

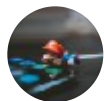


获得无限制访问

在应用程序中打开



出版于 创业公司



梅格的科技角

跟随

2020 年 7 月 21 日 · 8 分钟阅读 · 听

节省



# 亚马逊 Aurora 数据库的架构以及为什么它更好

## 抽象的

Aurora 数据库是亚马逊的云原生数据库。它可以容纳多达 64TB 的数据，并且比 MySQL 数据库快得多。许多公司都采用了 Aurora 数据库。

在本文中，我们将介绍亚马逊的 Aurora 数据库的架构。我们将从传统关系数据库系统的架构开始，例如 MySQL 数据库，并讨论它们的局限性。然后，我们将讨论 Aurora 数据库如何扩展传统数据库的功能以提高可用性、可靠性和可扩展性。

如果您对 AWS 的另一个特色数据库 DynamoDB 的架构感兴趣，可以阅读我的另一篇博客[\[点击这里\]](#)。

## 传统数据库的架构

数据库是数据的集合，通常描述组织的实体和活动。例如，电子商务数据库可能包含有关客户、产品和销售的信息。

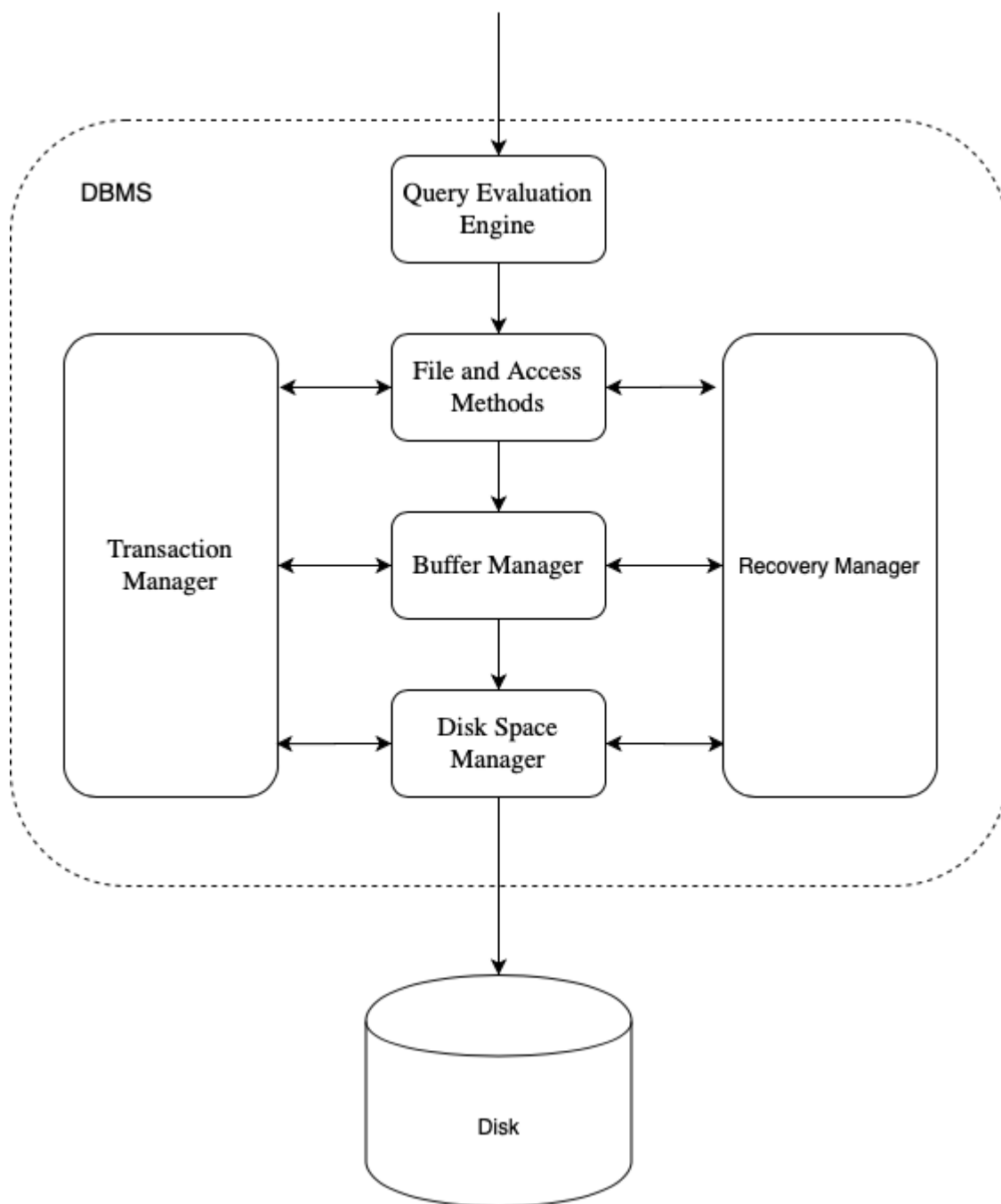
数据库管理系统 (DBMS) 旨在维护和利用这些数据。具体来说，DBMS 通常提供以下功能。

