

Escenario y Pruebas de Estrés

API REST y Batch

Entorno de prueba

Se utiliza una maquina virtual de Google Cloud con 1 vCPU, 1 GiB RAM, 10 GiB en almacenamiento, sobre esta se abrieron en el firewall, en http protocolo tcp los puertos 3000, 4000, 5000, 5672 y 15672, puertos que son usados para la comunicación de los distintos servicios, otras características por defecto se pueden ver en la siguiente tabla

Name	Enforcement order	Type ↑	Rule priority	Targets	Protocols and ports	Action
▼ vpc-firewall-rules	1	VPC firewall rules				
default-allow-https		Ingress firewall rule	1000	Tags: https-server	tcp:443	Allow
default-allow-http		Ingress firewall rule	1000	Tags: http-server	tcp:3000, 4000, 5000, 5672, 15672	Allow
default-allow-icmp		Ingress firewall rule	65534	Apply to all	icmp	Allow
default-allow-rdp		Ingress firewall rule	65534	Apply to all	tcp:3389	Allow
default-allow-internal		Ingress firewall rule	65534	Apply to all	tcp:0-65535 udp:0-65535 icmp	Allow
default-allow-ssh		Ingress firewall rule	65534	Apply to all	tcp:22	Allow

En materia de software se utilizo una imagen de sistema debain-11-bullseye-v20220920, en esta maquina se instalaron:

- Docker 3
- Contenedores Docker
 - Base de datos PostgreSQL
 - Message bróker RabbitMQ
- Python 3.9.2
 - Aplicación web para API
 - Tecnología: Framework Flask
 - Componente Converter
 - Tecnologías:
 - Framework Flask
 - Librería de conversión Pydub
 - Componente Notifications

- Tecnologías:
 - Framework Flask
 - Librería Yagmail
- Librerías
 - Flask 1.1.2
 - Flask-JWT 0.2.0
 - Flask-JWT-Extended 4.2.3
 - Flask-RESTful 0.3.9
 - Flask-SQLAlchemy 2.5.1
 - Itsdangerous 2.0.1
 - Jinja2 3.0.1
 - marshmallow 3.12.1
 - marshmallow-sqlalchemy 0.25.0
 - PyJWT 2.1.0
 - pytz 2021.1
 - six 1.16.0
 - SQLAlchemy 1.4.15
 - Werkzeug 2.0.1
 - gunicorn 20.1.0
 - faker
 - pika 1.3.1
 - psycpg2-binary 2.9.4
 - psycpg2 2.9.4
 - pydub 0.25.1
 - yagmail 0.15.293
 - requests
 - pathlib 1.0.1

Criterios de aceptación

- Tiempo de respuesta: no debe ser superior a 1.5s en promedio
- Utilización de recursos: No se establecieron límites formales para este parámetro, aun con ello se busca que los consumos de memoria y de CPU no pasen de 90% ya que esto suele acarrear riesgos de que la máquina virtual deje de funcionar del todo, aclarado esto si la máquina presenta un uso de CPU o memoria del 100% y se mantiene en funcionamiento durante las pruebas de manera consistente esto no será un factor que genere obligación para hacer ajustes.
- Rendimiento: Se ha definido que el límite inferior de rendimiento está en iniciar un nuevo procesamiento cada 600 segundos (10min)
- Al menos el 99% de las peticiones deben ser exitosas

De los criterios mencionados cabe resaltar la ausencia de un valor definido para la utilización de recursos, esto podría implicar fallos en producción que no son tan fáciles de detectar en ambientes de pruebas

Escenarios de prueba

Escenario 1

Máxima cantidad de requests HTTP por minuto que soporta la aplicación web con usuarios (al menos un usuario) que cuenten con 30 archivos disponibles

En las pruebas de estrés el tiempo de respuesta promedio de la aplicación debe ser de máximo 1.500 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de requests de la prueba. En caso de que durante una prueba se generen más de un 1% de errores en los requests de la prueba, se concluye que la aplicación NO soporta la cantidad de requests de la prueba.

