5119510006 余阳

做了什么？

这学期本来可以说做了开源软件的贡献的，但是合作公司把项目变成private了。 但是这学期没做什么东西就只好拿着个凑数了。 之后用XXX代表这个合作的原型项目。

背景

DBS以SAAS云服务的形态为数据库用户提供了多种备份能力，但这些能力要么基于数据库自身的特性、要么基于OSS存储特性，硬核的东西比较少。 这样，在产品活下来后，构建技术壁垒就是当前DBS迫切需要解决的问题。

XXX的目标

存储的核心是什么？一定是围绕RPO、RTP、和存储的成本来构建的。PRO部分DBS使用WAL日志拉取方式已经可以达到秒级，而NearStore将聚焦在低成本、RTO这两方面，希望在数据库这个细分的备份场景下，构建DBS的核心能力。

XXX为DBSNode、DBSAgent提供文件读写接口，而在NearStore内部进行文件的再组织，最后将组织好的数据写入到OSS、Nas、或Pangu之上。 这里的再组织过程，就是实现存储低成本、数据快速恢复（低RTO）的过程。

XXX备份数据的低成本存储

NearStore使用重复数据删除（Deduplication)和数据压缩技术，以最大化的压缩存储成本。重复数据删除旨在达到全局（所有备份数据）的单副本，而数据压缩则进一步的把单副本数据局部最小化。

XXX快速数据恢复

XXX提供了文件读、写接口，同时也需要提供多文件的快照接口，用以支持快照操作。而快照操作即是NearStore进行快速数据恢复的核心能力。

XXX的形态

首先，XXX不应该是存储系统，而更多是要依赖当前阿里云上的存储系统去实现其目标；其次，XXX要尽可能的轻量化；最后，XXX的所需的资源应该是按需分配和占用的，不需要是能达到0资源占用的状态。基于这样的考虑，XXX的核心以lib库的方式提供给使用者。DBS Node使用这个库完成备份数据的读写，重删进程使用这个库进行重复数据的再组织，DBS Agent使用这个库完成快照数据的读取和少量写入。

入手项目

读论文

读研之前看过的论文也就是ioi国家集训队论文，那个信息量非常大，要看的非常仔细。但是学术论文废话是真的多，所以开始不太习惯，看的非常慢，后来感觉都差不多就主要看不同的部分。

在这个阶段主要对Deduolication有个大致了解，但是相关开发经验的不够让我对这个东西在系统中大概定位在哪并不是很清晰。

看项目

感谢前人的付出，Deduplication的结构基本被玩烂了，大家都是在一些地方替换自己的算法和数据结构看能不能达到更好的效果。 代码结构大同小异，很快就能开始改。

改项目

因为是原型项目，我们主要是在学校的服务器上做实验，首先算出大概哪个模块可能是性能瓶颈，然后就需要动手改了。 因为最终的产品的存储是不在本地的，最终存储是依赖于对象存储服务。为了模拟这个环境，我实现了一套POSIX类似的RPC服务，然后给项目写了一个单独的模块来掩盖这些存储环境的差异。 后来合作公司提供了OSS环境，我也在修改rocksdb把文件创建在OSS上来用作索引。

有何改变

背景

我大学时候本来是想做游戏客户端开发的，并以游戏开发论坛中的大佬们为榜样，对这条路也是觉得一定能走下去。但是最后只学习了一些皮毛，或者没有找到重点，与其学什么东西用了什么牛逼的算法进行实时渲染提高了什么，不如去熟悉主流游戏引擎，因为牛逼的效果基本可以调库实现的，再不济也能购买他人做出来的效果。 后来有一家公司雇佣我，没让我去做客户端，而是给技术美术打杂，但是很幸运的是离我工位很近的有YDWE的作者，虽然大佬不认识我。 后来就是著名的停发版号的事情，我失业了，大佬也失业了，只不过大佬现在去做帝国时代3了。 而我就到了这个学校想改变一下技术方向，因为我研究了以下游戏服务端比客户端工资高百分之50,当然技术要求也要高一些。

项目学到了什么

主要是把linux下面的系统编程和网络编程又重新捡了起来，当然老毛病还是犯了，就是总是放着成熟的实现不用而是总要自己实现一套，所以即使付出了不小的努力还是开发进度缓慢。 因为每个月都要汇报，我还学会了简单的矢量图和ppt制作，和不同背景的人讲技术应该讲到什么程度。看论文的过程中学会了把简单道理讲复杂，把关键问题糊弄过去。

课程反馈

大部分人给某个项目贡献代码并不是为了这么做才去这么做，基本上本身就是某个软件的深度使用者。 我第一次用git是因为某个游戏开活动但是对应辅助工具资料更新太慢，我自己认为我对这个游戏已经有深入的了解了，但是研究代码的过程中才发现自己并不是骨灰玩家，对游戏机制了解不够。不是一个好心华人老哥给我耐心讲解，我第一次用git基本就失败了，所以我认为融入社区最关键的是两个，一个是对某个东西特别了解，一个是有人推荐你进去。 专业知识和代码能力真的不重要，因为很多开源软件代码都是一坨，改到能用就行。 这门课放大家去选项目加入是非常不负责任的，因为选项目加入是结果不是动机，比如，我曾经的动机是游戏开活动了， 而我至少高强度玩了一年还有没懂的。 所以我建议的是模拟这个多人多地点开发软件的流程，所以这个软件应该是大家都能会做的，然后在开发者水平残次不其的情况下来控制软件的质量，到最后从开发和管理两个角度来说各自都做了什么。