基于 IPFS 分布式存储的移动硬件研发方案

2023 年 06 月 29 日

一、项目背景和需求

IPFS 又称“星际文件系统”由协议实验室发布的一种点对点传输的分布式文件存储系统。点对点传输是一种硬盘共享的互联网底层协议，简单的说 IPFS 可以打造一个让更多的人将自己闲置的存储空间进行共享并获取收益的去中心化存储网络。我们将这种人人共享硬盘的存储方式称之为分布式存储。IPFS 分布式存储作为一项十分具有突破性的区块链技术，它能够很好地解决当下互联网数据存储瓶颈及数据存储安全问题，不但如此，IPFS 技术更能够为人工智能、大数据、物联网、分布式存储等提供底层支持，为其建立良好的信息基础。

本项目基于 IPFS 分布式存储，研发了一种移动的硬件。采用无线传感器网络技术（WSN）是物联网领域中的基础技术，其中无线传感器网络节点的部署是关键技术。常见的无线传感器网络技术包括 ZigBee、Lora、4G(5G)、WiFi 和蓝牙等，而其中 4G(5G)是将 WLAN 技术和 3G 通信技术进行了很好的结合，使图像的传输更快，让传输图像的质量和图像看起来更加清晰。在智能通信中应用 4G(5G)通信技术让用户的上网速度更加迅速，速度可以高达 100Mbps。所以 4G(5G)通信技术被广泛利用在智能家居，环境监测、自动化医疗和智慧城市等领域。

本项目主要以 4G(5G)通信技术及其组网技术为核心，组建一种远距离、低功耗和低成本的 IPFS 分布式存储系统。实现布式系统与集中式系统的不同之处。本项目有以下几个特性

