Año VII, v.1 2026 | Envío: 06/02/2026 | aceito: 06/05/2026 | publicación: 06/08/2026

Técnicas de reducción de latencia en pipelines de procesamiento en tiempo real con computación de borde y sin servidor.

Técnicas de reducción de latencia en tuberías de procesamiento en tiempo real con computación de borde y sin servidor

Técnicas de reducción de latencia en pipelines de procesamiento en tiempo real con computación en borde y serverless

Lucas Mohallem Ferraz

Resumen: Este artículo examina las técnicas de reducción de latencia en pipelines de procesamiento en tiempo real, con énfasis en la convergencia de las arquitecturas Edge Computing y Serverless. El análisis adopta un enfoque histórico-evolutivo, trazando el camino desde la consolidación de la computación en la nube —definida por el NIST como el acceso bajo demanda a recursos computacionales compartidos, elásticos y medibles— hasta la descentralización promovida por las arquitecturas Edge (Publicaciones Técnicas del NIST). Se argumenta que la reducción de latencia depende no solo de una mayor capacidad computacional, sino también de decisiones arquitectónicas sobre la ubicación del procesamiento, la granularidad funcional, la comunicación asíncrona, la partición de datos, la observabilidad y el arranque en frío.
mitigación en plataformas sin servidor.
Palabras clave: Computación perimetral. Sin servidor. Latencia. Procesamiento en tiempo real. Sistemas distribuidos.

1 Introducción

La evolución de los sistemas distribuidos ha desplazado progresivamente el eje computacional desde los centros de datos centralizados hacia arquitecturas híbridas, elásticas y distribuidas geográficamente.
Inicialmente, los modelos computacionales dependían en gran medida de grandes centros de datos centralizados que almacenaban, procesaban y distribuían información a clientes y aplicaciones. Sin embargo, el crecimiento exponencial del volumen de datos y del número de dispositivos conectados hizo cada vez más evidente la necesidad de arquitecturas más flexibles y escalables. En este contexto, la computación en la nube consolidó conceptos como la elasticidad, el autoservicio y el modelo de pago por uso, lo que permitió una mayor eficiencia operativa y una reducción de los costes de infraestructura.
A pesar de las ventajas que ofrece la computación en la nube, las aplicaciones modernas sensibles al tiempo han comenzado a enfrentar desafíos relacionados con la latencia y la variabilidad de la red. En entornos donde las respuestas del sistema deben producirse en milisegundos, la distancia física entre el usuario, la fuente de datos y los centros de datos centrales puede afectar significativamente el rendimiento percibido. Este problema se ha acentuado aún más en aplicaciones críticas como los sistemas de monitorización sanitaria, los vehículos autónomos, la realidad aumentada y los entornos industriales conectados, donde incluso retrasos mínimos pueden causar un impacto operativo significativo o riesgos para la seguridad.
En este contexto, han surgido paradigmas computacionales centrados en la descentralización del procesamiento, entre los que destacan la computación de borde (Edge Computing) y la computación sin servidor (Serverless Computing). La computación de borde acerca el procesamiento y el almacenamiento a las fuentes que generan los eventos, reduciendo el tiempo necesario para transmitir datos a la nube central. El modelo sin servidor, por su parte, introduce una abstracción operativa basada en la ejecución orientada a eventos y la escalabilidad automática, lo que reduce la necesidad de gestionar directamente la infraestructura. La combinación de estos enfoques representa una importante evolución arquitectónica para los sistemas distribuidos modernos, especialmente en escenarios de procesamiento en tiempo real.

Año VII, v.1 2026 | Envío: 06/02/2026 | aceito: 06/05/2026 | publicación: 06/08/2026

2. Evolución histórica: De la nube centralizada al borde programable

Históricamente, la computación distribuida ha experimentado diversas transformaciones estructurales impulsadas por el crecimiento de las redes y la demanda de una mayor eficiencia computacional. El primer gran cambio se produjo con la virtualización de la infraestructura, una tecnología que permitió consolidar múltiples máquinas virtuales en un único servidor físico. Esta innovación mejoró la utilización de los recursos y redujo los costos operativos, sentando las bases para el auge de la computación en la nube. La virtualización también impulsó la expansión de modelos de aprovisionamiento de recursos escalables, lo que permitió a las organizaciones consumir infraestructura de forma flexible y bajo demanda.

