Infraestructura para Big Data
Computación en la nube

Daniel Pérez Efremova
Nicolás Argota Lemme

Ejercicios 1 y 2.

- Acceda a https://cloudpingtest.com/. Identifique el centro de datos mejor conectado con su localización. Identifique que centros de datos podrían proveernos servicios de tipo VoIP y cuales no. Identifique centros de datos localizados de manera similar y compare la latencia a nuestra localización. ¿Hay algún CSP "mejor" para nosotros en términos generales?

El mejor proveedor es Vultr, con un tiempo de ping medio de 17 ms.

En general, se observa que los servicios cloud con menor tiempo ping son Vultr y Google con un ping promedio de entre 17 y 57 ms. Por este motivo, para desplegar un servicio VoIP (Voice over Internet Protocol) son la mejor opción, ya que este tipo de servicios precisan baja latencia para que las llamadas sean estables y funcionales.

En la tabla se muestra el top 10 de proveedores ordenados por ping medio de manera ascendente.

Provider	Region	Region Code	Mean (ms)	Std (ms)
Vultr	Spain (Madrid)	mad-es	17	2.42
Vultr	France (Paris)	par-fr	36	1.47
Vultr	United Kingdom (London)	lon-gb	42	5.54
GCP	Belgium (St. Ghislain)	europe-west1	46	3.19
GCP	Germany (Frankfurt)	europe-west3	49	20.26
GCP	UK (London)	europe-west2	50	32.61
Vultr	Netherlands (Amsterdam)	ams-nl	50	3.13
Vultr	Germany (Frankfurt)	fra-de	50	6.05
GCP	Netherlands (Eemshaven)	europe-west4	55	44.12
GCP	Switzerland (Zürich)	europe-west6	57	35.03

Se observa que la región con menor ping es Madrid, algo razonable ya que se estan haciendo los ping desde Madrid y a medida que el centro de computación cloud se aleja de Madrid, la latencia aumenta.

- Compare las medidas en términos absolutos con latencias contra medidas de una web a su elección que esté en las distintas localizaciones medidas

Se realizan experimentos en distintas webs con servidor ubicado en las Regiones de la tabla anterior. Los resultados se muestran en la tabla siguiente.

Web	IP	Region	Mean (ms)	Median (ms)	Std (ms)
www.fotocasa.es	52.84.66.67	Madrid	10.79	5.47	13.33
www.paris-web.fr	92.243.17.46	Paris	72.22	50.5	51.48
www.imperial.ac.uk	146.179.42.148	London	32.4	30.6	5.54
www.keytradebank.be	93.191.218.12	Belgium	60.78	55.8	10.2
www.giz.de	193.97.170.57	Frankfurt	51.5	46	20.26
www.government.nl	178.22.85.8	Amsterdam	42.2	30.35	29
web.unican.es	193.144.193.60	Cantabria	32.15	25.4	21.48

El menor ping se obtiene para la región de Madrid y el resto de resultados son consistentes con la tabla del primer apartado, es decir, a medida que la región se aleja de Madrid, los tiempos de respuesta aumentan.

- Compare, ahora, la varianza de las muestras que nos da cloudpingtest con la varianza de medidas tomadas de la web de Unican

En la última fila de la tabla del apartado anterior se observa que la varianza es aproximadamente 441 ms, una variana muy superior a la del top 5 de medidas tomadas en el primer apartado y comparable a la de GCP de Frankfurt.

Ejercicio 3.

1. Calcular costes de un despliegue de 100 VMs durante 1 mes de 31 días en un centro de datos por especificar (requisitos mínimos):
- **Con capacidad 8 x ECU (2 GHz Xeon) + 8 GB RAM durante 12 horas al día**
- **Con capacidad 16 x ECU (2 GHz Xeon) + 16 GB RAM durante 10 horas al día**
- **Con capacidad marginal + 1 GB RAM durante 2 horas al día**
- **Con capacidad 64 x ECU (2 GHz Xeon) + 64 GB RAM durante las 24 horas del día 31**

2. Sin necesidad de disco duro local pero sí 50% de equipos con IP pública
3. Sí almacenamiento bulk, 25 TB nuevos cada día (agregados, esto es no se sobrescriben/liberan hasta final de mes)
4. Se genera un tráfico ascendente a clientes de 100 GB diarios. Destinos:
- **70% Sudamérica**
- **20% Europa**
- **10% Sudeste asiático (no considere los descuentos por volumen/tráfico interno EC2 o primeros GB gratis)**

5. Parametrice cualquier otro dato que necesite

Calcule resultados para un centro de datos en Europa, EEUU, Sudamérica, Sudeste asiático y Australia. ¿Qué opción recomienda?