Para esta prueba se realizan pruebas ajustando la cantidad de usuarios y el tiempo en que se tarda en llegar a la cantidad de usuarios máximos haciendo 2 peticiones cada uno

Escenario 2

Máxima cantidad de archivos que pueden ser procesados por minuto en la aplicación local, la prueba se da por fallida al llegar al punto en que el tiempo para iniciar el procesamiento de un archivo enviado por un usuario supere los 10 minutos (600 segundos). El archivo enviado a convertir durante las pruebas debe ser de un tamaño mínimo de 5 MB.

Parámetros de configuración

Previo a la realización de las pruebas se debe realizar la creación de un usuario y un login para obtener el token de autorización que es usado durante la prueba

Para las pruebas se utiliza JMeter con la siguiente configuración:

CONFIGURACION JMETER

Resultados Escenario 1

Para este primer escenario se realizaron algunas pruebas exploratorias, tras leer la documentación se encontró que para cada prueba en JMeter se pueden cambiar el numero de usuarios, el tiempo en que se llegara a ese numero máximo de usuarios y finalmente la cantidad de veces que se llevara a cabo la prueba, tras las pruebas exploratorias se decidió que solo el numero de usuarios seria cambiado y que se iniciaría con los valores aconsejados en la pagina de JMeter que son 100 usuarios, 100 segundos y 1 iteración completa, en dicha prueba no se logro llegar a un tiempo de respuesta superior a 1.5s y se opto por hacer pruebas con 150 usuarios, luego 170 y luego 185, estos valores fueron escogidos considerando el promedio de tiempo de respuesta del valor anterior, sin embargo al pasar a 180 y después de realizar las anteriores pruebas de manera repetida se evidencio que el tiempo de respuesta era mucho peor aun cuando se repetían pruebas con usuarios probados anteriormente, de lo cual se opto por reiniciar la maquina con cada nuevo set de pruebas y se obtuvo el limite de tiempo d respuesta al tener 200 usuarios, las siguientes imágenes tienen como titulo el numero de usuarios de la prueba (ya que los otros valores son constantes) y son el Aggregation Report, en el cual la tercera columna indica al final el promedio total, seguido de esto una captura de pantalla del estado de la maquina virtual en un momento al azar mostrando uso de CPU y memoria en particular, luego el Graph Results que muestra una grafica con el comportamiento del tiempo promedio de respuesta y el Throughput (tasa de transferencia efectiva), finalmente una captura de pantalla con la configuración de dicha prueba

170

Aggregate Report

Aggregate Report

Name:

Aggregate Report

Comments:

Write results to file / Read from file

Filename

C:\Users\JorgeS\Downloads\Maestria Los Andes\Nube\PruebasCargaEntrega1\Aggregate-170-100.txt

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Label ↑	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
HTTP Request convert to ogg	170	835	758	1285	1546	2522	196	4010	0.00%	1.7/sec	0.32	26692.06
HTTP Request convert to wav	170	717	589	1128	1431	2983	185	3114	0.00%	1.7/sec	0.32	26799.93
TOTAL	340	776	695	1225	1637	2983	185	4010	0.00%	3.3/sec	0.63	53243.84

Estatus de la VM

CPU[100.0%]												
Mem[586M/977M]												
Swp[292M/975M]												
Tasks: 98, 107 thr; 1 running												
Load average: 1.39 1.33 1.50												
Uptime: 07:20:48												
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command	
39342	root	20	0	225M	49940	37428	R	96.7	5.0	0:10.11	ffmpeg -y -f wav -i /tmp/tmpy3jy4mxd -acodec libvorbis	
28214	root	20	0	9120	2300	1064	R	0.7	0.2	0:50.01	htop	
38247	root	20	0	695M	3524	0	S	0.7	0.4	0:00.32	/usr/bin/containerd-shim-runc-v2 -namespace moby -id ad	
38254	root	20	0	695M	3524	0	S	0.7	0.4	0:00.03	/usr/bin/containerd-shim-runc-v2 -namespace moby -id ad	
1	root	20	0	99452	3000	1344	S	0.0	0.3	0:02.00	/sbin/init	
183	root	20	0	48336	2684	2268	S	0.0	0.3	0:02.58	/lib/systemd/systemd-journald	
202	root	20	0	21592	976	652	S	0.0	0.1	0:00.20	/lib/systemd/systemd-udev	
236	systemd-t	20	0	88440	104	0	S	0.0	0.0	0:00.12	/lib/systemd/systemd-timesyncd	

