



Usa CCM e QPM per ottimizzare le prestazioni di ripristino e i piani di esecuzione in Amazon Aurora PostgreSQL

AWSGuida prescrittiva



AWSGuida prescrittiva: Usa CCM e QPM per ottimizzare le prestazioni di ripristino e i piani di esecuzione in Amazon Aurora PostgreSQL

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Introduzione	1
Destinatari principali	1
Risultati aziendali mirare	2
Gestione della cache del cluster	3
Come funziona CCM?	3
Restrizioni	5
Casi d'uso per CCM	6
Gestione del piano di query	7
Come funziona QPM?	8
Restrizioni	8
Casi d'uso per QPM	9
Risorse	10
Documentazione AWS	10
AWSpost sul blog	10
AWSworkshop	10
Cronologia dei documenti	11
.....	xii

Usa CCM e QPM per ottimizzare le prestazioni di ripristino e i piani di esecuzione in Amazon Aurora PostgreSQL

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

Gennaio 2023 ([cronologia dei documenti](#))

Man mano che le aziende si espandono, utilizzano sempre più dati per prendere decisioni critiche. Con l'aumento della quantità di dati, è importante ottimizzare le prestazioni del database e stabilizzarlo durante le modifiche al sistema. I carichi di lavoro altamente transazionali, come quelli che coinvolgono transazioni finanziarie o ordini dei clienti, richiedono prestazioni stabili, coerenti e veloci perché le scarse prestazioni possono influire sulla soddisfazione dei clienti e sui ricavi aziendali. Per i database che gestiscono questi carichi di lavoro altamente transazionali, come le istanze di database Edition compatibili con Amazon Aurora PostgreSQL, è fondamentale comprendere e implementare le funzionalità di ottimizzazione delle prestazioni disponibili.

[Amazon Aurora PostgreSQL compatibile](#) è un motore di database relazionale completamente gestito che ti aiuta a configurare, gestire e scalare le distribuzioni PostgreSQL. È un motore di database ampiamente utilizzato grazie alla sua architettura di storage autosufficiente e alle sue funzionalità, che consentono di ottimizzare le prestazioni in scenari di carico di lavoro reali con un sovraccarico di manutenzione minimo.

Due di queste funzionalità sono [la gestione della cache del cluster \(CCM\)](#) e [la gestione del piano di interrogazione \(QPM\)](#). CCM consente di ripristinare le prestazioni di applicazioni e database in caso di failover e QPM consente di gestire i piani di esecuzione delle query generati dall'ottimizzatore per le applicazioni SQL. Entrambe queste funzionalità possono aiutare a ottimizzare le prestazioni delle query SQL fornendo un maggiore controllo sul database. Questa guida ha lo scopo di aiutare i manager, i proprietari di prodotti e gli architetti di database (DBA) a comprendere i vantaggi e i potenziali risultati aziendali dell'implementazione di CCM e QPM.

Destinatari principali

I destinatari di questa guida sono gli stakeholder aziendali che desiderano comprendere le funzionalità disponibili per ottimizzare le prestazioni delle istanze di database compatibili con Amazon Aurora PostgreSQL e comprendere i casi d'uso di tali funzionalità.

Risultati aziendali mirare

È possibile utilizzare questa guida per ottenere i seguenti risultati aziendali con la gestione della cache del cluster (CCM):

- In caso di failover, ripristina rapidamente per mantenere prestazioni del carico di lavoro stabili e ottimali.
- Riduci le perdite aziendali causate da scarse prestazioni del carico di lavoro dopo un failover.
- Contribuisci a prevenire costi di I/O non necessari dopo un failover.

È possibile utilizzare questa guida per ottenere i seguenti risultati aziendali con Query Plan Management (QPM):

- Migliorare la stabilità del piano forzando l'ottimizzatore a selezionare tra un numero ridotto di piani approvati. Questo impedisce all'ottimizzatore di selezionare un piano non ottimale per una determinata istruzione SQL dopo modifiche al sistema o al database.
- Ottimizzare i piani centralmente e distribuire i migliori a livello globale.
- Deterizzare gli indici in uso e valutare gli effetti dell'aggiunta o della rimozione di un indice.
- Riconosci automaticamente tutti i nuovi piani a costo minimo identificati dall'ottimizzatore.
- Provare nuove funzionalità dell'ottimizzatore con un rischio minimo, perché è possibile decidere di accettare solo le modifiche dei piani che migliorano le performance.

