El impacto de la latencia en el negocio

Guillaume Laforge,
Developer Advocate en Google Cloud Platform

Guillaume Laforge

Como Developer Advocate en Google Cloud, Guillaume tiene la misión de dar a conocer el amplio conjunto de productos y servicios que ofrece la plataforma a los desarrolladores que buscan aprovechar la nube para sus proyectos y negocios. Este rol lo compagina con su puesto de Chair of the Project Management Committee for the Groovy project at Apache Software Foundation. Previa llegada a Google, Guillaume ha desempeñado distintos cargos en empresas como Restlet, Pivotal o VMware o IBM.

Índice

01
El impacto de la latencia en los negocios

A medida que la economía se vuelve global, invertir en tecnología en la nube ha pasado de ser opcional a convertirse en una necesidad. Tener tus propios centros de datos para dar soporte a todos tus clientes, sin importar dónde estén, no es óptimo por dos razones fundamentales: es demasiado complejo a nivel de infraestructura e implica altos costes asociados (instalación, personal, mantenimiento, etc.). La nube, precisamente, cubre ese *gap* de una manera eficiente, flexible y económicamente asumible.

De acuerdo con un estudio reciente de Gartner, se espera que **el gasto global en servicios *cloud* aumente un 23% en 2021** como consecuencia del cambio de mentalidad que están viviendo las empresas, que están pasando de solo respaldar las herramientas necesarias para el trabajo en remoto, a hacer lo propio con proyectos digitales y estrategias de TI a gran escala.

Es decir, podemos observar que las tecnologías en la nube empiezan a concebirse como un recurso clave para lograr una ventaja de negocio competitiva, ya que permiten rastrear el rendimiento de las operaciones, procesos y herramientas, al tiempo que ayuda a tomar decisiones más inteligentes. No obstante, y a pesar de la infinidad de beneficios que trae consigo, el aumento exponencial del uso de aplicaciones en la nube también acarrea algunos desafíos a los que hay que dar una respuesta eficaz para afrontarlos y solventarlos con éxito.

En concreto hablamos de la creciente demanda de un mayor rendimiento en todos los sistemas de telecomunicaciones para que las tecnologías en la nube puedan funcionar al nivel requerido. La latencia – o tiempo que tarda en transmitirse un paquete de datos de un punto de la red de conexión a otro – se ha convertido en un aspecto crítico para todos los usuarios de aplicaciones en la nube, ya que afecta drásticamente a la experiencia UX, por lo que resulta esencial trabajar para reducirla.

> **Se espera que el gasto global en servicios cloud aumente un 23% en 2021**

UN AUMENTO EN LA LATENCIA REDUCE LAS VENTAS ONLINE

Una alta latencia a la hora de visitar una página web tiene un fuerte impacto negativo: los usuarios cerrarán la página si tarda mucho tiempo en cargarse. Si la web es, además, un espacio de venta de productos o servicios, el resultado puede ser catastrófico: a más latencia, mayor reducción de las ventas online, con las consiguientes pérdidas económicas.

Ya Google recogía en un estudio hace años que un delay de medio segundo causaba una pérdida del 20% en el tráfico desde Google, y un retraso de una décima de segundo podía reducir las ventas de Amazon en un 1%. Si lo comparamos con una investigación más reciente de Akamai, vemos cómo esto sigue pasando años después, e incluso las pérdidas van en un aumento: cada delay de 100 milisegundos en el tiempo de carga de un sitio web puede afectar a las tasas de conversión en un 7%, lo que se traduce en una caída de un 6% en las ventas.

Así, **la latencia es un componente esencial en cualquier interacción con un sitio web o con aplicaciones en la nube** – recordemos que las páginas web y *apps* están alojadas en servidores en la nube que no tienen por qué estar cerca del usuario final. En este sentido, el éxito de la experiencia de usuario dependerá de la latencia: se trata de un factor que se traduce en términos de productividad y beneficios para los negocios.

02
Comprender la latencia con prácticas y herramientas *ad hoc*

Para entender la latencia en las aplicaciones en la nube, es importante saber de dónde viene.