Graph Results

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

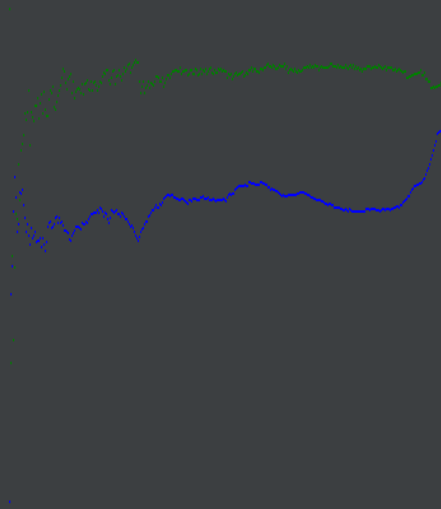
Filename C:\Users\JorgeS\Downloads\Maestria Los Andes\Nube\PruebasCargaEntrega1\Graph-170-100.txt

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Graphs to Display ☐ Data ☒ Average ☐ Median ☐ Deviation ☒ Throughput

1225 ms



0 ms

No of Samples 340

Deviation 407

Latest Sample 686

Throughput 107.612/minute

Average 776

Median 695

Thread Group

Name:

Prueba tiempo respuesta

Comments:

Máxima cantidad de request por minuto que soporta la app

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users):

170

Ramp-up period (seconds):

100

Loop Count:

☐ Infinite

1

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Aggregate Report

Name:

Comments:

Write results to file / Read from file

Filename: Log/Display Only: ☐ Errors ☐ Successes

Label ↑	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
HTTP Request convert to ogg	185	5943	4499	12938	15518	18520	493	19799	0.00%	1.6/sec	0.31	25699.87
HTTP Request convert to wav	185	5943	4185	13867	15424	17006	465	17043	0.00%	1.6/sec	0.32	26508.90
TOTAL	370	5943	4374	13867	15493	17748	465	19799	0.00%	3.2/sec	0.61	51194.52

```

CPU[|||||100.0%] Tasks: 100, 108 thr; 1 running
Mem[|||||596M/977M] Load average: 3.74 3.90 2.51
Swp[|||||273M/975M] Uptime: 07:29:23

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
40786	root	20	0	225M	47640	37428	R	98.0	4.8	0:08.99	ffmpeg -y -f wav -i /tmp/tmpufhwnaik -acodec libvorbis
28214	root	20	0	9120	2260	1024	R	0.7	0.2	0:55.05	htop
1	root	20	0	99452	2880	1268	S	0.0	0.3	0:02.03	/sbin/init
183	root	20	0	48336	2136	1724	S	0.0	0.2	0:02.64	/lib/systemd/systemd-journald
202	root	20	0	21592	968	656	S	0.0	0.1	0:00.22	/lib/systemd/systemd-udev
236	systemd-t	20	0	88440	1340	1228	S	0.0	0.1	0:00.13	/lib/systemd/systemd-timesyncd
263	systemd-t	20	0	88440	1340	1228	S	0.0	0.1	0:00.00	/lib/systemd/systemd-timesyncd
276	root	20	0	6748	500	396	S	0.0	0.0	0:00.11	/usr/sbin/cron -f
277	messagebu	20	0	8280	524	112	S	0.0	0.1	0:00.27	/usr/bin/dbus-daemon --system --address=systemd: --nofo
280	root	20	0	22236	248	0	S	0.0	0.0	0:00.23	/lib/systemd/systemd-logind

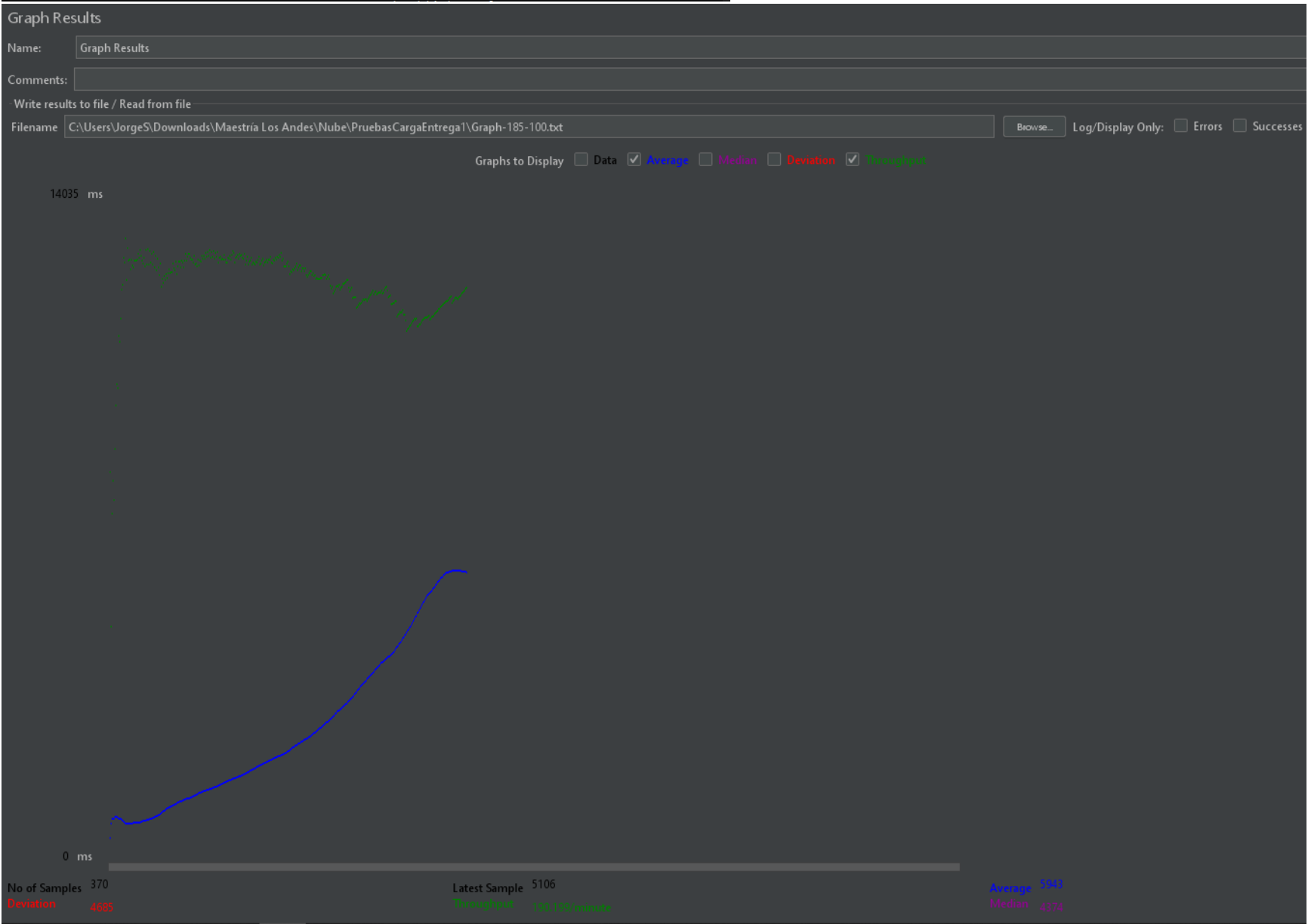
CPU[|||||100.0%]|

Mem[|||||556M/977M]|

Swp[|||||425M/975M]|

Tasks: 92, 111 thr; 1 running