数据跟踪：

在集中式存储网络中，只要信息通过单点服务器即单个中心点，便可以轻易跟踪数据流动路径，同时可以跟踪到数据本身。

但在分布式存储网络中，由于多个设备同时充当数据的存储服务器节点，跟踪数据流的难度呈指数级增长，在这种情况下，用户可以从多个位置访问给定的信息集。

隐私保护：

相较集中式存储网络，分布式存储网络可以提供更强的隐私性，在个人数据资产比重越来越大的现状之上，这是切身痛点。

在此网络环境下，个人在线数据行为不便跟踪，当然这也带来了一个潜在问题，即关于数据的网络伦理问题。

故障容错：

当数据在传播路径上处于单线通路时，集中式存储网络的单点故障带来的数据风险极大。

基于正在实施的备份系统，服务器托管的失败可能导致大量数据丢失，使人们难以在给定的时间访问数据。

相反，在分布式网络连接的情况下，单个接入点的故障永远不会使人们无法访问网络中存在的数据。

分布式存储网络中存在多个节点，使人们可以访问信息，并降低因各种问题带来的无法访问数据的风险。

可扩展性：

当所有核心程序位于单个服务器中时，集中式存储网络的扩展性普遍存在瓶颈。

随着对网络连接的需求的增加，需要考虑增加更多的存储和带宽及提高处理能力。

分布式存储网络的体系结构允许在多台计算机上分配工作负载，而不是

将其限制在一个地方。

网络中立性：

集中式存储网络给用户带来了较多不便。如互联网服务提供商允许自身主观规范数据的种类和询问信息的速度，愿意支付更多费用的用户才能享受更好的互联网连接。

总之，本移动 IPFS 系统具备扩展性、弹性、敏捷性、易用性、成本可控等特点。

二、 研究主要功能及技术参数

1. 主要功能

1.1 IPFS 系统网络架构

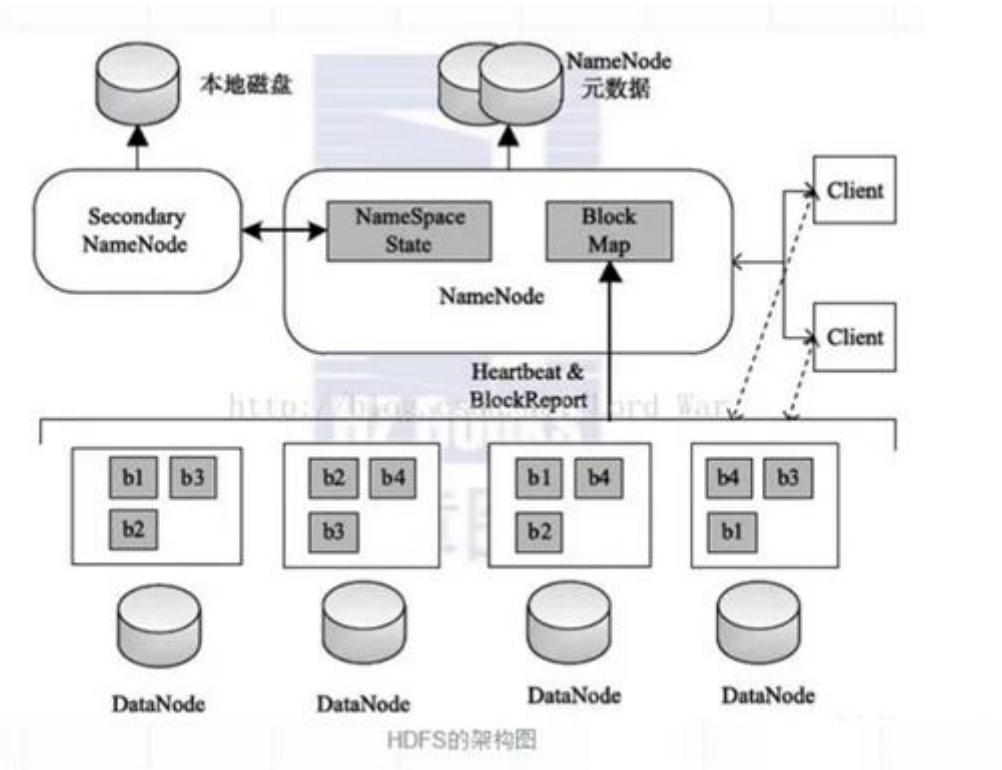
基于标准硬件和分布式架构，实现千节点/EB 级扩展，同时可以对块、对象、文件等多种类型存储统一管理。

分布式存储就是将数据分散存储到多个存储服务器上，并将这些分散的存储资源构成一个虚拟的存储设备，实际上数据分散的存储在企业的各个角落。

分布式存储技术的实现，往往离不开底层的分布式存储框架。根据其存储的类型，可分为块存储，对象存储和文件存储。在主流的分布式存储技术中，HDFS 属于文件存储，Swift 属于对象存储，而 Ceph 可支持块存储、对象存储和文件存储，故称为统一存储。

1.1.1.HDFS

HDFS 是 Hadoop 核心组成之一，是分布式计算中数据存储管理的基础，被设计成适合运行在通用硬件上的分布式文件系统。



HDFS的架构图

1.1.1.1 HDFS 的功能模块

Client

Client 是用户与 HDFS 交互的手段，当文件上传 HDFS 的时候，Client 将文件切分成一个一个的 Block，然后进行上传；Client 通过与 NameNode 交互，来获取文件的位置信息；与 DataNode 交互，读取或者写入数据；Client 还可以提供 NameNode 格式化等一些命令来管理 HDFS;同时,Client 可以通过对 HDFS 的增删改查等操作来访问 HDFS。

NameNode

NameNode 就是 HDFS 的 Master 架构，它维护着文件系统树及整棵树内所有的文件和目录，HDFS 文件系统中处理客

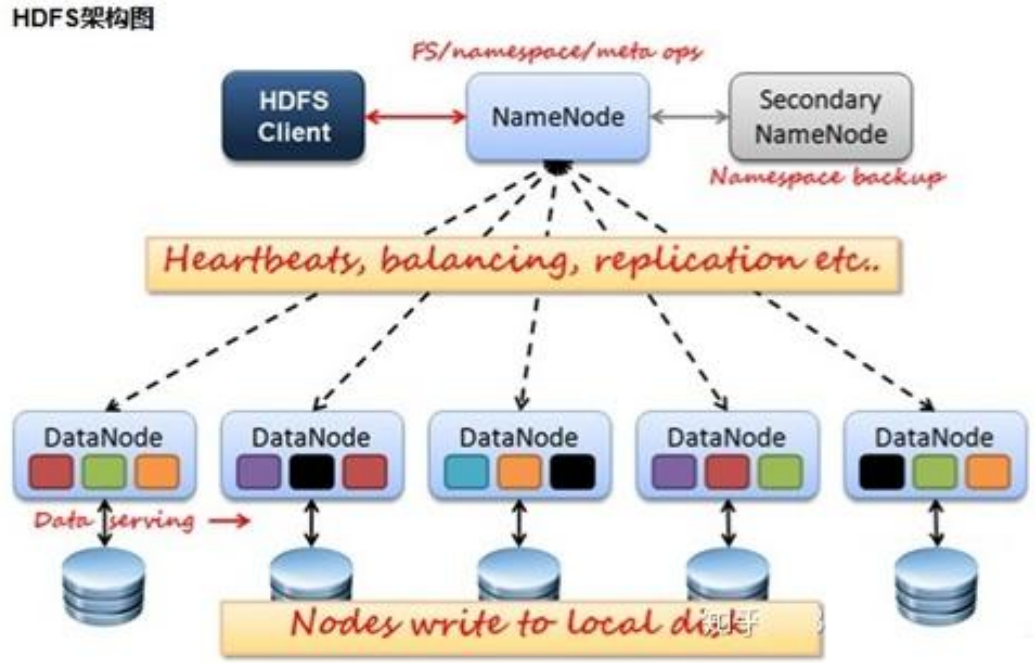
服端读写请求、管理数据块（Block）的映射信息、配置副本策略等管理工作由 NameNode 来完成。