Gestione della cache del cluster

La memorizzazione nella cache è una delle funzionalità più importanti di qualsiasi database (DB) perché aiuta a ridurre l'I/O del disco. I dati a cui si accede più frequentemente sono archiviati in un'area di memoria chiamata buffer cache. Quando una query viene eseguita frequentemente, recupera i dati direttamente dalla cache anziché dal disco. È più veloce e offre una migliore scalabilità e prestazioni delle applicazioni. Si configura la dimensione della cache di PostgreSQL utilizzando il `shared_buffers` parametro. Per ulteriori informazioni, consulta la pagina relativa alla [memoria](#) (documentazione di PostgreSQL).

Dopo un failover, [la gestione della cache del cluster \(CCM\)](#) nell'edizione compatibile con Amazon Aurora PostgreSQL è progettata per migliorare le prestazioni di ripristino di applicazioni e database. In una tipica situazione di failover senza CCM, è possibile che si verifichi un degrado delle prestazioni temporaneo ma significativo. Ciò si verifica perché all'avvio dell'istanza database di failover, la cache del buffer è vuota. Una cache vuota è anche nota col nome di cache fredda. L'istanza DB deve leggere dal disco, che è più lenta della lettura dalla cache.

Quando implementate CCM, scegliete un'istanza DB del lettore preferita e CCM sincronizza continuamente la sua memoria cache con quella dell'istanza DB primaria o di scrittura. Se si verifica un failover, l'istanza database di scrittura preferita viene promossa al ruolo di nuova istanza database di scrittura. Poiché dispone già di una memoria cache, nota come cache calda, ciò riduce al minimo l'impatto del failover sulle prestazioni dell'applicazione.

Come funziona la gestione della cache del cluster?

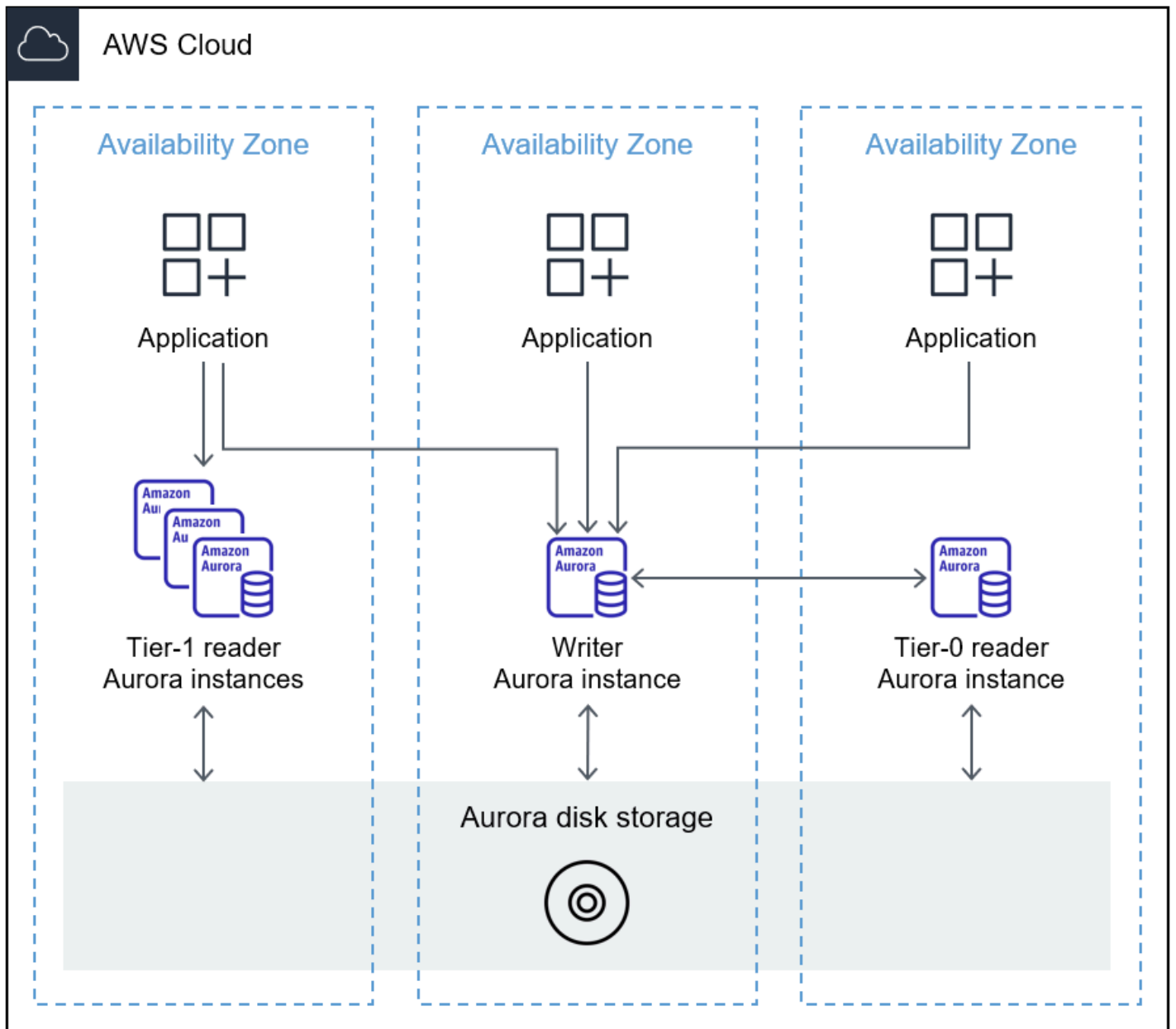
Le istanze DB di failover si trovano in zone di disponibilità diverse rispetto all'istanza DB primaria di scrittura. L'istanza DB del lettore preferita è l'obiettivo di failover prioritario, che viene specificato assegnandogli il livello di priorità tier-0.