Load average: 0.87 1.53 4.09
Uptime: 02:26:27

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
33219	root	20	0	225M	47752	37252	R	86.3	4.8	0:09.74	ffmpeg -y -f wav -i /tmp/tmpyryp2604p -acodec libvorbis
2626	root	20	0	9132	1964	856	R	0.7	0.2	1:01.97	htop
25558	root	20	0	695M	7820	0	S	0.7	0.8	0:02.46	/usr/bin/containerd-shim-runc-v2 -namespace moby -id 69
25615	systemd-c	20	0	1736M	19864	6104	S	0.7	2.0	0:28.45	/usr/local/lib/erlang/erts-13.1.1/bin/beam.smp -W w -MB
1	root	20	0	98296	4668	3320	S	0.0	0.5	0:01.33	/sbin/init
183	root	20	0	56608	7424	7000	S	0.0	0.7	0:01.10	/lib/systemd/systemd-journald
227	root	20	0	21460	1000	728	S	0.0	0.1	0:00.17	/lib/systemd/systemd-udev
299	systemd-t	20	0	88440	412	304	S	0.0	0.0	0:00.09	/lib/systemd/systemd-timesyncd
331	systemd-t	20	0	88440	412	304	S	0.0	0.0	0:00.00	/lib/systemd/systemd-timesyncd



Thread Group

Name:

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: ☐ Infinite

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

200 (maquina virtual reiniciada)

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename: C:\Users\JorgeS\Downloads\Maestria Los Andes\Nube\PruebasCargaEntrega\Agregate-200-100.txt

Log/Display Only: ☐ Errors ☐ Successes

Label ↑	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
HTTP Request convert to ogg	200	1641	1652	2776	3024	3513	231	3613	0.00%	2.0/sec	0.38	32025.14
HTTP Request convert to wav	200	1642	1666	2848	3126	3253	228	4103	0.00%	2.0/sec	0.38	32178.92
TOTAL	400	1641	1664	2806	3064	3513	228	4103	0.00%	4.0/sec	0.76	63833.19