DataNode

NameNode 下达命令，DataNode 执行实际操作。DataNode 表示实际存储的数据块，同时可以执行数据块的读写操作。

Secondary NameNode

Secondary NameNode 的功能主要是辅助 NameNode，分担其工作量；在紧急情况下可以辅助恢复 NameNode，但是它不能替换 NameNode 并提供服务。



1.1.1.2 HDFS 的优势

a. 容错性：数据自动保存多个副本。通过增加副本的形式，提高容错性。其中一个副本丢失以后，可以自动恢复。

b. 可以处理大数据：能够处理数据规模达到 GB、TB 甚至 PB 级别的数据；能够处理百万规模以上的文件数量。

c.可以构建在廉价的机器上，通过多副本机制，提高可靠性。

1.1.1.3 HDFS 的缺点

a.不适合低延时数据访问：比如毫秒级的存储数据，是做不到的。

b.无法高效对大量小文件进行存储：存储大量小文件的话，它会占用 NameNode 大量的内存来存储文件目录和块信息。这样是不可取的，因为 NameNode 的内存总是有限的。同时，小文件存储的寻址时间会超过读取时间，它违反了 HDFS 的设计目标。

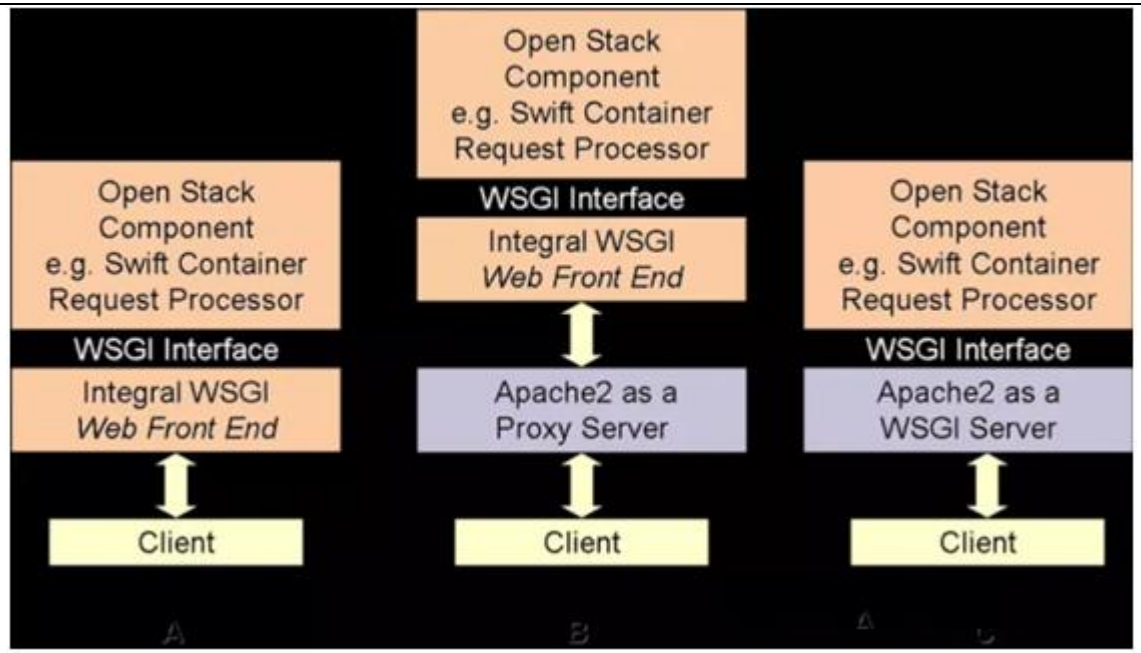
c.不支持并发写入、文件随机修改：一个文件只能有一个写，不允许多个线程同时写。仅支持数据 append（追加），不支持文件的随机修改。

