



使用 CCM 和 QPM 优化 Amazon Aurora PostgreSQL 中的恢复性能和执行计划

# AWS规范性指导



---

# AWS规范性指导: 使用 CCM 和 QPM 优化 Amazon Aurora PostgreSQL 中的恢复性能和执行计划

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

# Table of Contents

介绍 .....	1
目标受众 .....	1
有针对性的业务成果 .....	2
集群缓存管理 .....	3
CCM 如何工作？ .....	3
限制 .....	4
CCM 的使用案例 .....	5
查询计划管理 .....	6
QPM 是如何运行的？ .....	6
限制 .....	7
QPM 使用案例 .....	7
资源 .....	8
AWS 文档 .....	8
AWS 博客文章 .....	8
AWS 研讨会 .....	8
文档历史记录 .....	9
.....	x

# 使用 CCM 和 QPM 优化 Amazon Aurora PostgreSQL 中的恢复性能和执行计划

Amazon Web Services 的 Raunak Rishabh、Rohit Kapoor 和 Sujitha Sasikumaran (AWS)

2023 年 1 月 ( [文件历史记录](#) )

随着业务的扩展，他们使用越来越多的数据来做出关键决策。随着数据量的增加，优化数据库性能并在系统更改期间稳定数据库性能非常重要。高交易性工作负载，例如涉及金融交易或客户订单的工作负载，需要稳定、一致和快速的性能，因为性能不佳会影响客户满意度和业务收入。对于处理这些高事务性工作负载的数据库，例如 Amazon Aurora PostgreSQL 兼容版数据库实例，了解和实现可用的性能优化功能至关重要。

[兼容 Amazon Aurora PostgreSQL](#) 是一个完全托管的关系数据库引擎，可帮助您设置、操作和扩展 PostgreSQL 部署。它是一个广泛使用的数据库引擎，因为它具有自给自足的存储架构和功能，可帮助您以最小的维护开销优化实际工作负载场景中的性能。

其中两个功能是[集群缓存管理 \(CCM\)](#) 和[查询计划管理 \(QPM\)](#)。CCM 可让您管理优化器为 SQL 应用程序生成的查询执行计划。这两项功能都可以通过提供对数据库的更多控制来帮助优化 SQL 查询的性能。本指南旨在帮助经理、产品所有者和数据库架构师 (DBA) 了解实施 CCM 和 QPM 的好处和潜在的业务成果。

## 目标受众

本指南的目标受众是希望了解用于优化 Amazon Aurora PostgreSQL 兼容数据库实例性能的可用功能并了解这些功能的用例的业务利益相关者。

## 有针对性的业务成果

您可以使用本指南通过集群缓存管理 (CCM) 实现以下业务成果：

- 发生故障转移时，快速恢复以帮助保持稳定和最佳的工作负载性能。
- 减少故障转移后因工作负载性能不佳而造成的业务损失。
- 帮助防止故障切换后产生不必要的 I/O 成本。

您可以使用本指南通过查询计划管理 (QPM) 实现以下业务成果：

- 通过强制优化程序从少量的已批准计划中选择，提高计划的稳定性。这可以防止优化程序在系统或数据库更改后为给定的 SQL 语句选择次优计划。
- 集中优化计划，然后在全局范围内分发计划。
- 确定未在使用哪些索引，并评估添加或删除索引的影响。
- 自动识别优化器识别的任何新的最低成本计划。
- 尝试使用风险最小的新优化程序功能，因为您可以决定仅接受将提高性能的计划更改。

## 集群缓存管理

缓存是任何数据库 (DB) 最重要的功能之一，因为它有助于减少磁盘 I/O。最常访问的数据存储在称为缓冲区缓存的内存区域中。当查询频繁运行时，它会直接从缓存中检索数据，而不是从磁盘检索数据。这速度更快，可提供更好的可扩展性和应用程序性能。您可以使用 `shared_buffers` 参数配置 PostgreSQL 缓存大小。有关更多信息，请参阅 [内存](#) (PostgreSQL 文档)。