```
jorges@audio-converter-server: ~  
  
CPU[|||||100.0%] Tasks: 100, 105 thr; 1 running  
Mem[|||||558M/977M] Load average: 0.98 2.38 2.33  
Swp[|||||316M/975M] Uptime: 07:03:30  
  
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command  
36589 root 20 0 225M 47320 37248 R 100.0 4.7 0:06.84 ffmpeg -y -f wav -i /tmp/tmpnkd5j6gl -acodec libvorbis  
28214 root 20 0 9120 2244 1076 R 0.7 0.2 0:41.67 htop  
35214 systemd-c 20 0 1735M 76844 52892 S 0.7 7.7 0:07.64 /usr/local/lib/erlang/erts-13.1.1/bin/beam.smp -W w -MB  
1 root 20 0 99452 2196 756 S 0.0 0.2 0:01.93 /sbin/init  
183 root 20 0 48336 3140 2720 S 0.0 0.3 0:02.43 /lib/systemd/systemd-journald  
202 root 20 0 21592 1004 676 S 0.0 0.1 0:00.19 /lib/systemd/systemd-udev  
236 systemd-t 20 0 88440 108 0 S 0.0 0.0 0:00.12 /lib/systemd/systemd-timesyncd  
263 systemd-t 20 0 88440 108 0 S 0.0 0.0 0:00.00 /lib/systemd/systemd-timesyncd  
276 root 20 0 6748 692 588 S 0.0 0.1 0:00.09 /usr/sbin/cron -f  
277 messagebu 20 0 8280 592 244 S 0.0 0.1 0:00.24 /usr/bin/dbus-daemon --system --address=systemd: --nofo  
280 root 20 0 22236 1388 1036 S 0.0 0.1 0:00.22 /lib/systemd/systemd-logind  
281 root 20 0 14620 8 0 S 0.0 0.0 0:00.28 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant  
309 root 20 0 7168 348 348 S 0.0 0.0 0:00.05 /bin/login -p --  
314 root 20 0 1180M 19484 2812 S 0.0 1.9 0:14.95 /usr/bin/containerd
```

Graph Results

Name:

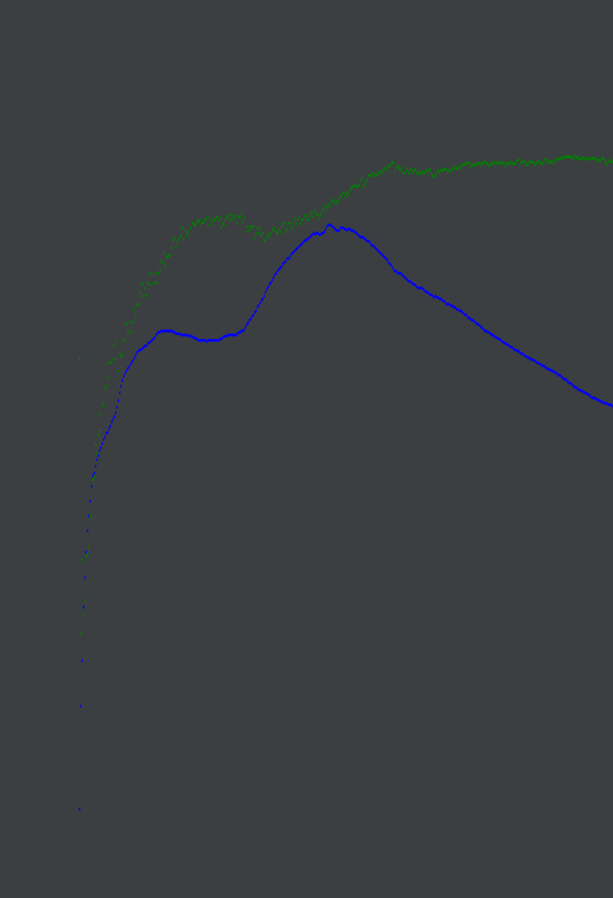
Comments:

Write results to file / Read from file

Filename

Graphs to Display ☐ Data ☒ Average ☐ Median ☐ Deviation ☒ Throughput

3047 ms



0 ms

No of Samples 400
Deviation 767

Latest Sample 824
Throughput 237.154/minute

Average 1641
Median 1664

Análisis de resultados

Vale la pena resaltar como primera medida que el consumo de CPU, desde la primera prueba registrada, fue de 100% casi desde el principio y esto se mantiene mientras exista una conversión en progreso, esto sin embargo no se manifestó en la máquina generando errores o viéndose bloqueada, el uso de memoria aunque no se ve reflejado en las capturas fue variado aunque también me mantenía con frecuencia en 100%

Finalmente se estableció un límite de 200 usuarios tras lo cual el tiempo de las peticiones excede el límite de 1.5 segundos con una máquina virtual reiniciada, estos resultados fueron consistentes tras repetirlos un par de veces

Resultados Escenario 2

Para estas pruebas ya se tenía un mayor conocimiento de JMeter y del comportamiento de la maquina virtual, con esto presente se realizaron algunas pruebas exploratorias aunque mucho mas rápidas en las cuales se utilizaron 250 users, un ramp-up de 500 segundos y un loop count de 4, con lo que se evidencio que había una tendencia de subida lenta pero constante, después de lo cual se utilizo una configuración con el doble de usuarios y de segundos en el ramp-up y se obtuvieron los resultados abajo presentados

```

CPU[|||||||||||||||||||||||||||||||||100.0%] Tasks: 92, 111 thr; 1 running
Mem[|||||||||||||||||||||||||||||||556M/977M] Load average: 0.87 1.53 4.09
Swp[|||||||||||||||||||||] 425M/975M Uptime: 02:26:27

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
33219	root	20	0	225M	47752	37252	R	86.3	4.8	0:09.74	ffmpeg -y -f wav -i /tmp/tmpypnp2604p -acodec libvorbis
2626	root	20	0	9132	1964	856	R	0.7	0.2	1:01.97	htop
25558	root	20	0	695M	7820	0	S	0.7	0.8	0:02.46	/usr/bin/containerd-shim-runc-v2 -namespace moby -id 69
25615	systemd-c	20	0	1736M	19864	6104	S	0.7	2.0	0:28.45	/usr/local/lib/erlang/erts-13.1.1/bin/beam.smp -W w -MB
1	root	20	0	98296	4668	3320	S	0.0	0.5	0:01.33	/sbin/init
183	root	20	0	56608	7424	7000	S	0.0	0.7	0:01.10	/lib/systemd/systemd-journald
227	root	20	0	21460	1000	728	S	0.0	0.1	0:00.17	/lib/systemd/systemd-udev
299	systemd-t	20	0	88440	412	304	S	0.0	0.0	0:00.09	/lib/systemd/systemd-timesyncd
331	systemd-t	20	0	88440	412	304	S	0.0	0.0	0:00.00	/lib/systemd/systemd-timesyncd