1.1.2 Swift

Swift 于 2008 年起步，最初是由 Rackspace 公司开发的分布式对象存储服务， 2010 年贡献给 OpenStack 开源社区。现如今已部署到大规模公有云的生产环境中使用。

1.1.2.1 Swift 的功能模块

Swift 采用完全对称、面向资源的分布式系统架构设计，所有组件都可扩展，避免因单点失效而影响整个系统的可用性。



Proxy Server（代理服务）：对外提供对象服务 API，Proxy Server 首先会通过 Ring 查找被操作实体的物理位置，随后将请求转发至相应的账户、容器或对象服务。

Authentication Server（认证服务）：验证用户的身份信息，并获得一个访问令牌(Token)。

Cache Server（缓存服务）：缓存令牌，账户和容器信息，但不会缓存对象本身的数据。

Account Server(账户服务)：Account Server 是存储节点中负责处理 Account 的 get、head、put、delete、relication 请求的服务进程。提供账户元数据和统计信息，并维护所含容器列表的服务。

Container Server（容器服务）：Container Server 是存储节点中负责处理 Container 的 get、head、put、delete、relication 请求的服务进程。提供容器元数据和统计信息，并维护所含对象列表的服务。

Object Server（对象服务）：Object Server 就是一个简单的 BLOB 存储服务器，可以存储、检索和删除保存再本地设备的对象。提供对象元数据和内容服务，每个对象会以文件存储在文件系统中。

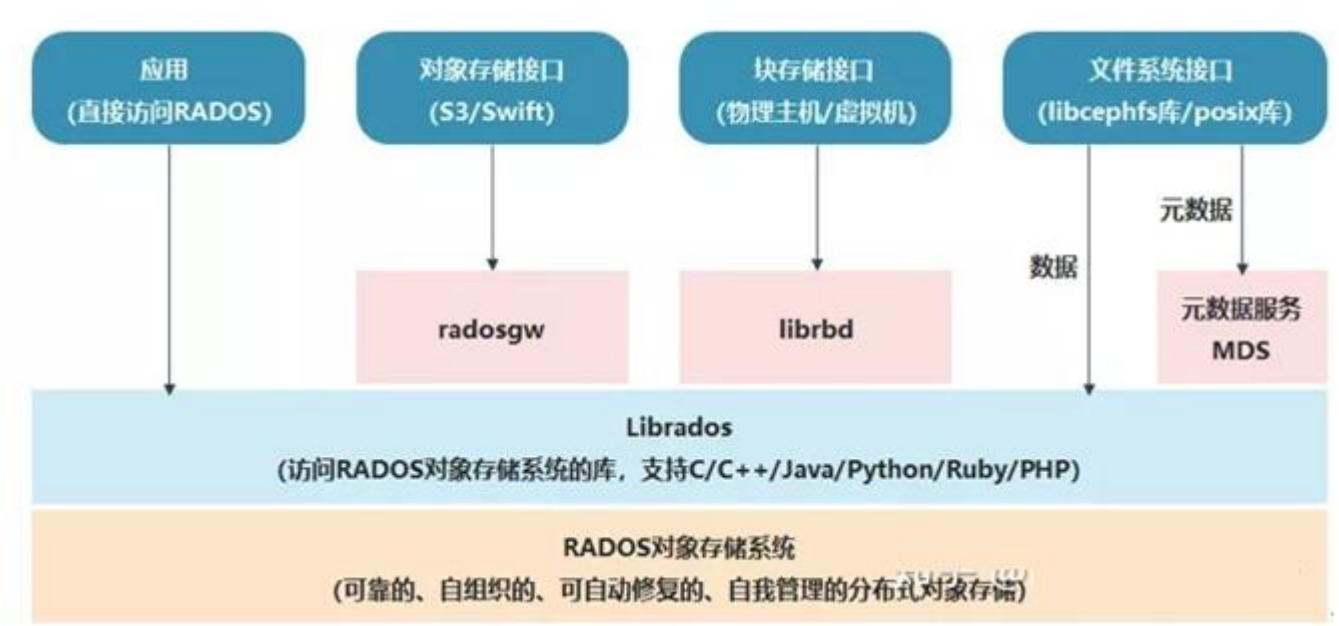
Replicator（复制服务）：检测本地副本和远程副本是否一致，采用推式(Push)更新远程副本。