Note

Il livello della priorità di promozione è un valore che specifica l'ordine in cui un'istanza di lettura di Aurora viene promossa al ruolo di istanza database di scrittura dopo un errore. I valori validi sono 0–15, dove 0 è la priorità massima e 15 è la priorità minima. Per ulteriori informazioni sul livello di promozione, consulta la pagina relativa alla [tolleranza ai guasti per un cluster database Aurora](#). La modifica del livello di promozione non causa un'interruzione.

CCM sincronizza la cache dall'istanza database del scrittore all'istanza database del lettore preferito. L'istanza DB del lettore invia il set di indirizzi buffer attualmente memorizzati nella cache all'istanza DB di writer come filtro bloom. Un filtro bloom è una struttura di dati probabilistica ed efficiente in termini di memoria che viene utilizzata per verificare se un elemento è membro di un set. L'uso di un filtro bloom impedisce all'istanza DB del lettore di inviare ripetutamente gli stessi indirizzi buffer all'istanza DB di writer. Quando l'istanza DB writer riceve il filtro bloom, confronta i blocchi nella sua cache buffer e invia i buffer usati di frequente all'istanza DB del lettore. Per impostazione predefinita, un buffer è considerato usato frequentemente se ha un numero di utilizzi maggiore di tre.

Il diagramma seguente mostra come CCM sincronizza la cache del buffer dell'istanza DB di Writer con l'istanza DB preferita del lettore.



Per ulteriori informazioni su CCM, vedere [Ripristino rapido dopo il failover con gestione della cache del cluster per Aurora PostgreSQL](#) (documentazione Aurora) e [Introduzione alla gestione della cache del cluster Aurora PostgreSQL](#) (postAWS sul blog). Per istruzioni su come configurare CCM, vedere [Configurazione della gestione della cache del cluster](#) (documentazione Aurora).

Restrizioni

La funzionalità CCM presenta i seguenti limiti:

- L'istanza database del lettore deve avere lo stesso tipo di classe di istanza database del lettore e la stessa dimensione dell'istanza database di scrittura, ad esempio `r5.2xlarge` o `db.r5.xlarge`.
- CCM non è supportata per i cluster Aurora PostgreSQL database che fanno parte dei database globali Aurora.

