**如何将一个大规模并行的Postgres数据库移植到Kubernetes上**

译者：李伦 校对：朱君鹏

扩展的分布式应用数据库（如Greenplum）非常适合Kubernetes

发布于 星期四，2019年3月14日

分类：Greenplum Kubernetes PostgreSQL 数据库

**作者简介**

**Oz Basarir**，Pivotal技术产品经理

**译者简介**

**李伦**，中兴通讯上海研发中心虚拟化架构师

如果您曾经想知道哪种类型的应用程序是在Kubernetes上运行的最佳候选者，那么扩展的分布式应用程序（如Greenplum）肯定是我们发现的最重要，最适合的应用程序。

我们的项目始于2017年底，目的是调查Greenplum如何从容器和Kubernetes等容器管理平台中获益。此时，Greenplum已经有了使用UDF（用户定义的函数）的SQL查询触发容器（请参阅PL / Container）。这些容器过去和现在仍然可用于以更精细和精确的方式隔离功能和管理资源。我们的下一步是将所有Greenplum转移到容器中。这是一种不自然的行为，一个可以被视为“提升和切换”遗留应用程序的合适解决方案，还是两种类似手套增强技术的结合？

在我们的探索中，首先，我们将Greenplum打包在一个容器中，以便在该单个容器中运行它。它当然不是一个可行的解决方案，而是朝着正确的方向迈出的一步，可以测试并了解隔离，资源管理和存储在这种环境中的工作方式。这里需要注意的一点是，容器不是一种在执行代码和硬件之间引入另一层的新技术。相反，容器只使用基本的Linux内核构造，例如过去四十年中已添加到内核中的cgroup，namespaces和chroot。