Updater（更新服务）：对象内容的更新。
Auditor（审计服务）：检查对象、容器和账户的完整性，如果发现错误，文件将被隔离。
Account Reaper（账户清理服务）：移除被标记为删除的账户，删除其所包含的所有容器和对象。

1.1.3.Ceph

Ceph 最早起源于 Sage 就读博士期间的工作、成果于 2004 年发表，并随后贡献给开源社区.经过十几年的发展，已成为应用最广泛的开源分布式存储平台。

1.1.3.1 Ceph 的主要架构



基础存储系统 RADOS

Ceph 的最底层是 RADOS（分布式对象存储系统），它具有可靠、智能、分布式等特性，实现高可靠、高可拓展、高

性能、高自动化等功能，并最终存储用户数据。RADOS 系统主要由 Ceph OSD、Ceph Monitors 两部分组成，Ceph OSD 的功能是存储数据，处理数据的复制、恢复、回填、再均衡，并通过检查其他 OSD 守护进程的心跳来向 Ceph Monitors 提供一些监控信息。Ceph Monitor 维护着展示集群状态的各种图表，包括监视器图、 OSD 图、归置组（ PG ）图、和 CRUSH 图。

基础库 LIBRADOS

LIBRADOS 层的功能是对 RADOS 进行抽象和封装，并向上层提供 API，以便直接基于 RADOS 进行应用开发。RADOS 是一个对象存储系统，因此，LIBRADOS 实现的 API 是针对对象存储功能的。物理上，LIBRADOS 和基于其上开发的应用位于同一台机器，因而也被称为本地 API。应用调用本机上的 LIBRADOS API，再由后者通过 socket 与 RADOS 集群中的节点通信并完成各种操作。

上层应用接口

Ceph 上层应用接口涵盖了 RADOSGW（RADOS Gateway）、RBD（Reliable Block Device）和 Ceph FS（Ceph File System），其中，RADOSGW 和 RBD 是在 LIBRADOS 库的基础上提供抽象层次更高、更便于应用或客户端使用的上层接口。

应用层

应用层就是不同场景下对于 Ceph 各个应用接口的各种应用方式，例如基于 LIBRADOS 直接开发的对象存储应用，基于 RADOSGW 开发的对象存储应用，基于 RBD 实现的云主机硬盘等。

1.1.3.2 Ceph 的功能模块

Client 客户端：负责存储协议的接入，节点负载均衡。

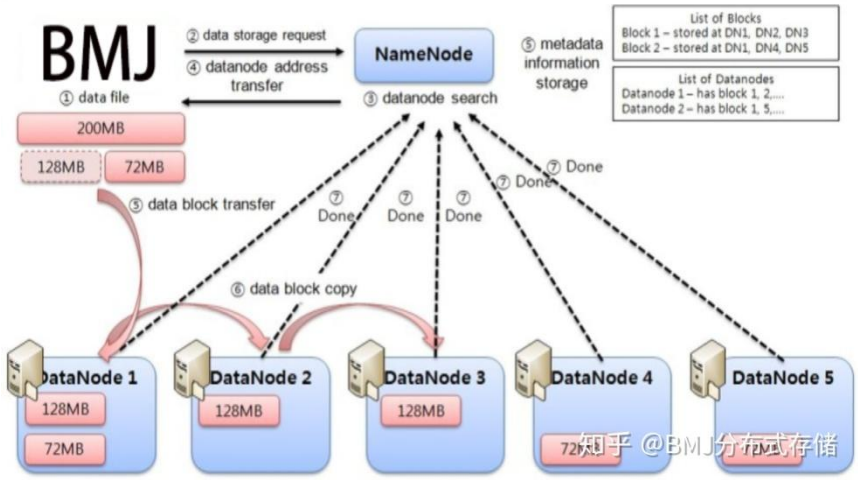
MON 监控服务：负责监控整个集群，维护集群的健康状态，维护展示集群状态的各种图表，如 OSD Map、Monitor Map、PG Map 和 CRUSH Map。

MDS 元数据服务：负责保存文件系统的元数据，管理目录结构。

OSD 存储服务：主要功能是存储数据、复制数据、平衡数据、恢复数据，以及与其它 OSD 间进行心跳检查等。一般情况下一块硬盘对应一个 OSD。

1.1.3.3 Ceph 的优点

1.1.3.3 .1.CRUSH 算法



CRUSH 算法是 ceph 的两大创新之一，简单来说，ceph 摒弃了传统的集中式存储元数据寻址的方案，转而使用 CRUSH 算法完成数据的寻址操作。采用 CRUSH 算法，数据分布均衡，并行度高，不需要维护固定的元数据结构。

