



Utilice CCM y QPM para optimizar el rendimiento de la recuperación y los planes de ejecución en Amazon Aurora PostgreSQL

AWSGuía prescriptiva



AWSGuía prescriptiva: Utilice CCM y QPM para optimizar el rendimiento de la recuperación y los planes de ejecución en Amazon Aurora PostgreSQL

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción	1
Destinatarios previstos	2
Resultados comerciales específicos	3
Administración de caché de clúster	4
¿Cómo funciona CCM?	4
Limitaciones	6
Casos de uso de CCM	7
Administración de planes de consultas	8
¿Cómo funciona QPM?	9
Limitaciones	9
Casos de uso de QPM	10
Recursos	11
Documentación de AWS	11
AWSpublicaciones de blog	11
AWStalleres	11
Historial de documentos	12
.....	xiii

Utilice CCM y QPM para optimizar el rendimiento de la recuperación y los planes de ejecución en Amazon Aurora PostgreSQL

Raunak Rishabh, Rohit Kapoor y Sujitha Sasikumaran, Amazon Web Services (AWS)

Enero de 2023 ([historial del documento](#))

A medida que las empresas se expanden, utilizan más y más datos para tomar decisiones críticas. Con el aumento de las cantidades de datos, es importante optimizar el rendimiento de la base de datos y estabilizarla durante los cambios en el sistema. Las cargas de trabajo altamente transaccionales, como las que implican transacciones financieras o pedidos de clientes, requieren un rendimiento estable, coherente y rápido, ya que un rendimiento deficiente puede afectar a la satisfacción de los clientes y a los ingresos empresariales. Para las bases de datos que gestionan estas cargas de trabajo altamente transaccionales, como las instancias de bases de datos de la edición compatible con Amazon Aurora PostgreSQL, es fundamental que comprenda e implemente las funciones de optimización del rendimiento disponibles.

[Compatible con Amazon Aurora PostgreSQL](#) es un motor de base de datos relacional totalmente administrado que le ayuda a configurar, operar y escalar las implementaciones de PostgreSQL. Es un motor de base de datos muy utilizado debido a su arquitectura de almacenamiento autosuficiente y a sus funciones, que ayudan a optimizar el rendimiento en situaciones de carga de trabajo reales con una sobrecarga de mantenimiento mínima.

Dos de estas funciones son la [administración de caché de clústeres \(CCM\)](#) y la [administración del plan de consultas \(QPM\)](#). El CCM le permite administrar los planes de ejecución de consultas generados por el optimizador para sus aplicaciones SQL. Ambas funciones pueden ayudar a optimizar el rendimiento de las consultas SQL al proporcionar un mayor control sobre la base de datos. El objetivo de esta guía es ayudar a los gerentes, propietarios de productos y arquitectos de bases de datos (DBA) a comprender los beneficios y los posibles resultados comerciales de implementar el CCM y el QPM.

Destinatarios previstos

El público objetivo de esta guía son las partes interesadas de la empresa que desean conocer las funciones disponibles para optimizar el rendimiento de las instancias de bases de datos compatibles con Amazon Aurora PostgreSQL y conocer los casos de uso de esas funciones.

Resultados comerciales específicos

Puede utilizar esta guía para lograr los siguientes resultados comerciales con la administración de caché del clúster (CCM):

- En caso de una conmutación por error, recupérese rápidamente para ayudar a mantener un rendimiento de carga de trabajo estable y óptimo.
- Reduzca las pérdidas empresariales causadas por un rendimiento deficiente de la carga de trabajo tras una conmutación por error.
- Ayude a evitar costes de E/S innecesarios después de una conmutación por error.

Puede utilizar esta guía para lograr los siguientes resultados empresariales con la gestión del plan de consultas (QPM):

- Obligar al optimizador a elegir a partir de un número limitado de planes aprobados para mejorar la estabilidad de los planes. Esto evita que el optimizador seleccione un plan subóptimo para una sentencia SQL determinada después de cambios en el sistema o la base de datos.
- Optimizar los planes de manera centralizada y, a continuación, distribuirlos globalmente.
- Determinar qué índices no se utilizan y evaluar los efectos de agregar o anular un índice.
- Reconozca automáticamente cualquier plan nuevo de costo mínimo que identifique el optimizador.
- Probar características de optimizador nuevas con un menor nivel de riesgo, porque puede optar por aceptar únicamente las modificaciones de planes que mejoren el rendimiento.