La segunda transformación significativa fue la consolidación de la computación en la nube mediante los modelos IaaS, PaaS y SaaS. Estos modelos redefinieron la forma en que se desarrollaban, implementaban y consumían las aplicaciones, lo que permitió la elasticidad, la alta disponibilidad y la abstracción de la infraestructura.

Sin embargo, a medida que las aplicaciones dependían cada vez más de respuestas en tiempo real, las limitaciones de las arquitecturas centralizadas se hicieron más evidentes. El creciente tráfico de datos, la necesidad de comunicación continua entre dispositivos inteligentes y la expansión del Internet de las Cosas intensificaron los problemas relacionados con la latencia, el ancho de banda y la dependencia de centros de datos remotos.

Ante estas limitaciones, surgió el paradigma de Edge Computing, caracterizado por el procesamiento computacional descentralizado. En lugar de dirigir todos los datos a servidores centrales, una parte del procesamiento se realiza cerca del origen de los eventos, lo que reduce las demoras y mejora la eficiencia operativa. Simultáneamente, surgió el paradigma Serverless como una evolución de la computación en la nube, simplificando la ejecución de aplicaciones mediante la abstracción del servidor y la escalabilidad automática basada en eventos. Como resultado, Edge Computing y Serverless se han convertido en pilares complementarios en el diseño de arquitecturas modernas para aplicaciones distribuidas sensibles a la latencia.

3. Latencia en sistemas de procesamiento en tiempo real

La latencia en los sistemas de procesamiento en tiempo real se refiere al intervalo de tiempo necesario para que un evento sea capturado, procesado y entregado al usuario final. En las arquitecturas distribuidas modernas, este proceso implica múltiples etapas, como la captura de eventos, la serialización de datos, la transmisión por red, la puesta en cola, el procesamiento, la persistencia y el consumo. Cada una de estas fases introduce pequeños retrasos que, al acumularse, pueden comprometer el rendimiento general del sistema. En aplicaciones críticas, como los sistemas financieros o la monitorización industrial, incluso diferencias de unos pocos milisegundos pueden afectar directamente la experiencia del usuario y la fiabilidad operativa.

En los sistemas distribuidos, el análisis de latencia no puede limitarse a la media aritmética de los tiempos de respuesta. Métricas estadísticas como p95, p99 y p99.9 se han vuelto esenciales para evaluar con precisión la calidad del servicio, ya que representan los peores escenarios que experimentan los usuarios. En muchos entornos críticos, una pequeña fracción de solicitudes extremadamente lentas puede ser suficiente para incumplir los acuerdos de nivel de servicio y provocar fallos operativos. Por lo tanto, las arquitecturas modernas deben diseñarse no solo teniendo en cuenta el rendimiento promedio, sino también la estabilidad, la previsibilidad y la resiliencia ante las fluctuaciones de carga.

Las principales causas de retraso en las arquitecturas distribuidas incluyen la distancia geográfica entre componentes, la congestión de la red, la serialización ineficiente, las llamadas síncronas encadenadas, la contención de colas y la inicialización tardía de funciones sin servidor. Además, los problemas relacionados con el almacenamiento remoto, la replicación robusta de datos y la ausencia de mecanismos de contrapresión pueden aumentar significativamente los tiempos de respuesta. Por lo tanto, abordar la latencia requiere un enfoque sistémico que considere tanto los aspectos físicos de la infraestructura como las decisiones arquitectónicas que rigen el flujo de datos y la organización de los servicios distribuidos.

Año VII, v.1 2026 | Envío: 06/02/2026 | aceito: 06/05/2026 | publicación: 06/08/2026

4 Técnicas de Reducción de Latencia Arquitectónica

Una de las principales técnicas para reducir la latencia consiste en el procesamiento en el borde de la red, conocido como Edge Computing. En este modelo, operaciones como el filtrado, la agregación, la validación y la inferencia ligera se realizan cerca de la fuente de datos, lo que reduce el volumen de información transmitida a la nube central. Este enfoque disminuye significativamente el tiempo de respuesta de las solicitudes, a la vez que reduce los costos de comunicación y el consumo de ancho de banda. En aplicaciones distribuidas a gran escala, Edge Computing también mejora la disponibilidad del sistema al permitir que ciertas funcionalidades continúen operando incluso cuando se interrumpe la conectividad con las regiones centrales.