A la hora de garantizar la disponibilidad de las aplicaciones es fundamental establecer y monitorizar las métricas de nivel de servicio, algo que, por ejemplo, desde Google Cloud hacen de forma constante. La suite de herramientas de operaciones de Google Cloud Platform permite a los desarrolladores y operadores monitorizar y observar las aplicaciones en la nube, así como recibir alertas cuando aumenta la latencia. Esto es especialmente importante en un entorno en el que los líderes empresariales planean adoptar soluciones de *edge computing* para reducir este aspecto: de acuerdo con IDC, **9 de cada 10 encuestados considera que el éxito empresarial depende de una baja latencia, y el 75% de las compañías espera menos de 5 milisegundos de latencia gracias a la adopción de esta tecnología.** En este sentido, el ecosistema de casos de uso que están surgiendo entorno al *edge computing* está creciendo exponencialmente, ya que la mejora en la latencia permite desplegar arquitecturas mucho más eficaces y adaptadas a los requisitos de cada negocio. Un buen ejemplo es el caso de IE University, donde el cliente aplica la realidad virtual basada en 5G y *edge computing* para impartir un seminario de arquitectura, y despliega esta tecnología en una central muy cerca del campus, lo que garantiza una latencia mínima. En esencia, estas métricas deben estar estrechamente vinculadas con los objetivos de negocio, de forma que, si la meta es mejorar los servicios y, como consecuencia, la experiencia de usuario, es necesario implementar determinadas prácticas para monitorizar las aplicaciones. En concreto, las prácticas de **SRE** (*Site Reliability Engineering*) alientan a las empresas a implementar técnicas **SLO** (*Service-Level Objective*), **SLA** (*Service-Level Agreement*) y **SLI** (*Service-Level Indicator*), para realizar dicha monitorización. Sin estas herramientas, es imposible saber si el sistema en la nube está disponible, es útil e incluso si es o no fiable. Por ello, si no tienen relación alguna con los objetivos de negocio, faltarán datos sobre si las decisiones que se están tomando o los procesos y las operaciones en marcha están ayudando o perjudicando al negocio.

9/10 encuestados consideran el éxito con la **baja latencia**

75% de las compañías espera **menos de 5 milisegundos** de latencia

En este sentido, para entender la latencia es fundamental ser consciente de qué beneficios aportan los objetivos, acuerdos e indicadores a nivel de servicio:

1. SRE (*Site Reliability Engineering*). Es un conjunto de prácticas y enfoques creados por Google para tratar las operaciones (la administración de servidores, hardware y redes, la verificación de que todo funcione correctamente y todos los servicios estén operativos, etc.) como un problema de software. El objetivo es mantener los sistemas críticos para el negocio (como, por ejemplo, el software de compras online o los sistemas de transacciones financieras) en funcionamiento en todo momento, en las mejores condiciones posibles, con tiempos de respuesta rápidos, baja latencia y suficiente ancho de banda para atender a todos los clientes. Los ingenieros especialistas en SRE definen los SLI, los indicadores de nivel de servicio.

2. SLI (*Service-Level Indicator*). El indicador de nivel de servicio o SLI es una medida directa del comportamiento de un servicio, definido como la frecuencia de sondeos exitosos del sistema en cuestión. Este indicador se puede medir y rastrear a lo largo del tiempo. Por ejemplo, cuántas transacciones se están ejecutando, cuántos recursos están disponibles en el sistema, cuánto tiempo se tarda en completar el pedido de un cliente o cuál es el histograma de los tiempos de respuesta a las solicitudes de los clientes. Cuando se evalúa si el sistema se ha estado ejecutando dentro del parámetro fijado por el SLO, se mira el SLI para obtener el porcentaje de disponibilidad de servicio: si cae por debajo del SLO, significa que el sistema no está funcionando de forma adecuada y hay que encontrar la forma de incrementar su disponibilidad para disminuir la latencia (por ejemplo, ejecutando una segunda instancia del servicio en una ciudad diferente para equilibrar la carga entre ambas).

3. SLO (*Service-Level Objective*). El objetivo de nivel de servicio o SLO es un objetivo numérico preciso que se asocia con la disponibilidad del sistema en la nube en cuestión. Su función es definir si el sistema está funcionando de manera fiable o si necesita algún cambio de diseño y arquitectura.