Administración de caché de clúster

El almacenamiento en caché es una de las funciones más importantes de cualquier base de datos (base de datos) porque ayuda a reducir la E/S del disco. Los datos a los que se accede con más frecuencia se almacenan en un área de memoria denominada caché de búfer. Cuando una consulta se ejecuta con frecuencia, recupera los datos directamente de la memoria caché en lugar del disco. Esto es más rápido y proporciona una mejor escalabilidad y rendimiento de las aplicaciones. El tamaño de la caché de PostgreSQL se configura mediante el `shared_buffers` parámetro. Para obtener más información, consulte [Memoria](#) (documentación de Memoria de PostgreSQL sobre Memoria).

Tras una conmutación por error, la [administración de caché de clústeres \(CCM\)](#) de la edición compatible con Amazon Aurora PostgreSQL está diseñada para mejorar el rendimiento de recuperación de aplicaciones y bases de datos. En una situación de conmutación por error típica sin CCM, puede que observe una degradación del rendimiento temporal, pero significativa. Esto se debe a que cuando se inicia la instancia de base de datos de conmutación por error, la caché del búfer está vacía. Una caché vacía se conoce también como caché fría. La instancia de base de datos debe leer desde el disco, lo que es más lento que leer desde la memoria caché.

Al implementar CCM, elige una instancia de base de datos de lector preferida y CCM sincroniza continuamente su memoria caché con la de la instancia de base de datos principal o de escritura. Si se produce una conmutación por error, la instancia de base de datos del lector preferido se promociona en la nueva instancia de base de datos del escritor. Como ya cuenta con una memoria caché, conocida como caché activa, esto minimiza el impacto de la conmutación por error en el rendimiento de la aplicación.

¿Cómo funciona la administración de la memoria caché de clústeres?