- 数据的可靠存储。
- 用于创建、更新、删除和查询数据的简单界面。





作。



传统 DBMS 的架构

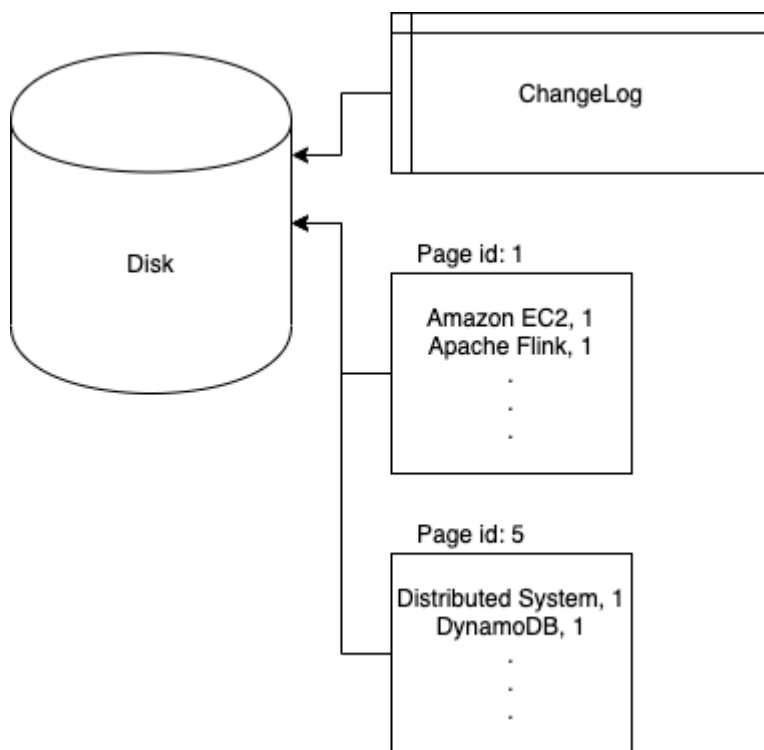
### • 示例 1: 写入数据库

想象一下我们想要在数据库表中插入两行的情况。我们将向数据库发出以下命令：





该请求首先被转发到查询评估引擎。查询评估引擎解析SQL命令，明白该命令是在任务表中插入两行。引擎下方的3层、文件和访问方法、缓冲区管理器和磁盘空间管理器协同工作以将数据持久保存到磁盘中。



磁盘结构

DBMS 以固定大小（称为页）从磁盘读取数据或将数据写入磁盘。页通常为 4KB 或 8KB，是 DBMS 的可配置参数。每个页面都关联一个唯一的 id，页面 id。每个数据库表中的数据都存储在几页中。为了加快搜索速度，对不同页面中的数据进行了排序。例如，第 1 页包含键在 [a, c] 范围内的行。第 2 页用于 [d, f] 范围内的键。除了页面，DBMS 还在磁盘中存储了一个变更日志。我们需要更新日志的原因是一些操作需要修改不同的页面，我们不能原子地将不同的页面刷新到磁盘中。在将第一页刷新到磁盘并使 DBMS 处于不一致状态后，DBMS 可能会崩溃。因此，我们首先记录我们想要进行的更改，这些更改可以自动刷新到磁盘。如果 DBMS 崩溃，我们可以根据变更日志重建状态。有关示例，请参见下一段。