1.1.3.3.2.CRUSH 算法

a.高可用

Ceph 中的数据副本数量可以由管理员自行定义，并可以通过 CRUSH 算法指定副本的物理存储位置以分隔故障域，支持数据强一致性，适合读多写少场景；ceph 可以忍受多种故障场景并自动尝试并行修复。

b.高扩展性

Ceph 本身并没有主控节点，扩展起来比较容易，并且理论上，它的性能会随着磁盘数量的增加而线性增长。

c.特性丰富

Ceph 支持对象存储、块存储和文件存储服务，故称为统一存储

1.1.3.4 Ceph 的缺点

a.去中心化的分布式解决方案，需要提前做好规划设计，对技术团队的要求能力比较高。

b.Ceph 扩容时，由于其数据分布均衡的特性，会导致整个存储系统性能的下降。

1.2 IPFS 系统硬件架构

本去中心化存储的 IPFS 网络整个架构由节点，即**终端处理器**组成，由它与其他远端节点通过 4G(5G)组网，由它与本地手机和服务器通过蓝牙 mesh 组网（蓝牙无线网络）。整个系统网络架构如图 1 所示：

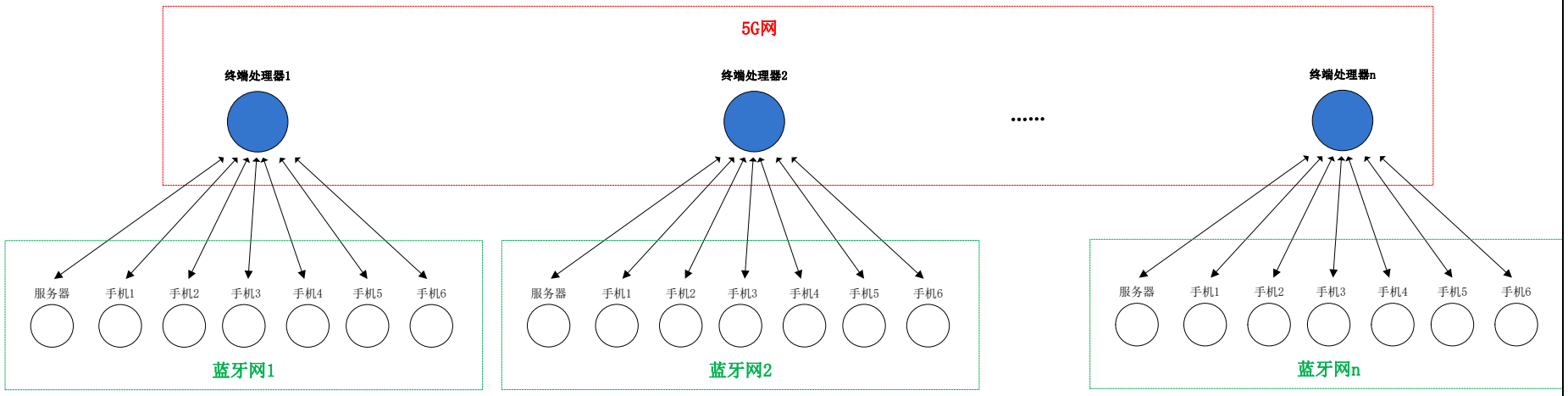


图 1 去中心化存储的 IPFS 网络架构图

1.2 终端处理器模块构成

终端处理器的结构如图 2 所示，它由 4G(5G)网络模块、数据处理模块、蓝牙通信模块和能量供应模块四部分组成。