故障转移后，Amazon Aurora PostgreSQL 兼容版中的 [集群缓存管理 \(CCM\)](#) 旨在提高应用程序和数据库的恢复性能。在没有 CCM 的典型故障转移情况下，您可能会看到暂时的但明显的性能下降。发生性能下降，是因为当故障转移数据库实例启动时，缓冲区缓冲区是空的。空缓存也称为冷缓存。数据库实例必须从磁盘读取，这比从缓存读取要慢。

在实现 CCM 时，选择首选的读取器数据库实例，CCM 会持续将其缓存内存与主数据库实例或写入器数据库实例的缓存内存同步。如果发生故障转移，首选读取器数据库实例将升级为新写入器数据库实例。由于它已经具有缓存内存，称为热缓存，因此可以最大限度地减少故障转移对应用程序性能的影响。

## 集群缓存管理是如何运作的？

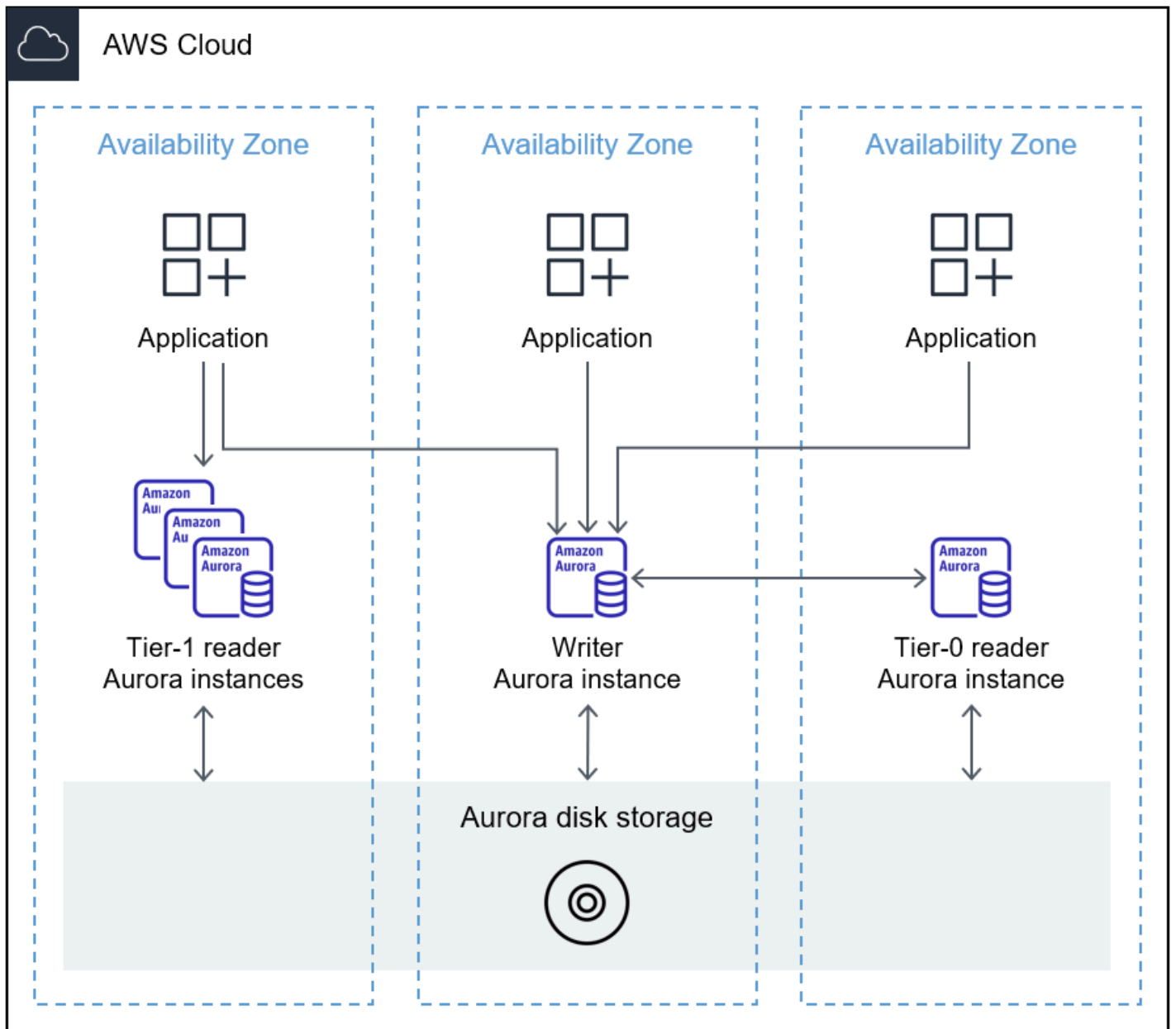
故障转移数据库实例与主 writer 数据库实例位于不同的可用区域。首选 reader 数据库实例是优先级故障转移目标，通过为其分配 Tier-0 优先级来指定。

