有关分布式系统的调研

概念及原理

分布式系统是一组电脑，透过网络相互连接传递消息与通信后并协调它们的行为而形成的系统。组件之间彼此进行交互以实现一个共同的目标。把需要进行大量计算的工程数据分割成小块，由多台计算机分别计算，再上传运算结果后，将结果统一合并得出数据结论的科学。（来自维基百科 分布式计算 - 维基百科，自由的百科全书 (wikipedia.org) ）

分布式操作系统（Distributed operating system），是一种软件，它是许多独立的，网络连接的，通讯的，并且物理上分离的计算节点的集合。每个节点包含全局总操作系统的一个特定的软件子集。每个软件子集是两个不同的服务置备的复合物。第一个服务是一个普遍存在的最小的内核，或微内核，直接控制该节点的硬件。第二个服务是协调节点的独立的和协同的活动系统管理组件的更高级别的集合。这些组件抽象微内核功能，和支持用户应用程序。（来自维基百科 分布式操作系统 - 维基百科，自由的百科全书 (wikipedia.org) ）

分布式系统的优缺点

参考：到底什么是分布式系统？你需要了解这些华为云官方博客-*CSDN*博客什么是分布式系统

优点

- 资源共享，节点与节点之间相互连接，资源共享很容易
- 可拓展性好，通过新增节点即可提升算力，无需像传统系统那样优化硬件性能
- 容错性高，单个节点的崩溃不会导致整个系统的瘫痪

缺点

- 安全性难以保证，节点的每次交互都需要通过网络，相当依赖安全性，在交互过程中也会存在数据丢失的风险
- 相较于单用户系统，分布式系统的数据库更加复杂
- 调度策略优化，如果多过节点同时发送数据，可能会导致网络过载

分布式系统的架构

- 客户端-服务器：服务器作为共享资源，客户端与服务器进行交互与数据交换
- 三层架构：把系统分为表现层、逻辑层和数据层
- 多层架构：将三层架构进行更详细的划分
- 点对点架构：没有专门的服务器提供服务，所有节点是对等的，既可以是服务器，又可以是用户端
- 以数据库为中心：各节点共享一个数据库，进行协同工作（个人感觉和“客户端-服务器架构”有点相似）

分布式操作系统的案例：**LegoOS**

原帖链接：OSDI 18| LegoOS: 分布式操作系统 - 知乎 (zhihu.com)

传统机器的缺点：CPU和内存在同一物理机上，资源利用率低；硬件弹性不够，加减硬件比较麻烦；容错性低，一个硬件坏了，整台机器都用不了。

分离式的内核（SplitKernel），分为三部分：计算节点、内存节点和存储节点，节点之间通过网络进行通信，如下图所示。

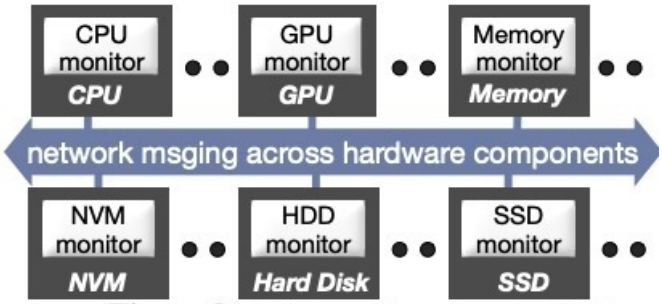


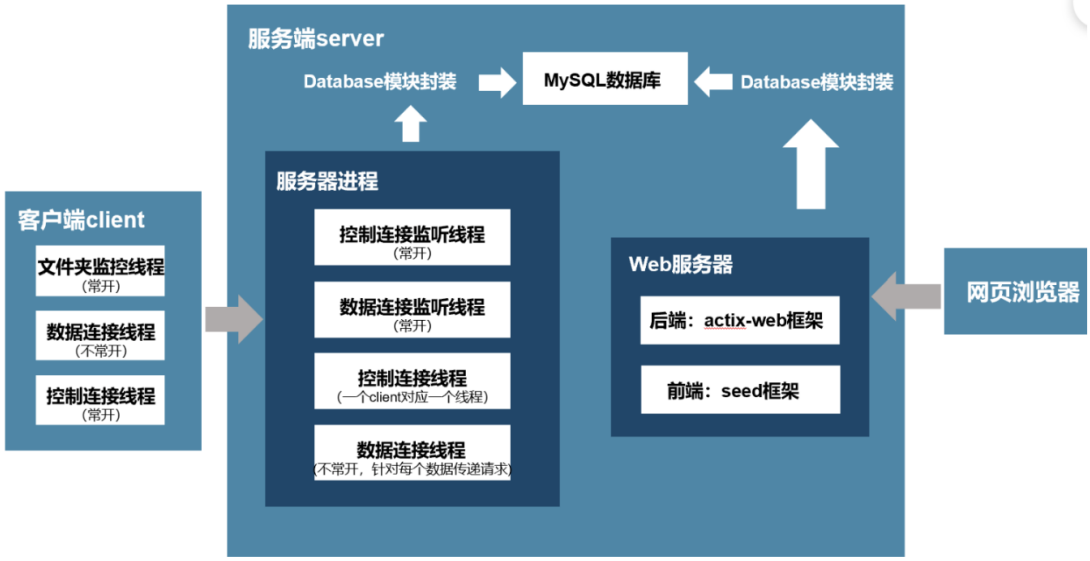
Figure 3: Splitkernel Architecture.