在我们的示例中，DBMS 将识别出我们需要将行 (“Aurora”, 1) 插入到第 1 页，将行 (“DBMS”, 2) 插入到第 5 页。在修改第 1 页和第 5 页中的数据之前，DBMS 首先将条目写





操作：插入

数据： (“Aurora”, 1) , (“DBMS”, 2)

如果在部分更新数据库后 DBMS 崩溃，Recovery Manager 将读取更改日志并继续更新数据库并确保插入两行。

更新日志刷新到磁盘后，DBMS 将 (“Aurora”,1) 插入到第 1 页并将该页面刷新到磁盘。然后它将行 (“DBMS”, 2) 插入到第 5 页并将其刷新到磁盘。

现在写入操作已成功完成，DBMS 可以返回到客户端应用程序。

- 示例 2：查询数据库

查询数据库的步骤类似。考虑以下命令

```
SELECT * FROM tasks WHERE title = "Apache Flink";
```

查询执行引擎首先解析命令。然后 DBMS 将识别哪个页面可能包含数据。在我们的例子中，第 1 页可能包含我们想要的数据库。如果缓冲区中不存在该页面，则磁盘空间管理器会将其读入内存。DBMS 然后在页面中搜索数据并将结果返回给客户端应用程序。

### 传统 DBMS 的局限性

传统 DBMS 几十年来一直运行良好，是几乎所有软件应用程序的关键组件。然而随着云的出现，以及对可扩展性、可靠性和可用性的更高要求，传统的 DBMS 逐渐跟不上人们的期望。

- 可扩展性

传统的 DBMS 仅在单台机器上运行。我们只能通过使用更强大的计算机来扩展它。这种方法很昂贵，而且不可扩展。磁盘支持的每秒 IO 操作数 (IOPS) 很快就会成为系统的瓶颈。