Para poder decir que el sistema está siendo eficaz, tiene que cumplir en todo momento con este SLO. Por ejemplo, las transacciones de los clientes no deben durar más de X milisegundos (el SLI asociado es la duración de una transacción, en este supuesto). Los SLO se pueden alinear con los objetivos comerciales, porque quizás la empresa sepa que, si las transacciones tardan demasiado, los clientes huirán y se dirigirán a proveedores más competitivos. Es importante tener en cuenta que cuanto más fiable sea el servicio, mayor será su coste de operación. Por ello, un paso crítico es definir el nivel más bajo de fiabilidad que es aceptable para los usuarios de cada servicio y que este sea el SLO. De esta manera será un target alcanzable, a la par que permitirá garantizar que la latencia es lo más baja posible. Por último, pero no menos importante, cuando una empresa trabaja con un proveedor que brinda algún servicio *cloud* (por ejemplo, Telefónica usando Google Cloud Platform), el proveedor puede definir un SLA o Acuerdo de nivel de servicio.

4. SLA (*Service-Level Agreement*). Un acuerdo de nivel de servicio o SLA es un compromiso del proveedor con el cliente que garantiza que los servicios estarán en funcionamiento, por ejemplo, el 99,99% del tiempo (varía en función del acuerdo alcanzado) y que el tiempo para solucionar un problema será inferior a una determinada duración. Se trata de un acuerdo contractual y podría ocurrir que el proveedor tuviera que abonar una penalización al cliente si no se cumple el acuerdo. Las penalizaciones suelen ir desde un reembolso parcial de la tarifa a los clientes hasta un tiempo de suscripción adicional gratuita, por ejemplo.

Google ha desarrollado todas estas prácticas con el objetivo de garantizar que los servicios estén siempre en funcionamiento. Se trata de un conjunto de prácticas que puede ayudar a las empresas a monitorizar los indicadores correctos para rastrear el comportamiento y la salud de los sistemas necesarios para el negocio de la empresa. Y ¿de qué manera estas prácticas pueden ayudar a las compañías a alcanzar sus objetivos de negocio? Por ejemplo, al alinear los SLO con las metas comerciales, el seguimiento de los indicadores subyacentes puede ayudar a conseguir ciertos objetivos, como: el número de transacciones (capacidad para atender picos de tráfico cuando todos los clientes se conecten al mismo tiempo), disponibilidad de un servicio (que un servicio bancario esté disponible todo el tiempo cuando un cliente inicia sesión en la interfaz web de su banco), que el ancho de banda esté disponible para atender el tráfico (por ejemplo, si los vídeos de una plataforma como Netflix se sirven en alta calidad, los clientes pagan por disfrutar contenidos en 4K, y quieren tener acceso a esta calidad siempre).

Analicemos otro ejemplo del sector *e-commerce*. Hablamos de un *retailer* que opera a nivel mundial con una aplicación de compras y sitio web para el consumidor final. Si el sitio web es demasiado lento para responder, es probable que el cliente abandone y termine comprando los artículos que buscaba en otra tienda online. Por lo tanto, reducir el tiempo de respuesta para tener un sitio web rápido, ayudará al cliente a comprar de manera más conveniente. Volviendo a los conceptos de SLI, SLO y SLA aplicados a este ejemplo, definiríamos los siguientes parámetros:

- **SLI:** el tiempo de respuesta de la aplicación web al navegar por artículos en la tienda online
- **SLO:** el 95% de las solicitudes entrantes deben ser atendidas por los servidores en menos de 200ms.

- **SLA:** va a depender de quién sea el responsable del SLA:

Como hablamos de un servicio interno de su aplicación, como una base de datos alojada, en este caso, quizás sea el equipo responsable de esa base de datos el que debe comprometerse a responder las consultas utilizadas por los desarrolladores web en menos de 100 ms.

Asimismo, también puede tratarse de un servicio (por ejemplo, una base de datos Cloud SQL de Google Cloud Platform) donde existe un SLA de Google para tener un "Porcentaje de tiempo de actividad mensual para el cliente de al menos 99,95%". Si no se cumple el acuerdo, Google Cloud tiene que compensar monetariamente al cliente en sus próximas facturas cloud.

Si el *retailer* en cuestión no supervisa el tiempo de respuesta, no notará que aumenta la latencia. Por tanto, lo primero y más importante: hay que monitorizar. Como parte del proceso de monitorización es posible configurar alertas y recibirlas cuando no se alcancen los umbrales de los SLO. Es en ese punto cuando se debe investigar por qué hay una mayor latencia y luego intentar mejorar algún elemento (código, servidor, redes) para mejorar el rendimiento del sistema en general, mediante algoritmos más eficientes, más RAM en la CPU, o proveer un mayor ancho de banda.

