando Quiz Access Proctoring Plugin (toma fotografías al estudiante durante la prueba). En nuestras pruebas, el servidor soportaba aprox. 75% de la carga que soportaría si no se usara el plugin.

Recursos

- Jmeter script (moodle 3.8), test data, y documentos:

<https://www.proyectos.udelar.edu.uy/redmine/projects/moodleperf/wiki/>

- El script jmeter fue realizado por Valentina Parula (Seciu - Udelar). Las pruebas fueron realizadas por Valentina Parula y Emilio Penna



Conclusiones

- Resultados que nos parecieron muy buenos, considerando que se usa un solo application server.
- Deployment simple, con herramientas conocidas.
- No hizo falta agregar más RAM
- Uso intensivo de CPU. Fue el último recurso limitante observado.
- El sistema se está usando en producción desde fines de 2020 con más de 250 exámenes, de hasta 1250 estudiantes, sin problemas.
- Los scripts y datos de prueba generados, se disponibilizaron para que otras facultades los utilicen para testear y mejorar sus sistemas.



Preguntas?

emilio.penna@seciu.edu.uy

WOPR Latam 2023