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename C:\Users\JorgeS\Downloads\Maestria Los Andes\Nube\PruebasCargaEntrega 1\Graph Results

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Configure

Graphs to Display ☐ Data ☒ Average ☐ Median ☐ Deviation ☒ Throughput

10644 ms

0 ms

No of Samples 2786

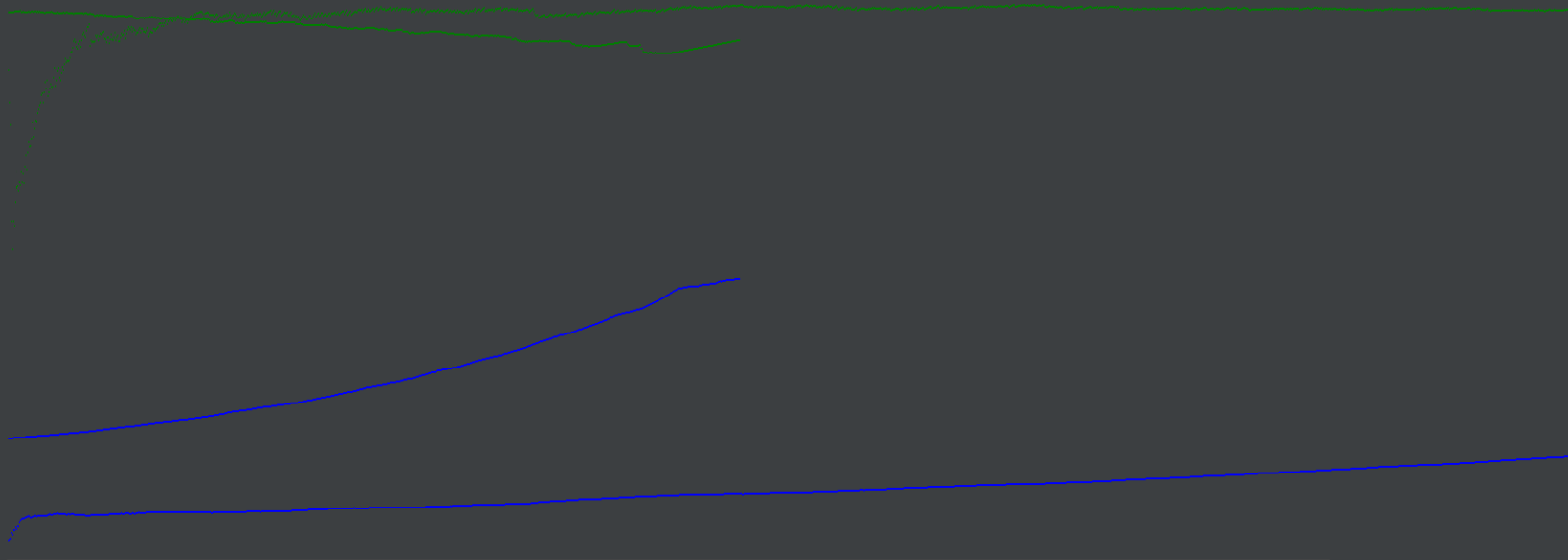
Latest Sample 3607

Average 8664

Deviation 5649

Throughput 218500/minute

Median 2492



```
flask_db=# select * from "task" where status = 'processed' order by "secondsTakedToStartConversion" desc;
```