a） 4G(5G)网络模块通过基站收发 IPFS 信息，并将信息通过串口与数据处理模块通信；

b） 数据处理模块包括树莓派、存储器等，负责控制整个终端处理器的操作，存储并处理 4G(5G)节点本身采集的数据及蓝牙子网节点生的数据，是整个节点的“大脑”；

c） 蓝牙无线通信模块负责与手机、服务器节点进行无线通信，交换控制信息及收发采集数据；

d） 能量供应模块一般采用可充电电池或太阳能，为终端处理器提供运行所需的能量。

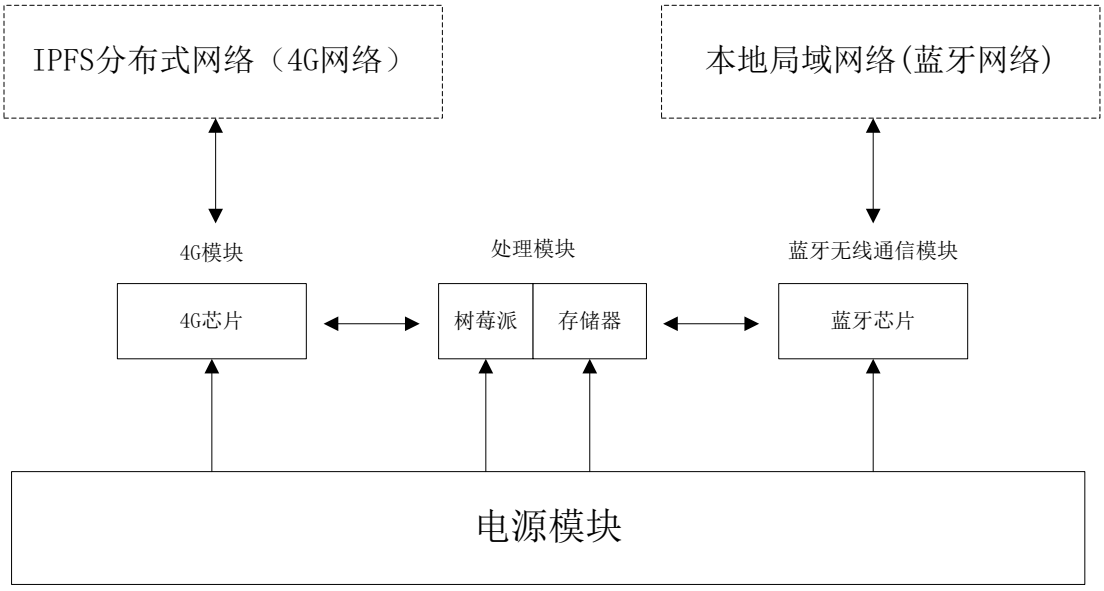


图 2 终端处理器模块结构图

1.3 4G(5G)模块

4G(5G)模块通过无线基站收发 IPFS 信息，并将信息通过串口与数

据处理模块通信。支持上电自动寻网。

1.4 处理模块

数据处理模块包括树莓派、存储器等，负责控制整个终端处理器的操作，存储并处理 4G(5G)节点本身采集的数据及蓝牙子网节点（手机和服务器）产生的数据。

1.5 蓝牙模块

蓝牙无线通信模块负责与手机、服务器进行无线通信，交换控制信息及收发采集数据。

2、网络硬件系统可采用的文件格式

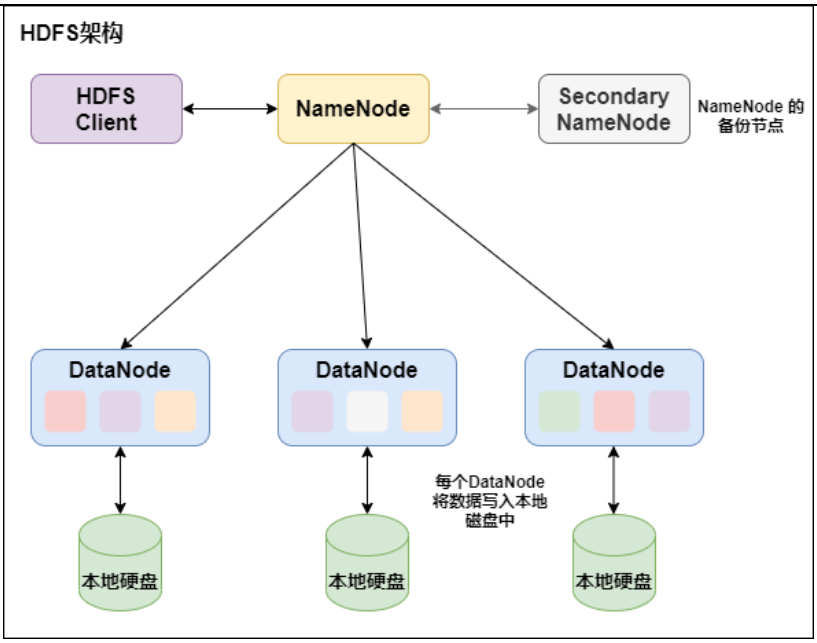
本 IPFS 移动硬件系统支持的主流的分布式文件系统有：GFS、HDFS、Ceph、Lustre、MogileFS、MooseFS、FastDFS、TFS、GridFS 等。