Este ejercicio se ha realizado en AWS bajo las siguientes consideraciones en cada punto propuesto:

1. Dado que se especifican franjas horarias de uso en las instancias EC2, el presupuesto se calcula usando el plan de gasto On-Demand.

2. Las instancias EC2 tienen asignadas una IP pública dinámica de manera gratuita. Los cargos se producen si se levanta un servicio Elastic IP (estática) y no se le asocia una instancia (0.005$ hora). El servicio Elastic IP es útil en caso de fallo de una instancia EC2 que haga de host de una aplicación. En el momento de fallo, la IP se mapea a otra instancia EC2 que esté activa y comenzará a hacer de Host de la aplicación sin detenerse el servicio a los usuarios. El coste es 0.10$ por remapeo tras los 100 primeros remapeaeos mensualmente. Como no se especifican fallos, se asume que serán lo suficientemente estables como para no incurrir en gastos adicionales al plan gratuito y no se incluye en el presupuesto.
NO APLICA CONSUMO

3. El almacenamiento *bulk* requiere transacciones constantes de datos en forma de ficheros que sean accesibles por el usuario en cualquier momento mediante descarga. Por esto, para el almacenamiento se ha seleccionado el servicio S3 (Simple Storage Servcice), ya que las EC2 no necesariamente se levantan con almacenamiento local. Como no se especifica el número de interacciones PUT, COPY, LIST, GET o SELECT sobre S3, no se especifican estos parámetros en el presupuesto del servicio S3. Asíque en caso de superarse los límites gratuitos, puede incurrirse en gastos adicionales.

ALMACENAMIENTO TOTAL: 25 TB x 31 días = 775 TB mes

4. Se asume que las trasferencias de datos van a ser entre centros de AWS y no por la red de Internet. Además, se asume que dentro de una misma zona geográfica, el tráfico se dirige enteramente a una única región de AWS. Por ejemplo, en el caso de que levantásemos el servicio en Europa, se asume que el tráfico de toda Europa se dirige enteramente a la misma región de AWS. Por otro lado, por "tráfico ascendente" se entiende que es saliente desde las instancias EC2 (outbound) ya que el entrante (inbound) actualmente es gratuito. En la siguiente tabla se detalla el tráfico sujeto a cargos según la zona geofráfica *host* y la distribución de tráfico propuesto por las zonas geográficas.

Region Host	Tráfico Europa (GB)	Tráfico Sudeste Asiático (GB)	Tráfico Sud América (GB)	Total
Europa	0	3100	21700	24800
EEUU	6200	3100	21700	31000
Sudamérica	6200	3100	0	9300
Sudeste Asiático	6200	0	21700	27900
Australia	6200	3100	21700	31000

A continuación se muestra una tabla comparativa de los precios estimados del servicio propuesto siguiendo las consideraciones antes expuestas para todos los casos posibles ordenados por precio estimado[1].

Region	Latencia Region (ms)	Uso (hr/día)	ECU	RAM (GB)	Tipo Almacenamiento	Capacidad Almac. (TB mes)	Tráfico Cargable (GB)	Precio Estimado ($ Mes)
EEUU (N. Virginia)	429	2	0	1	S3	775	31000	17362
EU (paris)	208	2	0	1	S3	775	24800	18012
Singapour	697	2	0	1	S3	775	21700	20418
EEUU (N. Virginia)	429	12	8	8	S3	775	31000	20815
Sydney	792	2	0	1	S3	775	31000	21652
EU (paris)	208	12	8	8	S3	775	24800	22043
EEUU (N. Virginia)	429	10	16	16	S3	775	31000	23167
Singapour	697	12	8	8	S3	775	21700	24733
EU (paris)	208	10	16	16	S3	775	24800	24787
Sydney	792	12	8	8	S3	775	31000	25645
Singapour	697	10	16	16	S3	775	21700	27673
Sydney	792	10	16	16	S3	775	31000	28585
Sao Paulo	710	2	0	1	S3	775	9300	31154
Sao Paulo	710	12	8	8	S3	775	9300	36384
Sao Paulo	710	10	16	16	S3	775	9300	40133
EEUU (N. Virginia)	429	24	64	64	S3	775	31000	73375