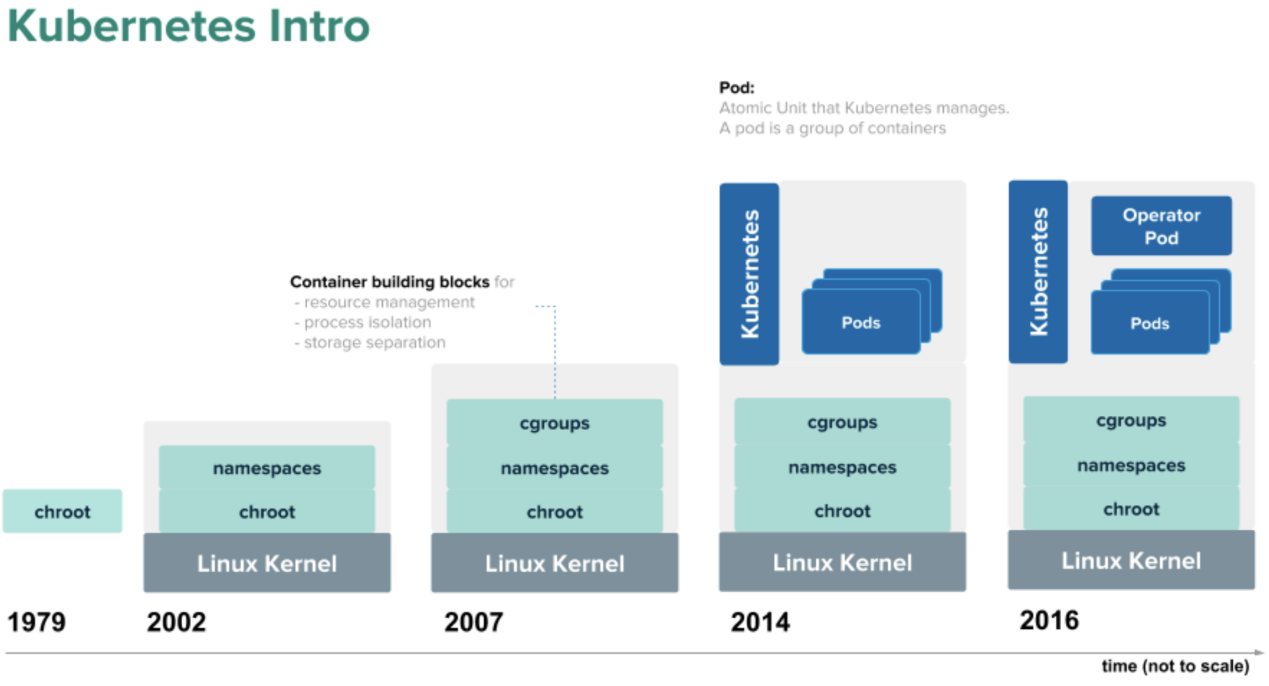


图1 Kubernetes介绍

在运行时，在容器中运行的应用程序代码与不在容器内的任何代码片段一样接近硬件。这不应该与用于打包容器以使其可移植的映像层所创建的抽象相混淆，因为这不是运行时关注的问题。因此，您仍然可以从容器内执行的代码中获得裸机性能。而且，如果您希望在虚机中分层以获得虚机提供的好处，那么这也是可能的，并且您的代码可以在容器内的虚机上高效运行，就像在虚机上直接运行一样。

当我们在单个容器中运行Greenplum后，我们意识到这不是容器中Greenplum的理想映射。由于Greenplum是一个MPP（大规模并行处理Postgres）系统，我们至少需要将集群分布在许多容器中，并以某种方式使这些容器相互了解并协同工作。这就是应该使用Kubernetes的时候。

Kubernetes是开源容器管理平台，它允许我们将Greenplum从单个容器中分离出来并将其作为真正的分布式横向扩展数据库运行。Kubernetes安排称为pods的容器组。一个pod可以有一个或多个可以共享存储的容器。在我们的示例中，存储设置为PV（持久卷），并且这些PV可以是本地的或远程的，具体取决于用户选择的存储类型。

同样，组成pod的容器只是使用Linux内核创建的运行时构造。Kubernetes通过为每个pod运行的节点选择一个工作节点，通过其调度程序来协调pod。Kubernetes集群还具有调度程序和其他管理功能所在的主节点。我们通过指定（反）关联性规则和节点选择器来帮助Kubernetes做出调度决策，这些规则和节点选择器是这种规则的简化形式。如果将来我们需要更复杂的逻辑，Kubernetes还允许我们定义自己的自定义调度程序。

对于我们的容器，我们还定义了CPU和内存资源限制，称为Kubernetes中的请求和限制。然后，工作节点上的可用资源由Linux内核功能cgroups处理。

Kubernetes为Greenplum提供了一个很好的平台，允许我们在计算和存储之间创建明确的界限，让我们通过精确控制在集群中分配计算，并为我们提供资源管理的旋钮。

反过来，Greenplum成为了Kubernetes的一个大租户，因为它在其集群中分发用户数据，并且并行处理这些数据的SQL和PL查询。让我们快速浏览一下Greenplum的架构。它有一个主设备，一个备用主设备，一些主要GP segment和它们的镜像段。其中每个都是Postgres数据库实例，还有许多用于AI /机器学习，图形，文本分析等的库和工具。