03
Prácticas de desarrollo que potencian la agilidad en la empresa

Hay muchos aspectos donde se puede mejorar la velocidad de los procesos, desarrollos, pruebas, aplicaciones, etc. Centrándonos en la latencia de las aplicaciones, elegir marcos de trabajo ligeros es muy positivo para reducir el delay. Además, también es necesario asegurar que el desarrollo y las pruebas son rápidas, ya que esto mejora el ciclo de retroalimentación y permite corregir los errores antes, así como lanzar al mercado nuevas funciones de manera más ágil.

Organizaciones, desarrolladores, líderes tecnológicos, arquitectos, equipos de TI, profesionales de DevOps, etc., desempeñan un rol que ayuda a potenciar la agilidad de toda la empresa, mejorando continuamente los procesos de los que se encargan en aras de optimizar ese ciclo de retroalimentación antes mencionado. Todo ello impactará de lleno en la latencia del servicio en la nube en cuestión.

¿QUÉ PODEMOS HACER PARA REDUCIR LA LATENCIA?

La latencia es visible en diferentes niveles del desarrollo y, como tal, hay distintas prácticas que se pueden implementar para reducirla y que las aplicaciones en la nube tengan un alto nivel de optimización, reduciendo así las posibilidades de una mala experiencia de UX, una baja productividad de los empleados que las utilizan y, en resumidas cuentas, de pérdidas económicas para el negocio.

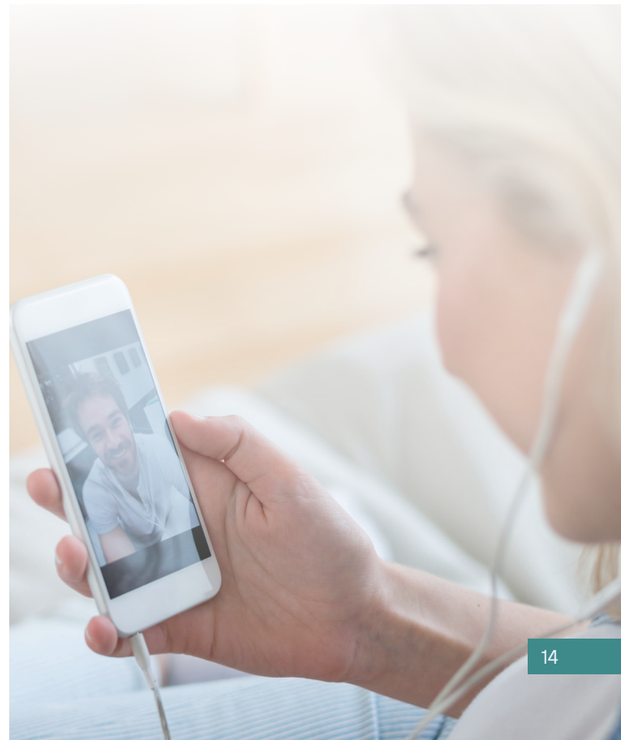
En lo que respecta a las buenas prácticas para reducir la latencia, vamos a centrarnos en tres aspectos críticos que hay que atajar lo antes posible:

1. Una interfaz lenta de la web o móvil. Como hemos visto antes, una interfaz lenta conlleva una bajada en la tasa de conversión y, en resumidas cuentas, en los beneficios finales. Si la página o la app de la competencia está mejor optimizada y carga más rápido, la audiencia potencial va a tender a irse, fruto de la frustración provocada por la percepción de pérdida de tiempo. Por ello, **es vital solucionar los problemas de latencia antes de que la página esté disponible para el público**.

Lo mismo que ocurre con la interfaz web/móvil, ocurre con el resto de aplicaciones o software: si la interfaz no está optimizada y la UX no está lo suficientemente trabajada, puede impactar en los tiempos de carga y, por ende, en la latencia. No importa que se trate de una aplicación para eCommerce, herramientas de trabajo colaborativo, *digital workplace*, software de gestión financiera, programas de gestión de reservas, etc.; todos los problemas de optimización hay que solventarlos antes de que la aplicación está disponible en el mercado.

¿Cómo podemos solucionar este problema?