Casi d'uso per la gestione della cache del cluster

In alcuni settori, come la vendita al dettaglio, il settore bancario e finanziario, i ritardi di pochi millisecondi possono causare problemi di prestazioni delle applicazioni e comportare una significativa perdita di attività. Poiché CCM aiuta a ripristinare le prestazioni delle applicazioni e del database sincronizzando continuamente la cache buffer dell'istanza del database principale con l'istanza di backup preferita, può aiutare a prevenire le perdite aziendali associate ai failover.

Gestione del piano di query

Le modifiche a statistiche, vincoli, impostazioni dell'ambiente, associazioni dei parametri di query e aggiornamenti al motore di database PostgreSQL possono tutte causare la regressione del piano di query. La regressione del piano di query si verifica quando l'ottimizzatore sceglie un piano non ottimale rispetto a prima di una determinata modifica all'ambiente del database.

Nell'edizione compatibile con Amazon Aurora PostgreSQL, la funzionalità di [gestione del piano di query \(QPM\)](#) è progettata per garantire l'adattabilità e la stabilità del piano, indipendentemente dalle modifiche all'ambiente del database che potrebbero causare la regressione del piano di query. QPM fornisce un certo controllo sull'ottimizzatore. Con QPM, puoi gestire il piano di esecuzione delle query generato dall'ottimizzatore per le query. Il piano di esecuzione delle query impone all'ottimizzatore di scegliere tra i piani approvati per le query critiche, al fine di ottimizzarne le prestazioni.

Le aziende in genere distribuiscono applicazioni e database a livello globale o gestiscono diversi ambienti per ogni database di applicazioni, ad esempio sviluppo, controllo qualità, staging, preproduzione, test e produzione. La gestione dei piani di esecuzione delle query per ogni database, in ogni ambiente e in tutti gli ambienti Regioni AWS può essere un'operazione complessa e dispendiosa in termini di tempo. QPM può esportare e importare piani gestiti compatibili con Amazon Aurora PostgreSQL da un database all'altro. Questo ti aiuta a gestire il piano di esecuzione delle query a livello centrale e a distribuire i database a livello globale. È possibile utilizzare questa funzionalità per esaminare una serie di piani in un database di preproduzione, verificare che funzionino correttamente e quindi caricarli nell'ambiente di produzione.

QPM offre anche molti altri vantaggi. Ad esempio, è possibile utilizzare QPM per migliorare i piani di esecuzione che non possono essere modificati nelle applicazioni o quando non è possibile aggiungere suggerimenti alla dichiarazione. QPM rileva automaticamente anche i nuovi piani a costo minimo rilevati dall'ottimizzatore, in modo da poter continuare a ottimizzare i costi oltre alle prestazioni.

Si consiglia di attivare QPM. Quando QPM è abilitato, l'ottimizzatore utilizza il piano a costo minimo che hai approvato. Questo aiuta a prevenire la regressione e riduce il tempo necessario per gestire e correggere i piani non ottimali.

Esistono due approcci diversi per l'utilizzo della funzionalità QPM: proattivo e reattivo. L'approccio proattivo è progettato per aiutare a prevenire la regressione delle prestazioni e l'approccio reattivo è progettato per rilevare e correggere le regressioni delle prestazioni dopo che si sono verificate.

Puoi selezionare il tuo approccio in base alle singole richieste. Per le interrogazioni complesse che potrebbero essere soggette a regressione o per le interrogazioni aziendali critiche, è possibile utilizzare un approccio proattivo e approvare i piani ottimali per tali interrogazioni. Se altre query presentano una regressione del piano di query durante l'esecuzione, è possibile utilizzare un approccio reattivo. Quando rilevi la regressione, modifica lo stato di quel piano `rejected` in modo che l'ottimizzatore scelga un piano diverso e approvato. Per ulteriori informazioni, vedere [Procedure consigliate per la gestione dei piani di query di Aurora PostgreSQL](#) (documentazione Aurora).

Come funziona la gestione dei piani di interrogazione?