1 **Estimador de precios de AWS:** https://calculator.aws/#/estimate

EU (paris)	208	24	64	64	S3	775	24800	83363
Singapour	697	24	64	64	S3	775	21700	90433
Sydney	792	24	64	64	S3	775	31000	91345
Sao Paulo	710	24	64	64	S3	775	9300	120152

Para elegir el servicio se evalúan 2 ejes: precio y capacidad de cómputo (ECU y RAM).

Los tres primeros resultados pertenecen a las EC2 con ECU marginal y 1 GB de RAM. Si bien tienen un precio bajo, su capacidad de cómputo es muy limitada por lo que tienen gran riesgo de fallo por sobrecarga.

La opción más asequible en precio y capacidad de cómputo es N.Virginia, aunque en precio es comparable con Paris. Para decidir entre N. Virginia y Paris, se puede incorporar como eje de decisión la latencia (medida respecto a Madrid). En caso de que sea relevante, se elegiría Paris y su diferencia en precio quedaría justificada por un mayor rendimiento (la mitad de latencia medida desde Madrid).

Ejercicio 4

Asuma que se necesita una latencia máxima de 100 ms:
- ¿Cambiaría esto la decisión del ejercicio 3? (Puede asumir que las 100 VMs no tienen que estar en el mismo centro datos)
- ¿Qué valor de latencia lo haría?

Dadas las especificaciones (70% de usuarios de Sud América) tiene sentido basar la infraestructura en San Paolo por criterios de rendimiento (latencia) y presupuesto, y dejar algunas EC2 en Europa y Sudeste asiático. Quizá siguiendo el mismo criterio de reparto (70-20-10) de clientes, es decir, 70 EC2 en Sudamérica (San Paolo), 20 EC2 en Europa (Irlanda) y 10 en Sudeste asiático (Seoul o Sinpagur).

Ejercicio 5. (Mismo proceso anterior en GCP y Azure.)

Se ha realizado sin documentar obteniendose conclusiones muy similares a las anteriores.

Ejercicio 6.

Compare y discuta qué es más económico si el escalado horizontal o vertical:
- En concreto en EC2, calcule para distintos escenarios de carga (esto es, equipos poco potentes contra equipos muy potentes).

En el siguiente gráfico se muestra una comparativa del $/CPU para instancias EC2 en la región de North Virginia. En el experimento, se escala al doble de capacidad de CPU de dos maneras:

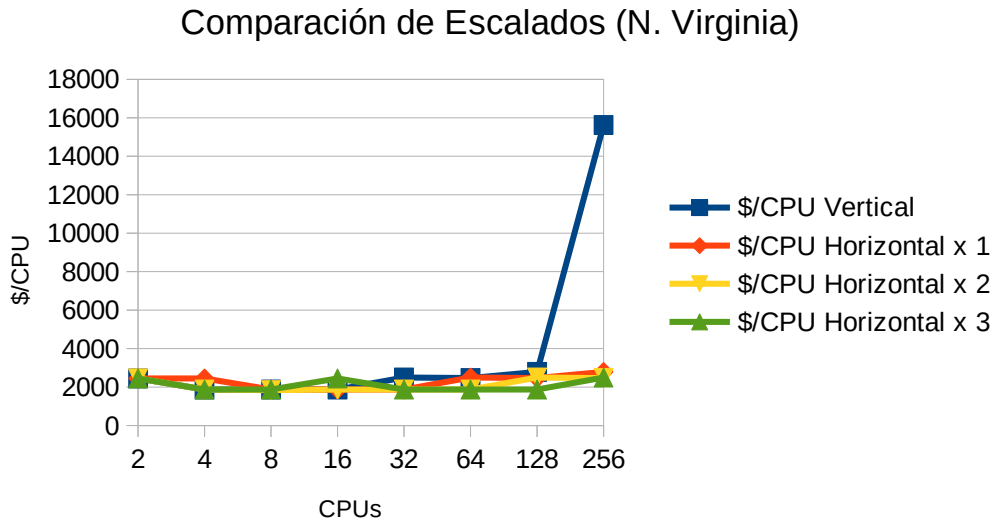
1. Vertical: Escalando al doble de CPUs en una misma instancia
2. Horizontal: Incorporando instancias adicionales para tener máquinas iguales que sumen la misma capacidad de computo (CPUs) que verticalmente.