- **En primer lugar, aplicando técnicas de optimización de SEO.**
- **En segundo lugar, comprobando la velocidad de carga de la web** a través de herramientas como Lighthouse, que se ejecuta en Chrome Devtools o con un módulo de Node para auditar la página en cuestión y generar un informe de estado del que partir para hacer las modificaciones pertinentes.
- **En tercer lugar, utilizando *'service workers'*** – secuencias de comandos que el navegador ejecuta en segundo plano que pueden interceptar y manejar solicitudes de red, incluida la administración programática de un caché de respuestas – para aplicaciones web enriquecidas y PWA (Progressive Web apps), incluyendo el uso de almacenamiento en caché de activos y respuestas.



Si volvemos a nuestro ejemplo de la empresa de *e-commerce*, una situación a la que habrá que hacer frente con frecuencia será una campaña de descuentos o promociones. Cuando haya que mostrar artículos en promoción tal vez se requiera una costosa consulta en la base de datos. Si todas las personas que navegan por la página de inicio del sitio web de compras *online* ven la lista de artículos promocionales, es posible que haya miles de peticiones para la misma consulta ejecutándose en paralelo, cuyo resultado arrojará siempre la misma lista de productos, ya que las promociones no cambian cada segundo. Si hay demasiadas consultas al mismo tiempo, quizás la base de datos alcance sus límites y tarde más en responder a cada consulta. La idea aquí sería introducir algún "almacenamiento en caché" de los datos. **Al usar Cloud Memorystore, es posible almacenar el resultado de esa consulta frecuente de la base de datos en la memoria y entregar el resultado rápidamente, y solo actualizar el resultado guardado en caché una vez cada minuto o cada hora.** Esto dará como resultado una sola consulta de base de datos cada minuto u hora, en lugar de miles de consultas de los miles de visitantes del sitio web.

2. Peticiones/respuestas de red que requieren mucho tiempo. Si la página web o aplicación tarda mucho en responder a la petición que se le envía – por ejemplo, cambiando de un dashboard a otro o tardando mucho en actualizar el carrito de la compra cuando se añade un nuevo item – lo más probable es que sea porque la región de la nube que se está utilizando sea demasiado lejana a los usuarios finales o porque no se estén gestionando bien los picos de tráfico. Sea como fuere, se trata de una situación crítica que puede suponer una tasa de abandono muy alto, por lo que hay que atajar el problema de latencia de raíz y lo antes posible.

¿Cómo podemos solucionar este problema?

- Lo más importante es **seleccionar una región *cloud* local**, para que sea mucho más cercana a los usuarios finales y el tráfico en el envío y recepción de información recorra el menor recorrido posible. Con esto garantizaremos mayor agilidad en el tiempo de respuesta.

- **Empleando máquinas virtuales o instancias que tengan una mayor banda ancha.** De esta manera, se podrá asignar más recursos a las aplicaciones o servidores en la nube si la latencia es muy alta, ayudando a reducirla.

- **Utilizando productos escalables sin servidor o *'serverless'*** (como Cloud Run o App Engine) para manejar los picos de tráfico. Con ellos podemos escalar hacia arriba y, posteriormente, hacia abajo según necesidades de tráfico. Las capacidades de escalado automático de la computación '*serverless*' permiten que, sin importar el tráfico que haya (los picos de usuarios en un momento particular del día), todo vaya a funcionar correctamente en la medida en que se desarrollen los servicios siguiendo buenos principios de desarrollo nativo en la nube, como la creación de servicios sin estado o '*stateless services*'. Poniendo un ejemplo práctico, cuando hay un pico de tráfico en una web que se ha mencionado en la televisión en horario *prime-time*, la infraestructura detrás de este tipo de tecnología escalará automáticamente, añadiendo tantas instancias de servidor como se necesiten para dar respuesta a la demanda creciente, sin que esto suponga un impacto negativo en los tiempos de respuesta.

En este sentido, una empresa puede implementar varias versiones de una aplicación para diferentes regiones alrededor del mundo. Existen diversos tutoriales sobre cómo implementar contenedores (que contienen dicha aplicación empresarial) en Google Cloud Run en varias regiones. La idea central es desplegar el mismo contenedor en cada región donde se desee tener disponible una aplicación, lo más cerca de sus usuarios. Posteriormente, será necesario configurar cosas como el equilibrio de carga, las entradas de DNS, etc., de modo que cuando una solicitud llegue al servidor de nombres de dominio, se envíe al contenedor más cercano a sus usuarios. Esto agrega cierta complejidad a las operaciones, implementación, monitorización, etc., pero la latencia se reducirá en gran medida: el tiempo en la red será menor, lo que implicará tener mejores tiempos de respuesta.