Otra técnica relevante es la partición funcional de la canalización, en la que las distintas etapas de procesamiento se distribuyen según su sensibilidad temporal. Las operaciones que requieren respuestas inmediatas permanecen cerca de la fuente de datos, mientras que las tareas analíticas, históricas o de menor prioridad se descargan en entornos centrales basados en la nube. Como complemento, las arquitecturas orientadas a eventos utilizan colas, intermediarios de transmisión y registros distribuidos para desacoplar productores y consumidores, reduciendo las dependencias síncronas entre servicios. Este desacoplamiento aumenta la escalabilidad del sistema y elimina los cuellos de botella asociados al procesamiento concurrente.

Además, las técnicas específicas del paradigma sin servidor se han vuelto esenciales para la reducción de la latencia. Estas incluyen la mitigación del arranque en frío, el uso de concurrencia aprovisionada, la inicialización diferida y la adopción de entornos de ejecución más ligeros. Las cachés jerárquicas, la replicación selectiva de datos y el procesamiento incremental basado en ventanas y agregaciones parciales también desempeñan un papel fundamental. En lugar de recalcular estados completos, los sistemas mantienen información resumida actualizada continuamente, lo que reduce el costo computacional de las operaciones en tiempo real. La combinación de Edge Computing, Serverless y optimizaciones arquitectónicas avanzadas permite la construcción de pipelines altamente responsivos y escalables.

5. Modelo de referencia propuesto

Una moderna arquitectura de baja latencia basada en Edge Computing y Serverless puede estructurarse en múltiples capas especializadas. La primera capa corresponde a los dispositivos de origen de datos, incluidos sensores, sistemas embebidos, dispositivos móviles y navegadores. Estos elementos actúan como productores continuos de eventos y representan la principal fuente de generación de datos en las aplicaciones distribuidas contemporáneas. El crecimiento del Internet de las Cosas ha aumentado significativamente el número de dispositivos conectados, lo que requiere arquitecturas capaces de procesar grandes volúmenes de datos en plazos mínimos.

La segunda capa consta de nodos de borde responsables de las operaciones de procesamiento inicial. En esta etapa, se llevan a cabo mecanismos de filtrado, normalización, autenticación, inferencia ligera y almacenamiento en caché. Posteriormente, los datos se pueden enviar a una capa regional sin servidor, compuesta por funciones basadas en eventos responsables del enriquecimiento, el enrutamiento y la persistencia. El uso de funciones sin servidor permite el escalado automático bajo demanda, lo que reduce el desperdicio de recursos computacionales y simplifica la gestión de la infraestructura.

Finalmente, la arquitectura incorpora una capa de transmisión responsable de las colas, los registros distribuidos y los mecanismos de control de flujo, así como una capa analítica para el almacenamiento histórico, la auditoría y el entrenamiento de modelos de aprendizaje automático. La decisión arquitectónica central de este modelo radica en determinar dónde se ejecutará cada operación, teniendo en cuenta la sensibilidad temporal, los costos de transferencia de datos, los requisitos de consistencia y las restricciones regulatorias. El modelo de referencia propuesto busca, por lo tanto, equilibrar el rendimiento, la escalabilidad y la eficiencia operativa en entornos distribuidos en tiempo real.

Año VII, v.1 2026 | Envío: 06/02/2026 | aceito: 06/05/2026 | publicación: 06/08/2026