id	fileName	newFormat	timeStamp	status	conversionTimeStamp	secondsTakedToStartConversion	id_usuario
68	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:38	processed	2022-10-23 22:14:12	634	1
67	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:39	processed	2022-10-23 22:13:34	595	1
66	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:39	processed	2022-10-23 22:13:22	583	1
65	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:38	processed	2022-10-23 22:13:08	570	1
64	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:38	processed	2022-10-23 22:12:54	556	1
63	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:38	processed	2022-10-23 22:12:45	547	1
62	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:37	processed	2022-10-23 22:12:30	533	1
61	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:37	processed	2022-10-23 22:12:13	516	1
60	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:37	processed	2022-10-23 22:11:59	502	1
59	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:37	processed	2022-10-23 22:11:51	494	1
58	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:36	processed	2022-10-23 22:11:43	487	1
57	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:37	processed	2022-10-23 22:11:36	479	1
56	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:36	processed	2022-10-23 22:11:24	468	1
55	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:36	processed	2022-10-23 22:11:15	459	1
54	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:35	processed	2022-10-23 22:11:00	445	1
53	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:35	processed	2022-10-23 22:10:44	429	1
52	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:35	processed	2022-10-23 22:10:30	415	1
49	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:35	processed	2022-10-23 22:10:23	408	1
51	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:34	processed	2022-10-23 22:10:17	403	1
50	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:35	processed	2022-10-23 22:10:10	395	1
48	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:34	processed	2022-10-23 22:09:58	384	1
47	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:34	processed	2022-10-23 22:09:20	346	1
46	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:34	processed	2022-10-23 22:09:14	340	1
45	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:33	processed	2022-10-23 22:09:09	336	1
44	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:33	processed	2022-10-23 22:08:42	309	1
43	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:33	processed	2022-10-23 22:08:14	281	1
42	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:32	processed	2022-10-23 22:08:00	268	1
41	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:33	processed	2022-10-23 22:07:54	261	1
40	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:31	processed	2022-10-23 22:07:45	254	1
39	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:32	processed	2022-10-23 22:07:40	248	1
38	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:31	processed	2022-10-23 22:07:36	245	1
37	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:31	processed	2022-10-23 22:07:31	240	1
36	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:31	processed	2022-10-23 22:07:27	236	1
35	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:31	processed	2022-10-23 22:07:18	227	1
34	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:31	processed	2022-10-23 22:07:13	222	1
33	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:30	processed	2022-10-23 22:07:00	210	1
31	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:30	processed	2022-10-23 22:06:54	204	1
30	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:29	processed	2022-10-23 22:06:43	194	1
29	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:29	processed	2022-10-23 22:06:40	191	1
28	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:29	processed	2022-10-23 22:06:36	187	1
26	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:29	processed	2022-10-23 22:06:32	183	1
27	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:28	processed	2022-10-23 22:06:24	176	1
25	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:29	processed	2022-10-23 22:06:13	164	1
24	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:28	processed	2022-10-23 22:06:08	160	1
23	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:27	processed	2022-10-23 22:06:05	158	1
22	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:27	processed	2022-10-23 22:05:52	145	1
21	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:27	processed	2022-10-23 22:05:31	124	1
20	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:27	processed	2022-10-23 22:05:19	112	1
19	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:27	processed	2022-10-23 22:05:15	108	1
18	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:26	processed	2022-10-23 22:05:10	104	1
17	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:26	processed	2022-10-23 22:05:07	101	1
16	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:26	processed	2022-10-23 22:04:58	92	1
15	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:25	processed	2022-10-23 22:04:45	80	1
14	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:25	processed	2022-10-23 22:04:42	77	1
13	/tmp/audios/flor.mp3	ogg	2022-10-23 22:03:24	processed	2022-10-23 22:04:34	70	1
12	/tmp/audios/flor.mp3	wav	2022-10-23 22:03:25	processed	2022-10-23 22:04:31	66	1

```
--More--
```


Análisis de resultados

Se resalta en la maquina nuevamente un uso persistente de 100% en CPU y 100% en memoria, y en los tiempos de respuesta, si bien no forman parte del interés principal de esta prueba, alcanzaron un tiempo de respuesta promedio de 4684 milisegundos, al finalizar la prueba que se detuvo al alcanzar el primer registro con un tiempo de demora en entrar a procesamiento de mas de 600 segundos y se dio al tener 68 archivos 13 minutos después de haber iniciado la prueba, la tabla presentada en la pagina anterior esta evidentemente mostrando archivos que fueron ya procesados, aunque el tiempo registrado en la columna secondsTakedToStartConversion corresponde con el dato de interés y se vuelve por ende la referencia para futuras comparaciones