Por ejemplo, el tiempo que tarda un navegador web en acceder a un servicio de Google Cloud Run que se implementa en todas las regiones donde existe Cloud Run será diferente dependiendo de dónde esté el usuario. Para alguien que quiera acceder desde París, una consulta a Cloud Run alojado en Londres tarda 40ms. Sin embargo, para todas las regiones en Asia, el tiempo de la red es siempre mayor que 300ms. Por lo tanto, si una aplicación de compras en línea está alojada en Mumbai, siempre habrá una latencia de red adicional de 260ms más que si la aplicación estuviera alojada en una región de Cloud Run en Europa, de modo que la navegación a través de este sitio web será más lenta y menos funcional.

3. Aplicación ‘backend’ o API que necesita tiempo para responder o estar lista para atender a las solicitudes entrantes de los usuarios. Este punto es, quizá, el que suele pasar más desapercibido puesto que se trata de aplicaciones o APIs que trabajan en un segundo plano, pero es igual de importante que los anteriores si lo que queremos es reducir al mínimo la latencia y, por ende, dar una respuesta óptima y ágil a las necesidades de los usuarios y desarrolladores.

¿Cómo podemos solucionar este problema?

Al utilizar herramientas ‘*serverless*’ como Cloud Run, podemos evitar lo que se conocen como ‘*cold starts*’ a través de instancias mínimas.

Una de las mejores características de la tecnología sin servidor es su modelo operativo de pago por uso que permite escalar un servicio a 0, dejándolo inoperativo. No obstante, en algunas aplicaciones, esto genera un incremento en la latencia a la hora de procesar la primera solicitud cuando la aplicación se reactiva de nuevo. Este llamado ‘startup tax’ o “impuesto inicial” es novedoso para la tecnología ‘serverless’, ya que, como su propio nombre indica, no hay servidores en ejecución si una aplicación no recibe tráfico.

Cloud Run se encarga de crear un endpoint HTTP, enrutar las solicitudes a los contenedores y escalar los contenedores hacia arriba y hacia abajo para manejar el volumen de solicitudes y reducir la latencia del tiempo de respuesta. Con la inclusión de las instancias mínimas se puede mejorar drásticamente el rendimiento de las aplicaciones, ya que se puede configurar determinadas instancias para que estén en standby y listas para ayudar con el tráfico tan pronto se detecten las solicitudes necesarias para ello. Así, se evitan los ‘cold starts’ y se disminuye la latencia.

04 Implementar más cerca de los usuarios finales

Para reducir la latencia de la red entre los usuarios finales y la aplicación en la nube, el mejor enfoque es implementar dicha aplicación en la región más cercana a sus usuarios. Esto hace que la información viaje menos distancia, con lo que las posibilidades de una alta latencia se reducen considerablemente.

En este sentido, para poder ofrecer una menor latencia y un mayor rendimiento a los usuarios, Google y Telefónica han llegado a un acuerdo para abrir una región de Google Cloud en Madrid. Dicha región utilizará la infraestructura de la compañía de telecomunicaciones para ofrecer a los clientes de Google Cloud que operan en España un servicio eficiente y de gran calidad, donde los flujos de trabajo y el acceso a los datos alojados en la nube sea ágil y sin fricciones.

Como el tendido de nuevos cables no siempre es posible debido a las limitaciones ya explicadas, además de por los costes asociados y otras posibles razones (medioambientales, por ejemplo), la nube adquiere un papel imprescindible a la hora de subsanar estos gaps. No obstante, para que la latencia no se incremente considerablemente, el problema de fondo que hay que solucionar es el mismo: hay que **diseñar y optimizar los protocolos y códigos de red teniendo en mente las limitaciones del ancho de banda disponible, así como reducir los viajes de ida y vuelta de la información y, sobre todo, acercar los datos al usuario final.**

¿Cómo conseguimos, precisamente, implementar más cerca? La clave está en ampliar la infraestructura que gestiona nuestra nube y que ésta esté presente en distintos puntos del mapa. Debido al elevado coste asociado que esto supone, aquí es donde entra en juego la computación 'serverless', así como las distintas plataformas asociadas a esta tecnología.