### Note

提升层优先级 是指定在发生故障后将 Aurora 读取器提升为写入器数据库实例的顺序。有效值为 0–15，0 是第一个优先级，15 是最后一个。有关提升层的更多信息，请参阅 [Aurora 数据库集群的容错能力](#)。修改提升层不会导致中断。

CCM 将写入器数据库实例中的缓存同步到首选读取器数据库实例。读取器数据库实例将当前缓存的一组缓冲区地址作为布隆过滤器发送到 writer 数据库实例。Bloom 过滤器是一种概率的、内存效率高的数据结构，用于测试元素是否是集合的成员。使用布隆过滤器可防止读取器数据库实例重复向写入器数据库实例发送相同的缓冲区地址。当 writer 数据库实例收到布隆过滤器时，它会比较其缓冲区缓存中的数据块，并将常用的缓冲区发送到读取器数据库实例。默认情况下，如果缓冲区的使用次数大于三，则认为该缓冲区的使用次数是经常使用的。

下图显示 CCM 如何将 writer 数据库实例的缓冲区缓存与首选读取器数据库实例同步。



有关 CCM 的更多信息，请参阅 [Aurora PostgreSQL 的集群缓存管理在故障转移后快速恢复](#) (Aurora 文档) 和 [Aurora PostgreSQL 集群缓存管理简介](#) (AWS 博客文章)。有关如何配置 CCM 的说明，请参阅 [配置集群缓存管理](#) (Aurora 文档)。

## 限制

CCM 功能具有以下限制：

- 读取器数据库实例必须具有与写入器数据库实例相同的数据库实例类型和大小，例如 r5.2xlarge 或 db.r5.xlarge。
- 作为 Aurora 全局数据库一部分的 Aurora PostgreSQL 数据库集群不支持 CCM。

## 集群缓存管理的使用案例

对于某些行业，例如零售、银行和金融，仅几毫秒的延迟就会导致应用程序性能问题并导致重大业务损失。由于 CCM 通过持续将主数据库实例的缓冲缓存同步到首选备份实例来帮助恢复应用程序和数据库性能，因此它可以帮助防止与故障转移相关的业务损失。



# 查询计划管理

更改统计数据、约束、环境设置、查询参数绑定以及升级 PostgreSQL 数据库引擎都可能导致查询计划回归。查询计划回归（当优化程序选择的计划更不理想时，就会出现查询计划回归）之前选择的计划更不理想时，就会出现查询计划回归。

在 Amazon Aurora PostgreSQL [QL 兼容版本中](#)，[查询计划管理 \(QPM\)](#) 功能旨在确保计划的适应性和稳定性，无论数据库环境发生什么变化都可能导致查询计划回归。QPM 为优化器提供了一些控制。使用 QPM，您可以管理优化器为 SQL 查询生成的查询执行计划。查询执行计划强制优化器从您批准的关键查询计划中进行选择，以优化其性能。

企业通常在全球部署应用程序和数据库，或者为每个应用程序数据库维护多个环境，例如开发、QA、暂存、预生产、测试和生产。维护每个数据库、每个环境中的查询执行计划 AWS 区域可能既复杂又耗时。QPM 可以将兼容 Amazon Aurora PostgreSQL 的托管计划从一个数据库导出和导入到另一个数据库。这可以帮助您集中管理查询执行计划并在全局范围内部署数据库。您可以使用此功能调查预生产数据库中的一组计划，验证它们是否运行良好，然后将它们加载到生产环境中。

QPM 还提供其他一些好处。例如，您可以使用 QPM 来改进无法在应用程序中更改或无法在语句中添加提示时的执行计划。QPM 还会自动检测优化器发现的新的最低成本计划，因此除了性能外，您还可以继续优化成本。