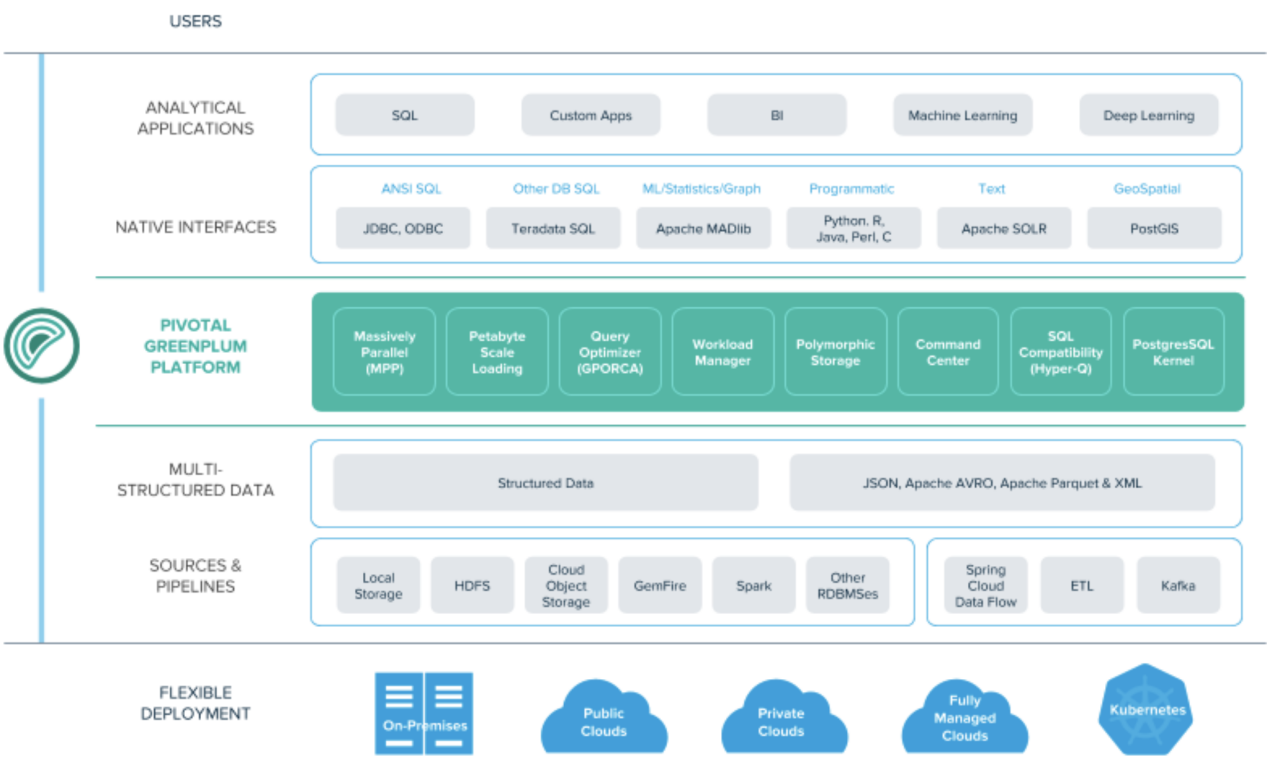


图2 Greenplum

收到查询后，主数据库会创建一个计划，在主要GP segment之间分配执行，从而允许Greenplum在很短的时间内返回结果。主数据GP segment越多，数据并行化程度越高，分析速度越快。

借助Greenplum的这种分布式横向扩展架构，当需要增加计算和/或存储资源时，我们可以通过简单地添加更多GP segment来实现。同样，在Kubernetes上，应用程序可以通过增加pod的数量来增加容量。这是Greenplum和Kubernetes的原子单位之间明显的一对一映射：GP segment映射到Kubernetes pod。