La clave de la computación serverless tiene que ver con la agilidad que brinda a los desarrolladores encargados de las aplicaciones. Es más sencillo utilizar servicios gestionados serverless en la nube que operar, aprovisionar y mantener servidores físicos o máquinas virtuales. Si bien es posible desarrollar aplicaciones de baja latencia tanto con computación serverless, como con máquinas virtuales y servidores físicos, resulta mucho más conveniente hacerlo con servicios serverless (como Cloud Run, App Engine o Cloud Functions). Por lo tanto, potencialmente, con picos de tráfico más altos, una aplicación alojada en un servicio serverless puede tener una latencia más baja en comparación con una alojada en una arquitectura clásica.

A la hora de implementar varias versiones de una aplicación para diferentes áreas del mundo, la herramienta Google Cloud Load Balancer tiene múltiples beneficios que hacen que implementar más cerca de los usuarios finales sea mucho más sencillo, pues permite enrutar las solicitudes de los usuarios a la región más cercana que aloja la aplicación cloud en cuestión. Cloud Load Balancer es capaz de descubrir cuál es la mejor manera de atender una solicitud entrante para enviarla al servicio de Cloud Run más conveniente, en la región más cercana al usuario final que realiza la petición desde su navegador web.

Sumando la integración HTTP(S) Load Balancing, la oferta 'serveless' de aplicaciones como APP Engine (estándar y flexible), Cloud Functions y Cloud Run puede utilizar las mismas capacidades de balanceo de carga HTTP(S) de nivel empresarial que el resto de Google Cloud, de tal manera que se puede realizar una gestión equilibrada de carga entre regiones, reduciendo la latencia.

Por último, gracias a Google Cloud CDN se puede realizar una distribución de contenido web y de vídeo rápida, fiable y con cobertura y alcance mundiales. Gracias a su activación con un solo clic para los usuarios de Cloud Load Balancing, ofrece conectividad a un mayor número de usuarios en todo el mundo de forma constante. Con esta herramienta se pueden ofrecer también, activos estáticos para aprovechar la forma más rápida de ofrecer contenido de este tipo reduciendo, nuevamente, la latencia de las aplicaciones y páginas web.

> **Hay que diseñar y optimizar los protocolos y códigos de red teniendo en mente las limitaciones del ancho de banda disponible, así como reducir los viajes de ida y vuelta de la información y, sobre todo, acercar los datos al usuario final.**

Bibliografía

1. “SRE fundamentals 2021: SLIs, SLAs and SLOs”
https://cloud.google.com/blog/products/devops-sre/sre-fundamentals-sli-vs-slo-vs-sla

“Caso de Éxito: IE University | Telefónica Cloud (telefonica.com)”
https://cloud.telefonica.com/es/case-studies/ie-university/

2. “What is Site Reliability Engineering (SRE)?”
https://sre.google/

3. “Acuerdo de Nivel de Servicio (ANS) de Cloud SQL”
https://cloud.google.com/sql/sla

4. “Web.dev measure” (using Lighthouse)

5. “Lighthouse”, an open-source project for improving the quality and performance of web pages
https://developers.google.com/web/tools/lighthouse/

6. “Compute Engine VMs with higher bandwidth”
https://cloud.google.com/compute/docs/networking/configure-vm-with-high-bandwidth-configuration

7. “Progressive Web Applications”
https://web.dev/progressive-web-apps/

8. “Service Workers: an introduction”
https://developers.google.com/web/fundamentals/primers/service-workers

9. “3 ways to optimize Cloud Run response times”
https://cloud.google.com/blog/topics/developers-practitioners/3-ways-optimize-cloud-run-response-times

10. “Cloud Run general development tips”
https://cloud.google.com/run/docs/tips/general

11. “Cloud Run adds ‘min instances’ feature for latency sensitive apps”
https://cloud.google.com/blog/products/serverless/cloud-run-adds-min-instances-feature-for-latency-sensitive-apps

12. “Entrega tráfico desde varias regiones”
https://cloud.google.com/run/docs/multiple-regions

13. “Global HTTP(S) Load Balancing and CDN now support serverless compute”
https://cloud.google.com/blog/products/networking/better-load-balancing-for-app-engine-cloud-run-and-functions

14. “Google Cloud CDN”
https://cloud.google.com/cdn