2.1 GFS

Google 公司为了满足本公司需求而开发的基于 Linux 的专有分布式文件系统。尽管 Google 公布了该系统的一些技术细节，但 Google 并没有将该系统的软件部分作为开源软件发布。

2.2 HDFS

HDFS（Hadoop Distributed File System）是 Hadoop 项目的一个子项目。是 Hadoop 的核心组件之一， Hadoop 非常适于存储大型数据 (比如 TB 和 PB)，其就是使用 HDFS 作为存储系统. HDFS 使用多台计算机存储文件，并且提供统一的访问接口，像是访问一个普通文件系统一样使用分布式文件系统。



2.3 TFS

TFS 是一个高可扩展、高可用、高性能、面向互联网服务的分布式文件系统，主要针对海量的非结构化数据，它构筑在普通的 Linux 机器 集群上，可为外部提供高可靠和高并发的存储访问。TFS 为淘宝提供海量小文件存储，通常文件大小不超过 1M，满足了淘宝对小文件存储的需求，被广泛地应用在淘宝各项应用中。它采用了 HA 架构和平滑扩容，保证了整个文件系统的可用性和扩展性。同时扁平化的数据组织结构，可将文件名映射到文件的物理地址，简化 了文件的访问流程，一定程度上为 TFS 提供了良好的读写性能。

2.4 Lustre

Lustre 是一个大规模的、安全可靠的，具备高可用性的集群文件系统，它是由 SUN 公司开发和维护的。该项目主要的目的就是开发下一代的集群文件系统，可以支持超过 10000 个节点，数以 PB 的数据量存储系统。目前 Lustre 已经运用在一些领域，例如 HP SFS 产品等。

2.5 MooseFS

MooseFS 是一款相对小众的分布式文件系统，不需要修改上层应用接口即可直接使用，支持 FUSE 的操作方式，部署简单并提供 Web 界面的方式进行管理与监控，同其他分布式操作系统一样，支持在线扩容，并进行横向扩展。MooseFS 还具有可找回误操作删除的文件，相当于一个回收站，方便业务进行定制；同时 MooseFS 对于海量小文件的读写要比大文件读写的效率高的多。

但 MooseFS 的缺点同样明显，MFS 的主备架构情况类似于 MySQL 的主从复制，

从可以扩展，主却不容易扩展。短期的对策就是按照业务来做切分，随着 MFS 体系架构中存储文件的总数上升，Master Server 对内存的需求量会不断增大。并且对于其单点问题官方自带的是把数据信息从Master Server同步到Metalogger Server上，Master Server一旦出问题 Metalogger Server 可以恢复升级为 Master Server，但是需要恢复时间。目前，也可以通过第三方的高可用方案（heartbeat+drbd+moosefs）来解决 Master Server 的单点问题。

2.6 MogileFS

由 memcahed 的开发公司 danga 一款 perl 开发的产品，目前国内使用 mogielFS 的有图片托管网站 yupoo 等。MogileFS 是一套高效的文件自动备份组件，由 Six Apart 开发，广泛应用在包括 LiveJournal 等 web2.0 站点上。

2.7 FastDFS

是一款类似 Google FS 的开源分布式文件系统，是纯 C 语言开发的。FastDFS 是一个开源的轻量级分布式文件系统，它对文件进行管理，功能包括：文件存储、文件同步、文件访问（文件上传、文件下载）等，解决了大容量存储和负载均衡的问题。特别适合以文件为载体的在线服务，如相册网站、视频网站等等。

2.8 GlusterFS

开源分布式横向扩展文件系统，可以根据存储需求快速调配存储，内含丰富的自动故障转移功能，且摈弃集中元数据服务器的思想。适用于数据密集型任务的可扩展网络文件系统，具有可扩展性、高性能、高可用性等特点。gluster 于 2011 年 10 月 7 日被 Red Hat 收购。

2.9 GridFS

MongoDB 是知名的 NoSQL 数据库，GridFS 是 MongoDB 的一个内置功能，它提供一组文件操作的 API 以利用 MongoDB 存储文件，GridFS 的基本原理是将文件保存在两个 Collection 中，一个保存文件索引，一个保存文件内容，文件内容按一定大小分成若干块，每一块存在一个 Document 中，这种方法不仅提供了文件存储，还提供了对文件相关的一些附加属性（比如 MD5 值，文件名等等）的存储。文件在 GridFS 中会按 4MB 为单位进行分块存储。