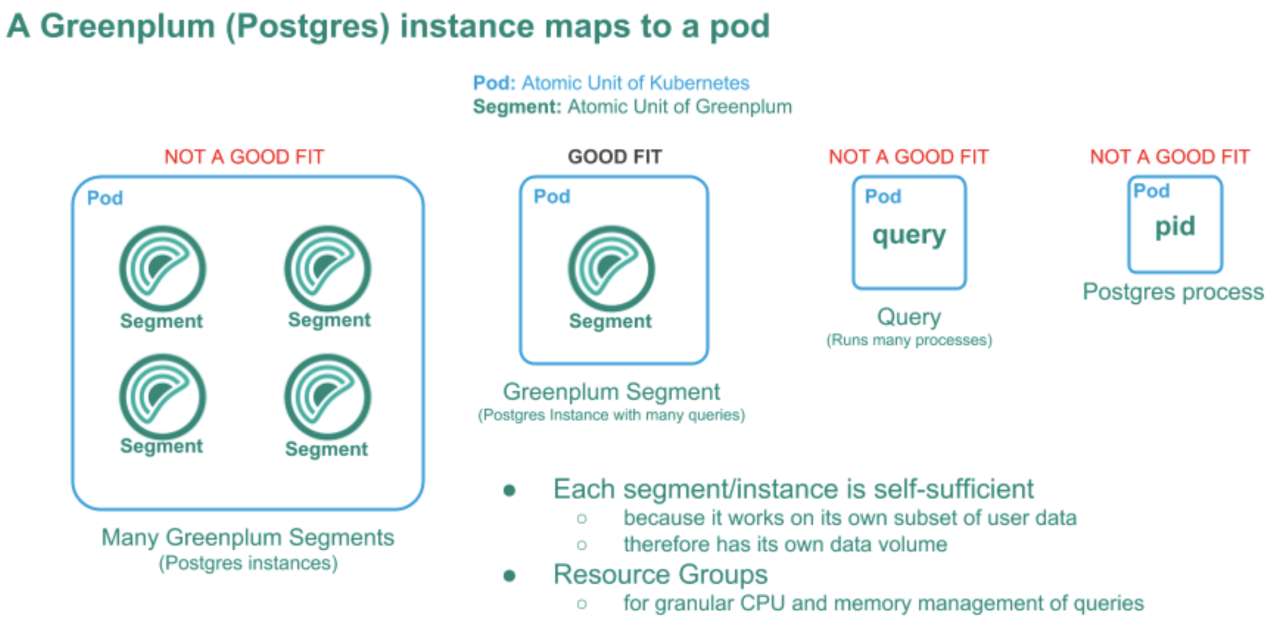


图3 将pod映射到GP segment

主要是在Greenplum和Kubernetes的分布式性质中的这种关系，这使得架构更加贴合。因此，看到这一点，我们能够将Greenplum从单个容器移到一个高度可用的，基于Linux内核的容器协调器上，并运行相同的Greenplum数据库以提供相同的用户体验。事实证明，整体解决方案有许多额外的好处。

其中一个好处是运营商模式，这是Kubernetes允许添加自定义逻辑来控制应用程序的方式。这意味着我们可以建立一个Greenplum运营商，一个Kubernetes的一等公民，并使用这个运营商自动化第1天和第2天的操作，如部署，故障转移，扩展，升级和修补，不仅仅是Greenplum，而是整套相关工具和组件。

我们应该注意到，自动升级和修补并不意味着零停机时间滚动升级; 相反，它意味着在计划的维护窗口期间，操作人员可以通过简单地触发Greenplum操作员来执行升级，以采取所有必要的步骤，保留过程中数据库的状态和数据。

另一个很大的好处是我们可以在CI管道中处理应用程序和库依赖关系管理，以便客户不必这样做。Greenplum及其丰富的生态系统现在可以进行安全性，配置，集成，网络和依赖性测试，然后在一个软件包中发布，准备创建一个新的Greenplum工作台或从现有的工作台升级。当然，用户仍然可以自定义要启用的组件和方式。

但有一点需要注意。性能仅与底层硬件一样好。与裸机部署一样，我们的硬件需要为pod提供足够的CPU，内存，磁盘IO和网络IO。Kubernetes无法神奇地让Greenplum飞起来。为Kubernetes构建正确的基础架构，特别是如果我们要在其上运行数据库，并不简单。这个领域对我们和巨大的Kubernetes社区来说仍然是一个挑战。我们已经看到在生产环境中部署无状态应用程序，但数据库对于这个场景来说相对还是比较新的挑战。

然而，我们的调查将我们带到了一个地方，在这里我们看到了在Kubernetes上运行像Greenplum这样的MPP（大规模并行处理Postgres）数据库的巨大机会和好处。我们评估了这是否是一种不自然的行为，一种可以被视为“提升和切换”遗留应用程序的可行解决方案，或者将两种技术结合在一起，就像手套一样。而且，我们坚信，大规模并行的Postgres数据库和Kubernetes就像带手套一样方便适合。

有关此主题的更多信息，请参阅我们团队的Pivotal Engineering Journal文章：

* 使用运营商模式管理有状态应用; 编排注意事项 （<https://pivotal-cf-blog-staging.cfapps.io/post/managing-stateful-apps/>）
* 存储超过容器或群集的有状态数据; 优化本地卷

（https://pivotal-cf-blog-staging.cfapps.io/post/storing-stateful-data/）

* 提供有状态的Kubernetes容器，努力工作并保持活力

（https://pivotal-cf-blog-staging.cfapps.io/post/provisioning-stateful-kube-containers/）

* Kubernetes中的有状态应用程序

（https://pivotal-cf-blog-staging.cfapps.io/post/stateful-apps-toc/）

* Greenbeum在Kubernetes运行

（https://pivotal-cf-blog-staging.cfapps.io/post/greenplum-for-kubernetes-operator/）

* 使用Minikube开发Kubernetes的Greenplum

（<https://pivotal-cf-blog-staging.cfapps.io/post/rapid-development-using-minikube/>）

原文：<http://engineering.pivotal.io/post/how_we_moved_a_massively_parallel_postgres_database_onto_kubernetes/>