大家好，我们的项目名称为 GraND Pro，全称是 Graph Network Disk with Prometheus，带有监控模块的图结构网盘

我们的项目基于 2021 年的 x-DisGraFS 展开，DisGraFS 是一个分布式图文件系统，我们改进了他们的存储设计，搭建起完全意义上的存储集群，并且简化用户的操作。此外，我们致力于设计一套带有监控模块的可运维系统，通过监控，运维工作者能够实时掌握系统的运行状态，保证服务正确稳定地运行，并且可以远程执行管理员操作。

以下为目录，我将从 what、why、how 三个方面来讲

首先我将简单介绍一下 DisGraFS，然后是我们实现分布式存储过程中参考的另一个项目 dontpanic，他们实现了比较高效的纠删码操作，最后我会介绍一下分布式监控

首先，DisGraFS 是一个分布式图文件系统，它将图结构的思想应用到分布式文件系统上面，使之兼具图文件系统方便用户快速搜索、模糊搜索、查找相关文件的特点，以及分布式文件系统的海量文件存储、云存储的特点。这个项目的计算集群使用 Ray，Ray 是一个高性能分布式计算框架，存储使用的是 JuicsFS，这是一个共享文件系统。

dontpanic 我们参考到了他们的纠删码操作，纠删码（Erasure Code）是一种编码技术。它通过计算将 n 份原始数据增加至 n+m 份数据，并能由其中的任意 n 份数据还原出原始数据，即可以容忍不多于 m 份的数据失效。纠删码操作用以提高存储系统的可靠性，相比多副本复制而言，它能以更小的数据冗余度获得更高数据可靠性， 但编码方式较复杂，需要大量计算。

dontpanic 采用纠删码技术中的里德-所罗门算法对文件进行冗余，他们编写了 Go 语言代码并编译成 WebAssembly 格式，用于网页端实现文件切片和编解码。

然后是分布式监控，分布式监控是部署在分布式系统内的监控组件，它可以监视和显示集群中各节点的状态信息，它有运行在各个节点的进程，可以采集不同节点之间的通信消息，采集各个节点的资源利用率，最后将采集到的数据汇总到一个数据库，进行分析处理后以直观的图形化界面进行呈现。

我们对运维提出了更高的要求，不仅是被动地接收监控信息，还应该可以远程地操控。

然后是 Why，我们项目的意义

首先，DisGraFS 的存储是使用的 JuiceFS，这是一个共享文件系统，他需要用户先进行一些安装配置，然后手动将 JuiceFS 挂载到本地，然后才可以上传和下载文件。所以，DisGraFS 的远程存储集群实际并没有实现，它是一个存储节点，而不是集群，而且手动挂载也增加了用户的操作难度，我们的项目实现了远程存储集群，用户不需要在网页端以外进行任何配置。其二是可用性问题，没有文件冗余操作的话，一旦存储节点失效，所有文件都会丢失，无法下载。最后一点也就是之前提到的运维性

（换页），也就是监控，监控可以让运维工作者实时掌握系统各个模块的状态，及时发现问题解决问题。

所以我们的项目最初的设想就是改进 DisGraFS 这四点不足。当然，在具体实现过程中我们还添加了其他一些内容。

然后是 How，我们的具体实现过程。首先我将介绍一下我们的系统架构，介绍一下一下我们各个模块的设计实现，我们有6个模块。不过，分布式系统既要包括是怎么把一个完整的系统拆分成不同模块，也要包括不同模块之间是怎样相互协调配合，共同实现预期功能的。所以我将分别从用户操作和运维操作的角度，介绍一下六个模块之间是如何协作配合。

以下都只是介绍原理，不会具体到代码细节，大家如果对某个代码细节感兴趣的话，我们的源码都在仓库中，也欢迎和我们讨论

首先是系统架构，共有六个模块。

- 索引服务器：与存储集群交互，获取存储节点的状态信息
- 图数据库：维护存储文件的图结构
- 分布式存储集群：文件切片后的存储位置
- 分布式计算集群：由 Ray 支持的打标计算集群
- 分布式监控：监控各模块状态、资源占用情况，可远程唤醒与关闭存储节点
- 网页端：呈现给用户的界面，用户可进行文件或文件夹操作，查看图结构

与 DisGraFS 相比，减少了客户端，他们的这个客户端，就是要用户把存储的文件夹安装并挂载到本地，我们将这一过程省去。另外就是增加了监控模块。这个图数据库，DisGraFS 也是有的。我们继承了下来。

图数据库使用的是 neo4j，Neo4j 是一个高性能的 NOSQL 图形数据库。Neo4j基于其特殊的储存结构与 Cypher 查询语言，设计并优化了多种图上的算法，使得查询、插入、删除等图操作的效率大大提高。我们在 DisGraFS 基础上做了一些改进，在实现多用户时，一个用户应该只能看到自己文件的图结构，最初我们对这个有两个设想。添加了owner属性，标识该文件的所有者。在使用GraND Pro的过程中，用户的操作在图数据库里只涉及到修改自己的节点和关系。我们对前端页面查看同样做了隔离，使得所有

用户都只能看到自己所有的文件和标签，从而实现GraND Pro对多用户的支持。还有一些改进，包括删除、重命名、将同名不同扩展名的文件建立联系等。

然后是分布式计算集群，分布式计算集群要完成三个动作，接收文件、打标、发送打标信息。用户上传的文件，也会传一份给计算集群，计算集群检测到有新文件传来，就会调用打标操作，打标就是计算集群的“计算”内容，多台电脑共同承担这些计算任务，这些任务的分发和负载均衡是依靠 Ray 来实现，Ray 是一个高性能分布式计算框架，它只需要将多台电脑构建起一个 Ray 集群即可。Ray 的启动需要先启动一个 head 节点。

