SPDF开发

1. spdk 是一个统一的存储开发工具包。这个工具包里面包含了最新的nvme 和PM（persistent memory， 就是一种掉电以后内存数据仍然保存的内存条， 里面有一个电池供应电，功耗非常的低）软件开发代码， 也包罗了老一代的存储设备例如 scsi rbd 等等， 里面也包括了null 块设备（这里我的理解就是一个黑洞设备， 可以达到一个天文数字的存储容量，但实际上根本没有现实中存储介质可以对应。），也包括的存储方面的算法库，例如EC的算法（这个是用intel 向量汇编代码实现。我们用不着）。再就是里面包含一些测试的代码和 配置文件，编译文件等等软件工程方面的辅助性文件。 其中我们要将来使用的是nvme的块设备。
2. 进入我们一直在做的东西 nvme的存储。 nvme的存储将来会替代当前的scsi的存储以及scis延申出来的各种存储设备。nvme的硬盘包括了 nvme over fabric ，nvme over tcp and nvme over pcie。 其中前2者是基于网络方面的nvme 协议。我们目前也用不到。 nvme over fabric/ tcp 将来可以用来扩充单个存储服务器的存储容量的时候，我们可以通过光纤40G连接到存储硬盘的盒子（类似JBOD，这个是一个铁壳，里面密密麻麻存储大约上百的硬盘。 如果按照一个硬盘32T计算， 这个大铁壳子可以有30PB的容量）。
3. 我们现在要做的第一期就是 nvme over pcie 。这是一种你们所能看到的服务器实物上面插入的硬盘，这中硬盘是通过pcie 4.0 或者最新的pcie5.0 传输速度可以达到1T每秒的高速总线协议。你们是学电子方面的，对于这种电子方面的东西应该不陌生。 不过市面上所能购买的成熟的pcie nevm 硬盘可能是pcie3.0 这个速度也很快的。具体的参数，你们可以去google一下或者让你们的胡老师跟你们大体普及一下。 我这里就不画图了。
4. 其实就跟你们目前使用的台式机一样。就是台式机器后面挂了24或48个或96个这样的nvme硬盘。一般情况下nvme的企业硬盘都是32T 和16T的。 最新的nvme 硬盘技术可以达到128T 甚至256T，当然这还在实验室阶段还没有在市场上销售。这样你们就有个直观的想象。当然你们可以自己画个图，然后我看一下，你们想象是否符合存储服务器的实物架构。
5. 好了，基于这样的一个存储系统 ：我们有5个节点。 每个节点是一个存储服务器。每个节点含有24个32T的nvme硬盘。 每个节点有32个cpu核 还有2个10G的网卡。内存一般在256GB/512GB。存储服务器就是比一般的服务器大气上档次。 我们选择其中的3个节点运行consul server agent ，另外的2个运行consul client agent然后这5个节点都会安装 spdk 的nvme bdev lib库。这样以来， nvme的bdev在每个节点上都会安装（共5个节点）。
6. 有了bdev的库安装在任何一个节点上了。 我们就可以取任何一个节点，运行bdevperf程序。 那个这个bdevperf的程序就是我们即将要对他进行修改的程序代码。由于bdevperf 是基于bdev 总体读写接口API下面开发的iO性能工具，我们就用malloc bdev设备来模拟（当前你们正在做的就是这个）。 其实关于bdev，如果你们了解对象编程:类和子类 ,抽象接口和实例，就会知道 bdev是一个抽象结构，是一种父类,similarly，一种对外的公共接口。那么 nvme nvmeovertcp malloc rbd iscsi 等等都是他的子类，必须实现bdev提供的接口。这里的每种设备（nvme ，nvmeovertcp， malloc， rbd ，iscsi等）都有自己的实现spdk代码。

参考资料：https://spdk.io/doc

<https://github.com/ecnuhulab/spdk>

1. git hub 二进制编译spdk

按照<https://github.com/ecnuhulab/spdk>步骤

执行Source Code（改 git submodule update –init depth 1）, Prerequisites, Build, Unit Tests, Hugepages and Device Binding（只执行setup.sh）

2、进入spdk文件夹

Vscode打开./test/bdev/bdevperf/bdevperf 中的conf.json 文件可更改num\_blocks（块存储数量）以及block\_size（块大小，单位字节）

执行 sudo ./test/bdev/bdevperf/bdevperf -q 32 -s 1024 -w randwrite -t 10 -o 4096 -m 0x3 --json test/bdev/bdevperf/conf.json

//模拟器分配的是双核则使用0x3，若是4核使用0xf

进入./test/bdev/bdevperf文件夹，执行./bdevperf --json conf.json可查看注释