Ai piani viene assegnato uno dei seguenti stati: `approved`, `unapproved`, `preferred`, o `rejected`. L'ottimizzatore imposta il primo piano generato per ogni rendiconto gestito `approved` e quindi imposta lo stato dei piani aggiuntivi su `unapproved`. Successivamente, puoi valutare i `unapproved` piani e modificarne lo stato in `approved`, `preferred`, o `rejected`. Per ulteriori informazioni, vedere [Understanding Aurora PostgreSQL Query Plan management](#) (documentazione Aurora).

I piani gestiti possono essere acquisiti manualmente o automaticamente. L'approccio più comune consiste nell'acquistare automaticamente i piani per tutti i rendiconti che vengono eseguiti due o più volte. Tuttavia, puoi anche acquisire manualmente i piani per una serie specifica di rendiconti. Per ulteriori informazioni, vedere [Acquisizione dei piani di esecuzione di Aurora PostgreSQL](#) (documentazione Aurora).

Dopo aver impostato un piano gestito, l'ottimizzatore utilizza il costo `preferred` o il `approved` piano che è valido e abilitato per ciascuna istruzione gestita. Per informazioni dettagliate, vedi [In che modo l'ottimizzatore sceglie quale piano eseguire](#) (documentazione Aurora).

Per istruzioni sulla configurazione della funzionalità QPM in Amazon Aurora PostgreSQL compatibile, consulta [Gestione dei piani di esecuzione delle query per Aurora PostgreSQL](#) (documentazione Aurora).

Restrizioni

Per utilizzare QPM, è necessario assicurarsi di soddisfare i requisiti per le istruzioni SQL supportate, che le istruzioni non facciano riferimento alle relazioni di sistema e che la classe di istanza DB disponga di vCPU sufficienti. Per ulteriori informazioni, vedere Istruzioni [SQL supportate](#) e [limitazioni per la gestione dei piani di interrogazione](#) (documentazione Aurora).

Casi d'uso per la gestione del piano di query

- **Prevenzione della regressione del piano di query:** mantenere aggiornata la versione del database offre molti vantaggi, come prestazioni e sicurezza migliorate, accesso a nuove funzionalità, correzioni di problemi noti e conformità ai requisiti normativi. Tuttavia, esiste il rischio che gli aggiornamenti del database possano causare una regressione delle prestazioni di alcune query. Questo rischio è maggiore con gli aggiornamenti delle versioni principali perché possono contenere modifiche che potrebbero non essere retrocompatibili con le query delle applicazioni esistenti. L'implementazione di QPM può aiutare a prevenire la regressione e stabilizzare le prestazioni durante le modifiche al sistema. Se aggiorni le statistiche, aggiungi un indice, modifichi i parametri o esegui l'upgrade a una nuova versione di Amazon Aurora compatibile con PostgreSQL, QPM rileva un nuovo piano ma continua a utilizzare il piano approvato, mantenendo così la stabilità del piano.
- **Funzionalità di test:** è possibile visualizzare la cronologia del piano per tutte le istruzioni SQL gestite e valutare se le nuove funzionalità di PostgreSQL o le modifiche al piano stanno migliorando le prestazioni. Potrai quindi decidere se implementare tali funzionalità o nuovi piani. Per ulteriori informazioni, vedere [Esame dei piani di query Aurora PostgreSQL nella vista dba_plans](#) (documentazione Aurora).
- **Miglioramento di un piano:** in alcuni casi, si potrebbe preferire correggere un piano non ottimale piuttosto che rifiutarlo, disabilitarlo o cancellarlo. Per ulteriori informazioni, vedere [Correzione dei piani utilizzando pg_hint_plan](#) (documentazione Aurora).

Risorse

Documentazione AWS

- [Gestione della cache del cluster \(CCM\)](#)
- [Gestione del piano di query \(QPM\)](#)

AWSpost sul blog

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

AWSworkshop

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

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti apportate a questa guida. Se desideri ricevere una notifica sugli aggiornamenti future, puoi iscriverti a un [feed RSS](#).

Modifica	Descrizione	Data
Pubblicazione iniziale	—	20 gennaio 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.