建议您启用 QPM。启用 QPM 后，优化器将使用您批准的最低成本计划。这有助于防止回归并减少管理和修复次优计划所需的时间。

使用 QPM 功能有两种不同的方法：主动和被动。主动方法旨在帮助防止性能回归的发生，而反应性方法旨在性能回归发生后检测和修复。您可以根据每个查询选择方法。对于可能容易出现回归的复杂查询或关键业务查询，您可以使用主动方法为这些查询批准最佳计划。如果其他查询在运行时遇到查询计划回归，则可以使用响应式方法。检测到回归时，将该计划的状态更改为 `rejected` 以便优化程序选择其他已批准的计划。有关更多信息，请参阅 [Aurora PostgreSQL 查询计划管理的最佳实践](#)（Aurora 文档）。

## 查询计划管理是如何运作的？

计划被分配以下状态之一：`approved`、`unapprovedpreferred`、或 `rejected`。优化器将每个托管语句的第一个生成的计划设置为 `approved` 然后将其他计划的状态设置为 `unapproved`。稍后，您可以评估 `unapproved` 计划并将其状态更改为 `approvedpreferred`、或 `rejected`。有关更多信息，请参阅 [了解 Aurora PostgreSQL 查询计划管理](#)（Aurora 文档）。

托管计划可以手动或自动获取。最常见的方法是自动捕获运行两次或更多次的所有语句的计划。但是，您也可以手动获取一组特定报表的计划。有关更多信息，请参阅[捕获 Aurora PostgreSQL 执行计划](#) (Aurora 文档)。

设置托管计划后，优化程序为各个托管语句使用有效且启用的最低成本 preferred 或 approved 计划。有关详细信息，请参阅[优化程序如何选择要运行的计划](#) (Aurora 文档) 文档)。

有关在兼容 Aurora PostgreSQL 的 Amazon Aurora 中配置 QPM 功能的说明，请参阅[管理 Aurora PostgreSQL 的查询执行计划](#) (Aurora 文档)。

## 限制

要使用 QPM，必须确保满足支持的 SQL 语句的要求，您的语句不引用系统关系，并且您的数据库实例类有足够的 vCPUs。有关更多信息，请参阅[支持的 SQL 语句](#)和[查询计划管理限制](#) (Aurora 文档)。

## 查询计划管理的使用案例

- 防止查询计划回归 — 保持数据库版本处于最新状态可以带来许多好处，例如提高性能和安全性、访问新功能、修复已知问题以及符合监管要求。但是，数据库更新有可能导致某些查询出现性能下降。在主要版本升级中，这种风险更高，因为它们可能包含与现有应用程序查询不向后兼容的更改。在系统变更期间，实施 QPM 可以帮助防止回归并稳定性能。如果您刷新统计数据、添加索引、更改参数或升级到与 Amazon Aurora PostgreSQL 兼容的新版本，QPM 会检测到新计划，但会继续使用已批准的计划，从而保持计划的稳定性。
- 测试功能-您可以查看所有托管 SQL 语句的计划历史记录，并评估新的 PostgreSQL 功能或计划更改是否在提高性能。然后，您可以决定是实现这些功能还是实施新计划。有关更多信息，请参阅在[dba\\_plans 视图中检查 Aurora PostgreSQL 查询计划](#)。
- 改进计划-在某些情况下，您可能会偏好修复次理想时，而不是拒绝、禁用或删除该计划。有关更多信息，请参阅[使用 pg\\_hint\\_plan 修复计划](#) (Aurora 文档)。

# 资源

## AWS 文档

- [集群缓存管理 \(CCM\)](#)
- [查询计划管理 \(QPM\)](#)

## AWS 博客文章

- [Aurora PostgreSQL CCM 简介](#)
- [Aurora PostgreSQL QPM 简介](#)

## AWS 研讨会

- [PostgreSQL 版 Amazon Aurora 实验室：CCM](#)
- [PostgreSQL 版 Amazon Aurora 实验室：QPM](#)

# 文档历史记录

下表描述了本指南的重大更改。如果您想收到有关future 更新的通知，可以订阅 [RSS 提要](#)。

变更	说明	日期
<a href="#">首次出版</a>	—	2023 年 1 月 20 日

---

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。