6. Discusión crítica

La convergencia de la computación perimetral y la arquitectura sin servidor representa un avance significativo en el diseño de sistemas distribuidos modernos, pero no elimina la complejidad inherente de estos entornos. En la práctica, la descentralización del procesamiento reduce la latencia, pero introduce nuevos desafíos relacionados con el despliegue distribuido, la sincronización de datos, la seguridad y la observabilidad.
Cuanto más dispersos geográficamente estén los nodos periféricos, más difícil será gestionar las operaciones y mantener la coherencia entre los componentes del sistema.
El paradigma sin servidor, a su vez, reduce significativamente la carga de la gestión de infraestructura, lo que permite a los equipos centrar sus esfuerzos en el desarrollo de la funcionalidad de las aplicaciones. Sin embargo, las plataformas sin servidor pueden imponer limitaciones en el tiempo máximo de ejecución, la variabilidad del rendimiento y la dependencia de un proveedor específico. Problemas como los arranques en frío y las restricciones de conectividad también pueden afectar a las aplicaciones que requieren una alta sensibilidad al tiempo, especialmente en escenarios con patrones de carga impredecibles o picos repentinos de solicitudes.
Por lo tanto, el diseño arquitectónico de las canalizaciones en tiempo real debe tener en cuenta múltiples factores más allá de la simple reducción de la latencia. Aspectos como el costo operativo, la portabilidad, la gobernanza, la resiliencia y la mantenibilidad deben evaluarse de forma holística. Para aplicaciones críticas, se recomienda encarecidamente el uso de métricas de observabilidad avanzadas, rastreo distribuido, simulación de fallas regionales y monitoreo continuo de percentiles de latencia. Solo un enfoque integrado que combine la ingeniería de software y la arquitectura distribuida puede garantizar un equilibrio adecuado entre
rendimiento, fiabilidad y sostenibilidad operativa a largo plazo.

Conclusión

La reducción de la latencia en los procesos en tiempo real se ha convertido en uno de los principales desafíos de la computación distribuida actual. El auge del Internet de las Cosas, las aplicaciones móviles y los sistemas inteligentes ha incrementado la demanda de respuestas rápidas, estables y escalables. En este contexto, la computación de borde (Edge Computing) se presenta como una estrategia fundamental para acercar el procesamiento y el almacenamiento a las fuentes que generan eventos, reduciendo así los retrasos causados por la distancia entre los dispositivos y los centros de datos centrales.
Paralelamente, el paradigma Serverless introdujo un modelo operativo más flexible y automatizado basado en la ejecución bajo demanda y la escalabilidad automática. Este enfoque ha reducido significativamente el esfuerzo requerido para la administración de la infraestructura, lo que permite una mayor agilidad en el desarrollo. aplicaciones distribuidas. Sin embargo, desafíos como los arranques en frío, la variabilidad del rendimiento y la dependencia del proveedor demuestran que la adopción de estas tecnologías requiere una planificación arquitectónica cuidadosa y mecanismos de optimización adecuados.
Se concluye que la combinación de Edge Computing y Serverless representa una etapa madura en la evolución de los sistemas distribuidos modernos. Sin embargo, la creación de pipelines de baja latencia requiere un enfoque sistémico que abarque la partición funcional, la comunicación asíncrona, la observabilidad avanzada, el procesamiento incremental y el uso inteligente de cachés jerárquicas.
En consecuencia, las organizaciones que buscan operar aplicaciones críticas en tiempo real deben adoptar arquitecturas que equilibren el rendimiento, la escalabilidad, la fiabilidad y la eficiencia operativa en entornos computacionales altamente distribuidos.

Referencias

AMAZON WEB SERVICES. ¿Qué es AWS Lambda? Disponible en: Documentación oficial de AWS Lambda. Consultado el 7 de mayo de 2026.

Año VII, v.1 2026 | Envío: 06/02/2026 | aceito: 06/05/2026 | publicación: 06/08/2026

AMAZON WEB SERVICES. Mejora del rendimiento de inicio con Lambda SnapStart. Disponible en: Documentación oficial de AWS Lambda. Consultado el 7 de mayo de 2026.

JONAS, E. et al. Programación en la nube simplificada: una perspectiva de Berkeley sobre la computación sin servidor. Berkeley: Universidad de California, 2019.

MELL, P.; GRANCE, T. Definición de computación en la nube del NIST. Publicación especial 800-145 del NIST. Gaithersburg: Instituto Nacional de Estándares y Tecnología, 2011.

SHI, W. et al. Edge Computing: Visión y desafíos. IEEE Internet of Things Journal, 2016.

SHI, W.; DUSTDAR, S. La promesa de la computación de borde. Computer, IEEE, 2016.