Los ratios se han calculado de la siguiente forma:

$/CPU Vertical = Precio de la instancia / n.º de CPUs
$/CPU Horizontal x i =

= n.º de instancias del nivel i-ésimo anterior * Precio de la instancia del nivel i-ésimo anterior / n.º de CPUs

(Nótese que $/CPU Horizontal x i solo es comparable al escalado vertical hasta el número i de CPUs, es decir, las 4 propuestas solo son comparables a partir de 16 CPUs)



Se observa que:
- para problemas que requieran de **pocas prestaciones** (menos de 8 CPUs) el escalado vertical es más eficiente y además más sencillo de mantener.
- para problemas que requieran de **prestaciones moderadas** (entre 8 y 64 CPUs) ambos escalados son casi idénticos desde el punto de vista de costes, aunque es probable que si se escala de manera horizontal con muchas máquinas, aparezcan costes de comunicación entre máquinas, software de gestión, mantenimiento, etc que hagan al escalado vertical más asequible.
- para problemas que requieran **altas prestaciones** (más de 64 CPUs), el escalado vertical es poco eficiente. Además, la eficiencia es mayor cuantas más máquinas se levanten para sumar la capacidad de CPU de una única máquina potente.

En general, se observa que el escalado horizontal tiene una curva $/CPU suave en comparación con el escalado vertical. Más suave cuanto mayor es el número de máquinas levantadas.

Ejercicios 8 y 9

En este ejercicio se han sufrido problemas técnicos debido a un cambio en las políticas de seguridad de Git. No es posible realizar la instalación de OpenStack[2].

La incidencia consiste en que se recomienda realizar la instalación con un usuario que es propietario de todos los ficheros de instalación. Sin embargo, Git exige ser propietario de todos los subdirectorios a los que se quiera acceder durante el clonado de un repositorio. Es decir, que sea el usuario root el que llame a la descarga e instalación de contenidos. Esto genera un conflicto de permisos que impide la correcta instalación del servicio. **El bug se reportó el 12 de Abril de 2022.**

Se dejan aquí los scripts probados sobre una máquina virtual Ubuntu 20.04.

2 https://bugs.launchpad.net/devstack/+bug/1968798

```
echo ---------------------------------------------- >> test_cpu.txt
echo $(date) >> test_cpu.txt
echo ---------------------------------------------- >> test_cpu.txt
sysbench --threads=3 cpu run >>test_cpu.txt
```

Cuya salida tiene la forma:

```
----------------------------------------------
sáb 16 abr 2022 12:30:31 CEST
----------------------------------------------
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 3
Initializing random number generator from current time


Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  3496.60

General statistics:
    total time:                          10.0005s
    total number of events:              34972

Latency (ms):
         min:                                    0.82
         avg:                                    0.86
         max:                                    3.56
         95th percentile:                        1.01
         sum:                                29992.61

Threads fairness:
    events (avg/stddev):           11657.3333/3.40
    execution time (avg/stddev):   9.9975/0.00
```

En este script debería observarse que al aumentar el número de eventos por segundo es mayor ya que se tiene mayor capacidad de cálculo.

El trabajo de cron que se hubiera implementado se deja acontinuación. Cada minuto se corre el test test_cpu.sh.

```
* * * * * /scripts/test_cpu.sh
```

ANEXOS

- aws_budget.ods: Hoja de trabajo LibreOffice Calc usada para calcular los presupuestos de cada máquina en cada zona. Contiene además el cálculo de tráfico cargable según la posición del centro de Datos y el cálculo de los ratios $/CPU.
- ping_cloud_service.ods: Hoja de trabajo LibreOffice Calc usada pararecabar los test de ping a cada centro de datos.
- test_cpu.sh: Script bash para medir el rendimiento de una MV volcando a fichero de texto.
- test_cpu.txt: Ejemplo de salida del script.