- 可靠性





- 可用性

传统的 DBMS 也不是很可用。如果机器崩溃，在机器修复并且所有 DBMS 恢复完成之前，DBMS 将无法处理用户请求。

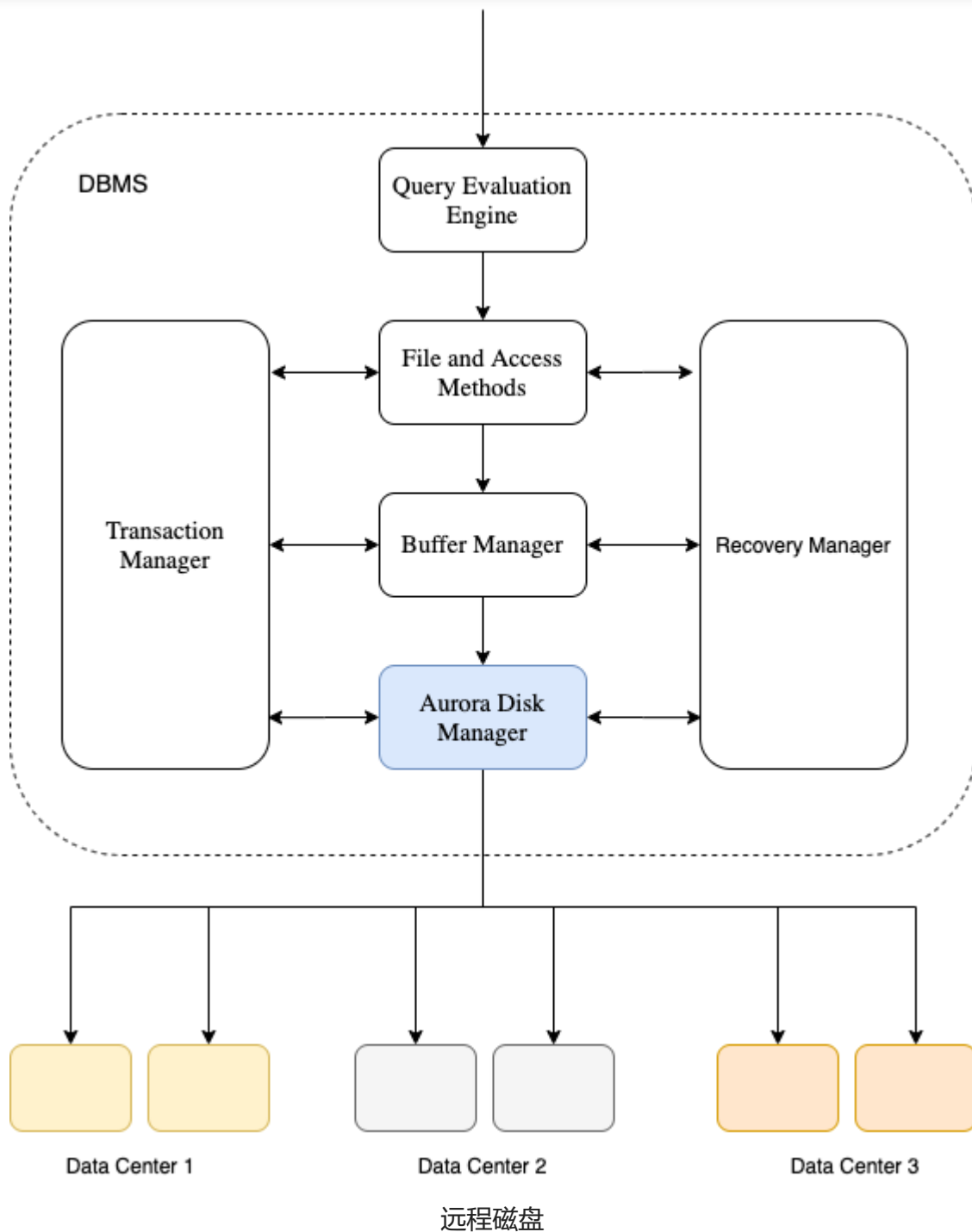
## Aurora 如何改进传统数据库

Aurora 数据库构建在传统 DBMS 之上。它重用了传统 DBMS 中的大部分组件，例如查询执行引擎、事务管理器和恢复管理器。（这就是为什么我们在上一节中花了一些时间在传统 DBMS 的架构上。）然而，它对传统 DBMS 进行了一些改进，以提高其可用性、可靠性和可扩展性。这些变化是

- 将数据复制到远程存储
- 仅将更改日志存储到远程磁盘
- 使用主副本设置

### 复制

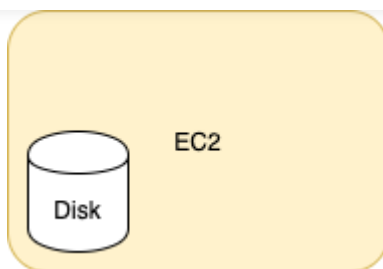
Aurora 做的第一件事是远程存储数据，而不是存储在本地磁盘上。如下图所示，Aurora 数据库扩展了磁盘管理器，使其与远程存储兼容。