此外还涉及全局资源管理，对应分为三个GPM, GMM, GSM，它们主要做粗粒度的资源分配和负载均衡（均衡各个节点之间任务量的分配）。

对往年**OSH**分布式系统的分析

2020 x-gkd

项目框架如下，类似于客户端-服务器框架。



客户端程序运行在客户主机（存储端）上，服务端与 Web 服务程序运行在服务端上，通过网页浏览器访问该分布式文件系统，无需运行其他程序。

客户端

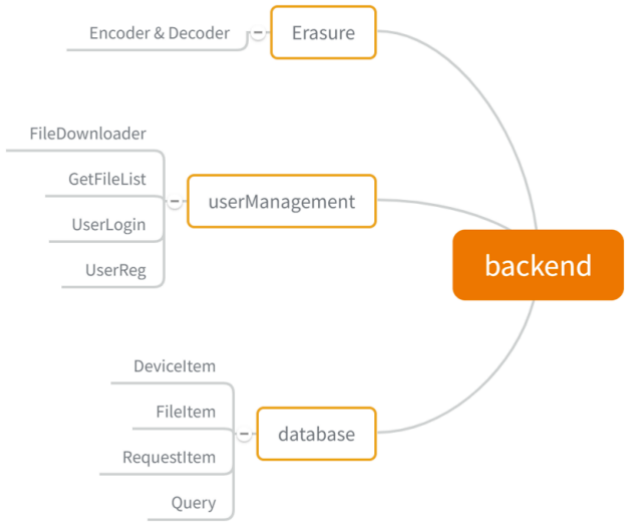
- 文件夹监控线程：用于每隔一段时间扫描一次共享文件夹，查看共享文件夹中是否有新文件需要上传至服务器。
- 控制连接线程：用于每隔5秒向服务器发送一次报文，确保客户端与服务器的长期连接。同时，该线程还负责处理服务器发回至客户端的控制报文。
- 数据连接线程：用于必要时与服务器交换数据。

服务器

- 控制连接监听线程：用于监听服务器的控制端口，若有客户端发来请求或连接，它将创建一个控制连接线程处理该连接。
- 控制连接线程：每个控制连接线程对应一个客户 端，该线程用于读取并处理客户端通过控制端口（对应客户端控制连接线程）发来的报文。
- 数据连接监听线程：该线程用于监听服务器的数据端 口，若有客户端发来请求/连接，其将创建一个数据连接线程处理该连接。
- 数据连接线程（不常开）：该线程用于读取并处理客户端通过 数据端口发来的报文。