这里我们用到了另外一个技术。Ray 集群的搭建需要各台电脑在同一个局域网中，而 Ray 的 head 节点需要保持运行，才能保证其他节点随时可以连接或断开。所以我们把 head 节点部署在一台服务器上，worker 节点目前位于我们本地电脑的虚拟机。局域网的实现是通过搭建 vpn，将本地接到服务器的内网。

监控模块，本项目采用Prometheus+influxDB+Grafana进行了对存储节点、计算节点的指标监控。

Prometheus 是一个开源系统监控和警报工具包，它将实时的指标数据（metrics）记录并存储在通过 Http 拉取模型构建的时间序列数据库中，有着较灵活的询问功能和实时告警功能。

Prometheus Server 本身就是一个时序数据库，将采集到的监控数据按照时间序列的方式存储在本地磁盘当中。但是本地存储也意味着 Prometheus 无法持久化数据，无法存储大量历史数据，同时也无法灵活扩展和迁移。InfluxDB 是一个由 InfluxData 开发的开源时序型数据库，着力于高性能地查询与存储时序型数据，在DB-Engines Ranking时序型数据库排行榜上排名第一，广泛应用于DevOps监控、IoT监控、实时分析等场景。

Grafana是一个跨平台的开源的度量分析和可视化工具，可以通过将采集的数据查询然后可视化的展示，并及时通知。

图形化展示。

运维对Storage节点的远程操作

远程唤醒，远程终止。用 ssh 建立运维服务器和 storage 的连接后实现。

网页端，

增加网页端功能：增加了文件删除、重命名和共享。因为采用了 mysql 存储文件元信息，图数据库存储图结构信息，以及存储集群存储文件碎片，所以这三个操作的实现都要考虑多方面内容。以共享为例，文件可以由一个用户共享给指定的用户，被分享的用户可以看到文件，并可以下载和删除共享文件。所有共享文件共用同一批文件碎片，共享者与被共享者对共享文件的操作的结果不同，被分享者删除文件时，只是在他的界面不会再显示这个文件，实际的文件碎片不会删除，而当文件的最初所有者选择删除文件时，文件碎片会被删除，所有共享出去的文件也会随之删除。

网页端功能还增加了目录操作，包括目录新建、删除和重命名。

网页端引入了第二个图结构页面，第一个是原来 DiaGraFS 实现的，我们进行了保留。原来的图界面是用于知道标签之后有目的性地进行筛选的操作界面，不利于用户把握整个文件系统的图结构。这句话这样说可能有点抽象，一会演示时就能看到两个图结构界面的区别。

我们也做了前端美化功能，使用了开源框架 Hendrix

此外还有两个模块，索引服务器和存储节点。索引服务器的功能不多，主要是是获取 storage 的状态信息，这一点是通过与 storage 通讯实现的。而存储节点我们是复用了 dontpanic 的代码，在他们的基础上新增了删除文件碎片的操作，也即实现了网页端的删除按钮。碎片删除是一个远程操作，也是通过 ssh 完成。

Docker，Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。

使用 Docker 部署有以下的优势：

首先，Docker的使用简单至极，从开发的角度来看就是三步走：构建，运输，运行。其中关键步骤就是构建环节，即打包镜像文件。但是从发布和运维的角度来看，那就只有两步：复制，运行。有了这个镜像，那么想复制到哪运行都可以，完全和平台无关了。同时Docker这种容器技术隔离出了独立的运行空间，不需要考虑应用之间的相互影响。

在我们的六个模块中，计算机点和存储节点是可横向扩展的，所以我们实现了计算节点和存储节点的容器化部署。

以上就是六个模块的主要内容了。

然后就是用户操作，

接下来是运维操作，

接下来就是实机演示。

然后是监控模块的演示。

最后，我想说一下我们这个 GraDN Pro 项目和过去五年的一些 OSH 项目的联系。

时间线的源头是 2017年的 DFS，它实现了分布式网盘，以及用户管理功能，它的网盘是共享网盘，所有登陆上的用户用的是同一个网盘。 2020 年的 dontpanic 项目是在 DFS 基础上做的优化提升，dontpanic 改进了 DFS 的纠删码，实现了更高效率的编解码，增加了初步的用户隔离操作，文件传输不再经过中央节点，而是网页端与 storage 直接传输，2020年的另一个项目 GBDFS，是基于数据库文件系统，将数据库文件系统改进成为了图数据库文件系统。2021年的 DisGraFS 项目受到 DFS 和 GBDFS 启发，提出了分布式图文件系统，只是在具体实现时没有做到分布式存储，时间线来到了 2022 年，我们组基于 DisGraFS，实现了远程分布式存储集群，简化了用户操作，新增了一些网页端功能，引入了监控运维模块。此外，我们也了解到今年有一个组在做 DisGraFS 的高性能方向的优化，我们和这个组在前期部署 DisGraFS 时也曾相互帮助。通观这条时间线，我认为，这是一种传承，我们总是在前人的基础上不断改进优化、推陈出新，一步一步让这个系统更加丰富、更加完善。

最后，感谢刑凯老师对我们组提出的指导和建议，感谢所有帮助过我们的同学，也感谢大家的观看，谢谢