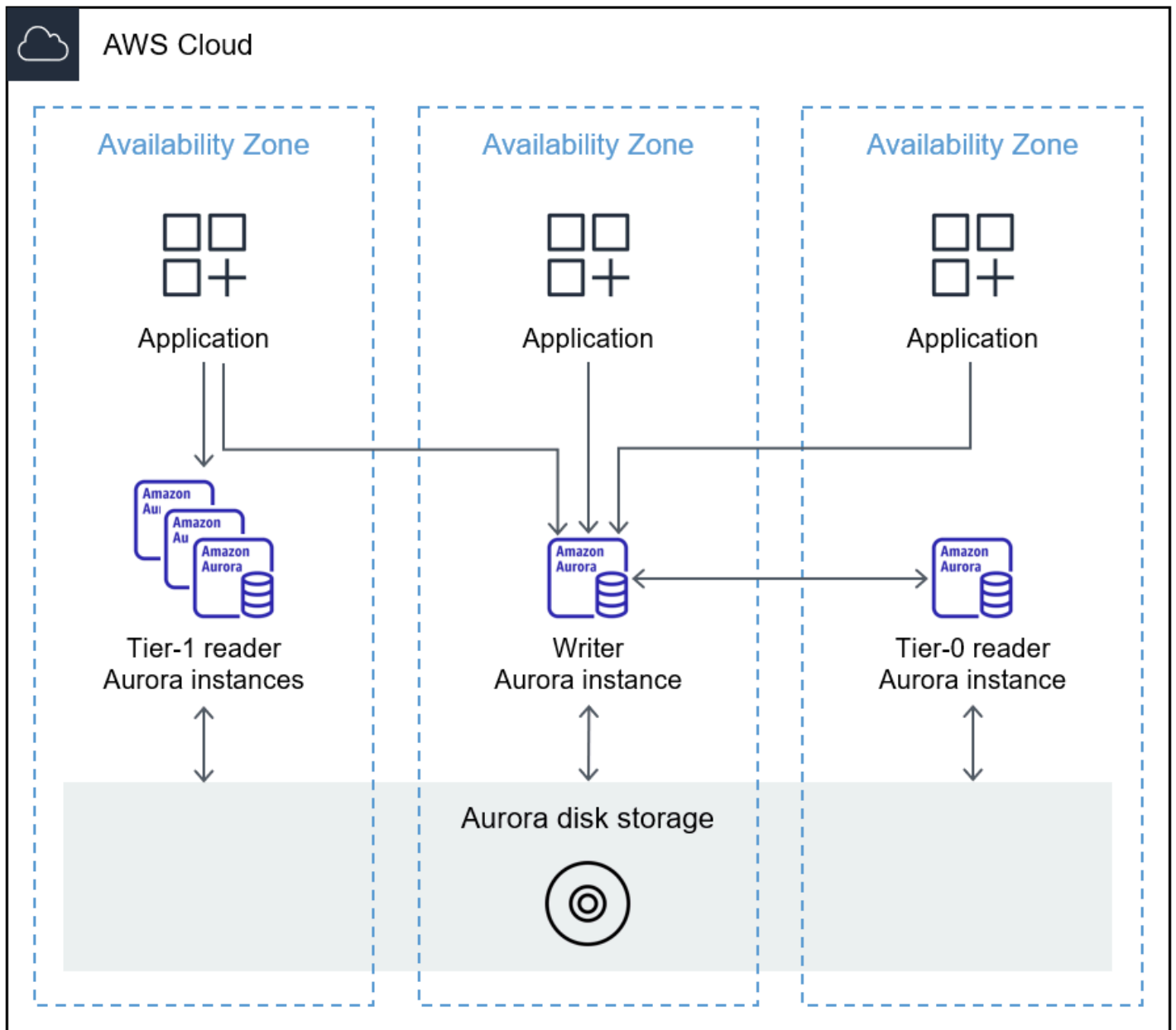
Las instancias de base de datos de conmutación por error se encuentran en zonas de disponibilidad diferentes a las de la instancia de base de datos principal de Writer. La instancia de base de datos del lector preferida es el objetivo de conmutación por error prioritario, que se especifica asignándole el nivel de prioridad de nivel 0.

Note

La prioridad de la capa de promoción es un valor que especifica el orden en el que se promociona el lector de Aurora en la instancia de base de datos del escritor después de un error. Los valores válidos son de 0 a 15, donde 0 es la primera prioridad y 15 la última. Para obtener más información sobre el nivel de promoción, consulte [Tolerancia a errores para un clúster de base de datos de Aurora](#). La modificación del nivel de promoción no provoca una interrupción.

CCM sincroniza la caché de la instancia de base de datos del escritor con la instancia de base de datos del lector preferido. La instancia de base de datos del lector envía el conjunto de direcciones del búfer que están actualmente almacenadas en caché a la instancia de base de datos del escritor como un filtro Bloom. Un filtro Bloom es una estructura de datos probabilística y eficiente en memoria que se utiliza para comprobar si un elemento es miembro de un conjunto. El uso de un filtro Bloom evita que la instancia de base de datos del lector envíe repetidamente las mismas direcciones del búfer a la instancia de base de datos del escritor. Cuando la instancia de base de datos del escritor recibe el filtro Bloom, compara los bloques de su caché de búfer y envía los búferes de uso frecuente a la instancia de base de datos del lector. De forma predeterminada, un búfer se considera de uso frecuente si tiene un recuento de uso superior a tres.

El siguiente diagrama muestra cómo CCM sincroniza la memoria caché del búfer de la instancia de base de datos del escritor con la instancia de base de datos del lector preferida.



Para obtener más información sobre CCM, consulte [Recuperación rápida después de una conmutación por error con administración de caché de clústeres para Aurora PostgreSQL](#) (documentación de Aurora) e [Introducción a la administración de la caché de clústeres de Aurora PostgreSQL](#) (entrada de AWS blog). Para obtener instrucciones sobre cómo configurar el CCM, consulte [Configuración de la administración de la caché de clústeres](#) (documentación de Aurora).