该项目在web开发的特点：前后端分离（前端负责页面的显示和刷新，后端负责提供响应的数据）、采用json格式交换数据。

后端框架

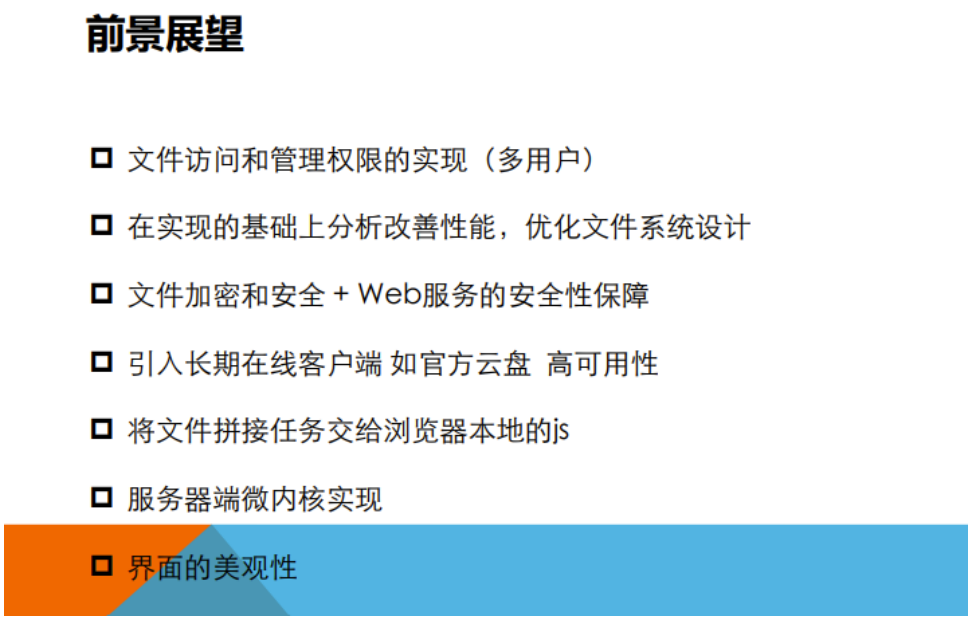


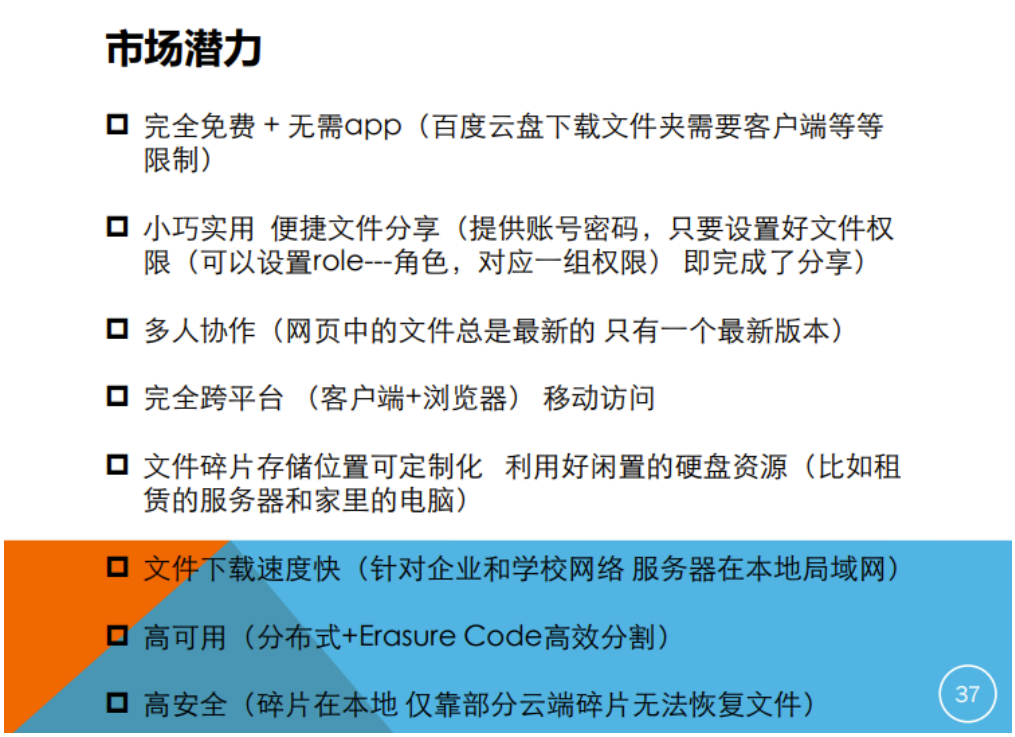
后端采用 Actix Web（小型、实用、快速的 Rust Web 框架 ）

前端

- Tomcat Web应用服务器（用于提供 Web 服务）
- Struts2 动态网站网站应用调度框架（Web 开发框架，似乎面临淘汰 Struts2 为什么被淘汰？自己作死！ - 知乎 (zhihu.com)）
- BOOTSTRAP 网页主题（前端框架，用于美化页面）
- JQery（ JavaScript 的库，便于实现AJAX，但是似乎也面临过时）+AJAX异步C/S通讯（前端通信方法，以网页局部刷新代替全部刷新） + JSON （数据格式）+MySql（数据库）

优势（来自该项目的原始项目IngramWang/DFS_OSH2017_USTC (github.com)）





个人的想法

待解决的问题：

- 如何实现分布式（部署到 Web 服务器还是在本地实现，抑或其他方法）
- 具体实现哪种分布式系统

分布式软件系统是支持分布式处理的软件系统，包括分布式操作系统、分布式程序设计语言及其编译(解释)系统、分布式文件系统和分布式数据库系统等。

~~会不会偏离操作系统的主题~~

一些思路：或许可以通过 LegoOS 的SplitKernel 寻找灵感，通过类似思路实现分布式操作系统，或者其中的一部分功能（如果做整个工作量太大的话）。