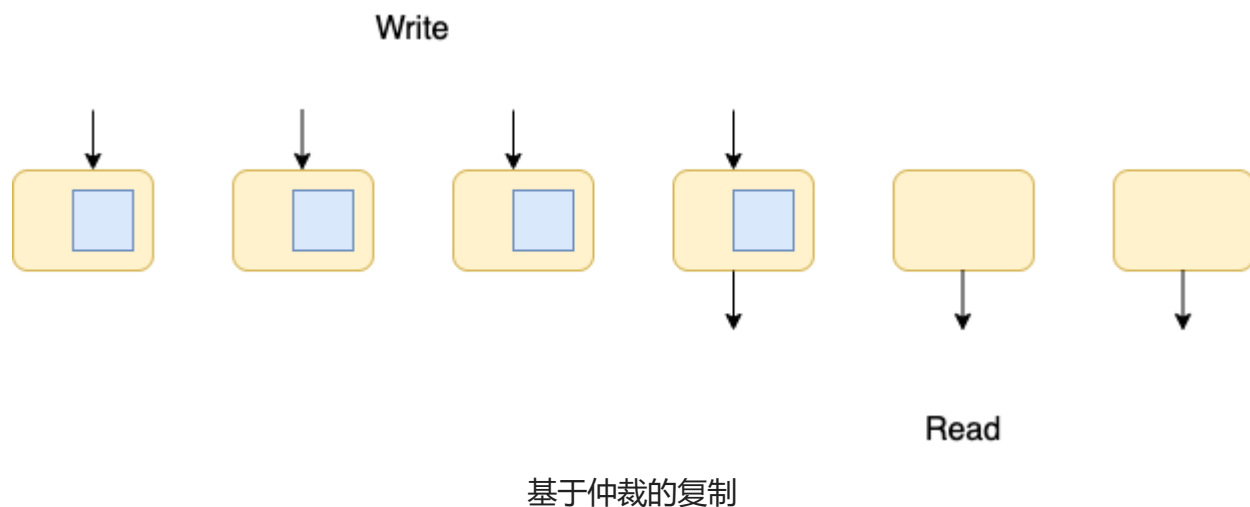
为了提高可靠性，Aurora 数据库复制数据。它通常在 3 个不同的数据中心复制数据 6 次。这种复制次数不太可能丢失用户数据。

Aurora 数据库使用 1 个虚拟机 (Amazon EC2) 来管理 1 个数据副本。数据存储在 EC2 实例的本地磁盘中。在我们的案例中，Aurora 数据库在 3 个不同的数据中心使用 6 个 EC2





然而，此设置的一个可能问题是磁盘管理器需要确保数据成功发送到 6 个不同的 EC2。如果任何 EC2 缓慢或忙于处理其他请求，则延迟可能会增加。对于 6 个实例，其中一个实例速度较慢的可能性更大，因此延迟更有可能增加。为了解决这个问题，Aurora 数据库只需要在返回给用户应用程序之前接收来自 EC2 子集的响应。我们将其表示为 EC2 的  $V_w$  数量。读取操作也是如此，Aurora 数据库仅从 EC2 的子集  $V_r$  中获取数据。只要  $V_w + V_r > \text{复制总数}$ ，我们就可以保证读操作会看到之前写操作的结果。如下图所示，保证至少有一台机器已经看到写入并为读取提供服务。 $V_w = 4$ ； $V_r = 3$ 。这是 Aurora 数据库中使用的基于仲裁的复制机制，有兴趣的用户可以参考本文了解更多信息。[[Amazon Aurora：高吞吐量云原生关系数据库的设计注意事项](#)。在 SIGMOD 2017]