Limitaciones

La función CCM tiene las siguientes restricciones:

- La instancia de base de datos del lector debe tener el mismo tipo y tamaño de clase de instancia de base de datos que la instancia de base de datos del escritor, como `r5.2xlarge` o `db.r5.xlarge`.
- CCM no es compatible con los clústeres de base de datos Aurora PostgreSQL que forman parte de bases de datos globales de Aurora PostgreSQL.

Casos de uso de la administración de caché del clúster

En algunos sectores, como el comercio minorista, la banca y las finanzas, los retrasos de solo unos pocos milisegundos pueden provocar problemas de rendimiento de las aplicaciones y provocar una pérdida significativa de negocio. Dado que el CCM ayuda a recuperar el rendimiento de las aplicaciones y las bases de datos al sincronizar continuamente la caché del búfer de la instancia de base de datos principal con la instancia de respaldo preferida, puede ayudar a evitar las pérdidas empresariales asociadas a las conmutaciones por error.

Administración de planes de consultas

Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan de consultas. La regresión del plan de consultas se produce cuando el optimizador elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos.

En la edición compatible con Amazon Aurora PostgreSQL, la función de [administración del plan de consultas \(QPM\)](#) está diseñada para garantizar la adaptabilidad y la estabilidad del plan, independientemente de los cambios en el entorno de la base de datos que puedan provocar la regresión del plan de consultas. QPM proporciona cierto control sobre el optimizador. Con QPM, puede administrar el plan de ejecución de consultas generado por el optimizador para sus consultas SQL. El plan de ejecución de consultas obliga al optimizador a elegir entre los planes aprobados para las consultas críticas, a fin de optimizar su rendimiento.

Las empresas suelen implementar aplicaciones y bases de datos a nivel mundial o mantener varios entornos para cada base de datos de aplicaciones, como el desarrollo, el control de calidad, la puesta en escena, la preproducción, las pruebas y la producción. Mantener los planes de ejecución de consultas para cada base de datos, en cada entorno y en todos ellos, puede resultar complejo y llevar mucho tiempo. QPM puede exportar e importar planes gestionados compatibles con Amazon Aurora PostgreSQL de una base de datos a otra. Esto le ayuda a administrar el plan de ejecución de consultas de forma centralizada y a implementar bases de datos a nivel mundial. Puede utilizar esta función para investigar un conjunto de planes en una base de datos de preproducción, comprobar que funcionan bien y, a continuación, cargarlos en el entorno de producción.

El QPM también ofrece varios otros beneficios. Por ejemplo, puede usar QPM para mejorar los planes de ejecución que no se pueden cambiar en las aplicaciones o cuando no se pueden añadir sugerencias a la sentencia. QPM también detecta automáticamente los nuevos planes de coste mínimo que descubre el optimizador, para que pueda seguir optimizando los costes además del rendimiento.

Le recomendamos que habilite QPM. Cuando QPM está activado, el optimizador utiliza el plan de coste mínimo que usted haya aprobado. Esto ayuda a evitar la regresión y reduce el tiempo necesario para gestionar y corregir planes que no sean óptimos.

Hay dos enfoques diferentes para usar la función QPM: proactivo y reactivo. El enfoque proactivo está diseñado para ayudar a evitar que se produzca una regresión del rendimiento, y el enfoque

reactivo está diseñado para detectar y reparar las regresiones de rendimiento una vez que se producen. Puede seleccionar su enfoque por consulta. Para consultas complejas que pueden ser propensas a la regresión o para consultas críticas para el negocio, puede utilizar un enfoque proactivo y aprobar los planes óptimos para esas consultas. Si otras consultas experimentan una regresión del plan de consultas durante el tiempo de ejecución, puede utilizar un enfoque reactivo. Cuando detecte la regresión, cambie el estado de ese plan a `rejected` para que el optimizador elija otro plan aprobado. Para obtener más información, consulte [Prácticas recomendadas para la administración del plan de consultas de Aurora PostgreSQL](#) (documentación de Aurora).

¿Cómo funciona la gestión del plan de consultas?

