绑核

概念：

在linux中，所谓绑核，其实就是设定某个进程/线程与某个CPU核的亲和力（affinity）。设定以后，Linux调度器就会让这个进程/线程只在所绑定的核上面去运行。使用cpuset命令，还可以实现某个进程/线程独占某个核。

k8s cpu管理策略

默认情况下，kubelet 使用 [CFS 配额](https://en.wikipedia.org/wiki/Completely_Fair_Scheduler) 来执行 Pod 的 CPU 约束。 当节点上运行了很多 CPU 密集的 Pod 时，工作负载可能会迁移到不同的 CPU 核， 这取决于调度时 Pod 是否被扼制，以及哪些 CPU 核是可用的。 许多工作负载对这种迁移不敏感，因此无需任何干预即可正常工作。然而，有些工作负载的性能明显地受到 CPU 缓存亲和性以及调度延迟的影响。 对此，kubelet 提供了可选的 CPU 管理策略，来确定节点上的一些分配偏好。

none 策略

默认的cpu管理策略，显式地启用现有的默认 CPU 亲和方案，不提供操作系统调度器默认行为之外的亲和性策略。 通过 CFS 配额来实现CPU 使用限制。

static 策略

static 策略针对具有整数型 CPU requests 的 Guaranteed Pod ，它允许该类 Pod 中的容器访问节点上的独占 CPU 资源。这种独占性是使用 [cpuset cgroup 控制器](https://www.kernel.org/doc/Documentation/cgroup-v1/cpusets.txt) 来实现的。通过此策略可以实现pod绑核效果。

前置条件：

Kubelet添加启动参数--cpu-manager-policy或kubelet配置文件添加字段cpuManagerPolicy

开启方法：

Pod 设置成 Guaranteed QoS 类型， requests 与 limit值相等，cpu资源为整数值。

关于 QoS 类为 Guaranteed 的 Pod有如下条件：

* Pod 中的每个容器都必须指定内存限制和内存请求。
* 对于 Pod 中的每个容器，内存限制必须等于内存请求。
* Pod 中的每个容器都必须指定 CPU 限制和 CPU 请求。
* 对于 Pod 中的每个容器，CPU 限制必须等于 CPU 请求。

测试