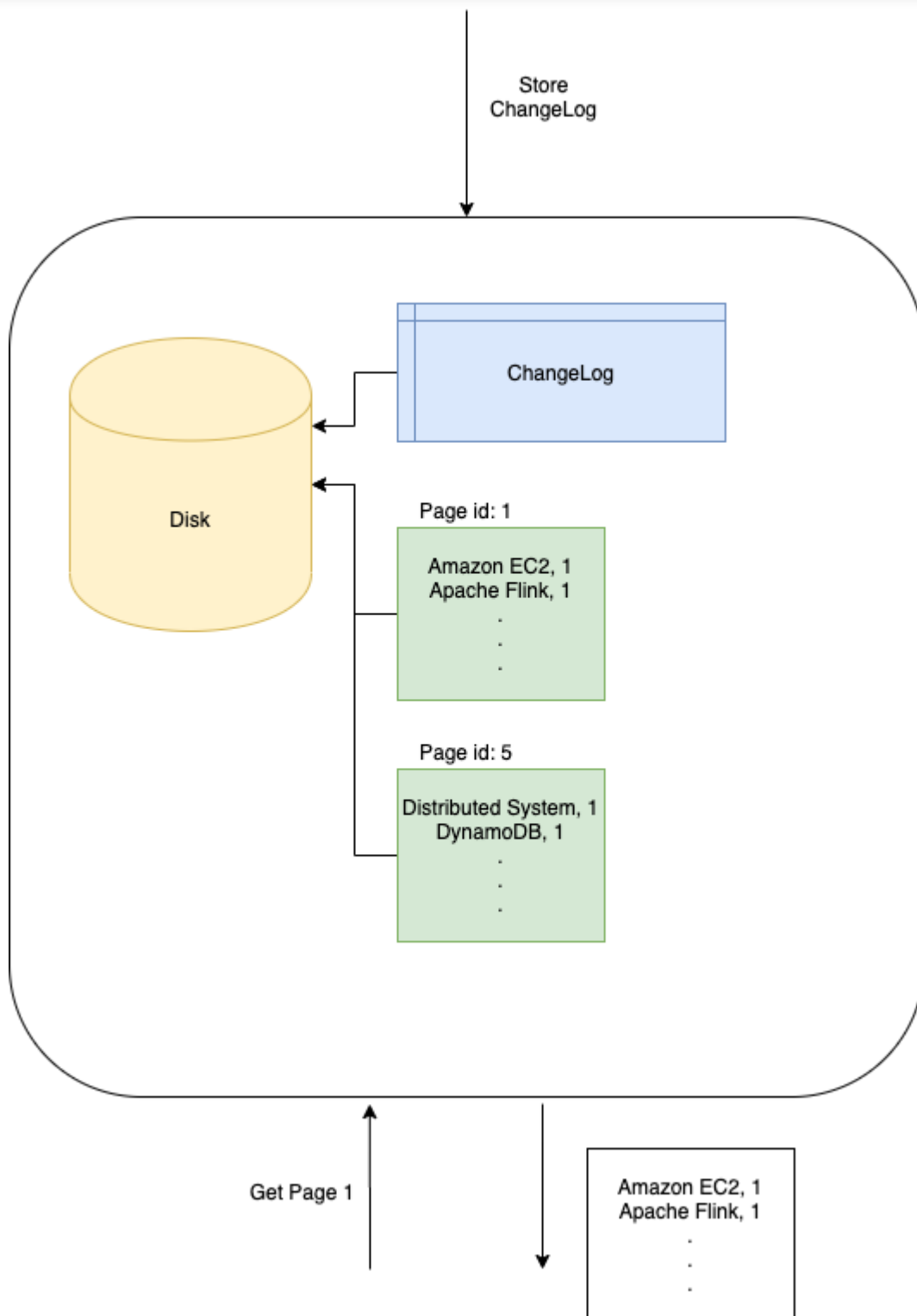


为了提高系统效率，Aurora 数据库仅从 1 个副本读取数据，因为它知道哪个副本拥有数据。这些是它收到响应的 EC2 实例。

- 变更日志作为数据

Aurora 数据库进一步提高了系统效率。它仅将更改日志存储到远程存储。在我们的写入





对于读取操作，它的行为与传统 DBMS 相同。如果该页面在缓冲区中不可用，则磁盘管理器向 EC2 发送请求，EC2 将该页面返回给磁盘管理器。







为了使系统更具可扩展性，Aurora 数据库支持主副本配置，如下图所示。[下图来自 [Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases](#)。在 SIGMOD 2017]



主副本配置

主实例可以同时处理读取和写入请求。更改日志被发送到 3 个不同数据中心的所有 6 个 EC2 实例。

副本实例仅服务于读取请求。当页面有更新时，主实例向副本实例发送通知，通知它该页面已过时。如果该页面在副本实例的缓冲区内，它将从缓冲区内逐出该页面。当副本实例收到读取请求时，如果缓冲区内不存在该页面，它将向 EC2 实例发送请求以获取该页面。

借助远程存储，Aurora 数据库支持多个实例并行运行，而无需在它们之间进行大量协调。通过主副本设置，Aurora 数据库能够支持比传统 DBMS 更高的吞吐量。