A los planes se les asigna uno de los siguientes estados: `approved`, `unapproved`, `preferred`, o `rejected`. El optimizador establece el primer plan generado para cada declaración gestionada en `approved` y, a continuación, establece el estado de los planes adicionales en `unapproved`. Más adelante, puede evaluar los `unapproved` planes y cambiar su estado a `approved`, `preferred`, o `rejected`. Para obtener más información, consulte [Introducción a la administración del plan de consultas de Aurora PostgreSQL](#) (documentación de Aurora).

Los planes administrados pueden capturarse de forma manual o automática. El enfoque más común consiste en capturar automáticamente los planes de todas las declaraciones que se ejecuten dos o más veces. Sin embargo, también puede capturar manualmente los planes de un conjunto específico de estados de cuenta. Para obtener más información, consulte [Capturar los planes de ejecución de Aurora PostgreSQL](#) (documentación de Aurora).

Una vez configurado un plan administrado, el optimizador utiliza el `approved` plan `preferred` o costo mínimo que sea válido y esté habilitado para cada instrucción administrada. Para obtener información detallada, consulte [Cómo selecciona el optimizador qué plan ejecutar](#) (documentación de Aurora).

Para obtener instrucciones sobre cómo configurar la función QPM en Amazon Aurora PostgreSQL compatible, consulte [Administración de planes de ejecución de consultas para Aurora PostgreSQL](#) (documentación de Aurora).

Limitaciones

Para usar QPM, debe asegurarse de cumplir con los requisitos de las sentencias SQL compatibles, de que las sentencias no hacen referencia a las relaciones del sistema y de que la clase de instancia

de base de datos tiene suficientes vCPU vCPUs. Para obtener más información, consulte [las sentencias SQL admitidas y las limitaciones de administración del plan de consultas](#) (documentación de Aurora).

Casos de uso para la administración de planes de consultas de consultas

- Evitar la regresión del plan de consultas: mantener la versión de la base de datos actualizada ofrece muchos beneficios, como la mejora del rendimiento y la seguridad, el acceso a nuevas funciones, la solución de problemas conocidos y el cumplimiento de los requisitos reglamentarios. Sin embargo, existe el riesgo de que las actualizaciones de la base de datos provoquen una regresión en el rendimiento de algunas consultas. Este riesgo es mayor con las actualizaciones de versiones importantes, ya que pueden contener cambios que pueden no ser compatibles con versiones anteriores de las consultas de aplicaciones existentes. La implementación de QPM puede ayudar a prevenir la regresión y estabilizar el rendimiento durante los cambios en el sistema. Si actualiza las estadísticas, agrega un índice, cambia los parámetros o actualiza a una nueva versión de Amazon Aurora PostgreSQL, QPM detecta un plan nuevo pero sigue utilizando el plan aprobado, manteniendo así la estabilidad del plan.
- Funciones de prueba: puede ver el historial del plan de todas las sentencias SQL administradas y evaluar si las nuevas funciones de PostgreSQL o los cambios en el plan están mejorando el rendimiento. A continuación, puede decidir si desea implementar esas funciones o nuevos planes. Para obtener más información, consulte [Examinación de los planes de consultas de Aurora PostgreSQL en la vista dba_plans](#) (documentación de Aurora).
- Mejora de un plan: en algunos casos, puede que prefiera reparar un plan que no sea óptimo en lugar de rechazarlo, deshabilitarlo o eliminarlo. Para obtener más información, consulte [Arreglar planes mediante pg_hint_plan](#) (documentación de Aurora).

Recursos

Documentación de AWS

- [Administración de caché de clústeres \(CCM\)](#)
- [Administración de planes de consultas \(QPM\)](#)

AWSpublicaciones de blog

- [Introducción a Aurora PostgreSQL CCM](#)
- [Introducción a Aurora PostgreSQL QPM](#)

AWStalleres

- [Amazon Aurora Labs para PostgreSQL: CCM](#)
- [Amazon Aurora Labs para PostgreSQL: QPM](#)

Historial de documentos

En la siguiente tabla se describen los cambios importantes de esta guía. Si quieres recibir notificaciones sobre future actualizaciones, puedes suscribirte a una fuente [RSS](#).

Cambio	Descripción	Fecha
Publicación inicial	—	20 de enero de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.