一、块存储系统、对象存储系统、文件存储系统的区别

1. 存储设备不同

对象存储对应的存储设备为 swift、s3 等内置大容量硬盘的分布式服务器；文件存储的对应存储设备为 FTP、NFS 服务器；块存储的对应存储设备为 cinder、硬盘、磁盘阵列。

2. 体现形式不同

比较常见的块存储形式是 Windows 系统硬盘或手机存储空间，数据是按字节来存储和访问的。文件存储一般体现形式是目录和文件（比如 C：\User\Program Files\Common Files），数据以文件的方式存储和访问，按照目录结构进行组织。对象存储一般的体现形式是一个 UUID，这个 UUID 是唯一性的。数据和元数据打包在一起作为一个整体对象存在一个大池子里。用户想访问，只能通过它的 UUID，才能找到它。

3. 使用数据的“用户”不同

块存储的用户是可以读写块设备的软件系统，例如传统的文件系统、数据库；文件存储的用户是自然人；对象存储的用户是其它计算机软件。

4. 适用场景不同

块存储系统适合小型机房、银行使用；文件存储适合数据中心等需要存储大型文件的场景使用；对象存储系统适合互联网环境异地存储，如网络媒体文件存储，即适合各种类型大小的文件存储。

5. 速度不同

块存储系统的延迟较低（10ms 左右），热点突出；对象存储系统的延迟在 100ms 到 1s 不等；文件存储系统的延迟取决于具体的技术。

6. 特性不同

对象存储的特点是具备块存储的高速以及文件存储的共享等特性；文件存储的特点是一个大文件夹，大家都可以获取文件。块存储的特点是分区、格式化后，可以使用，与平常主机内置硬盘的方式无异。

二、阅读论文《The Hadoop Distributed File System》

1. 客户端读取 HDFS 系统中指定文件指定偏移量处的数据时，工作流程是什么?

① HDFS 客户端首先向 NameNode 询问承载该文件块副本的 DataNode 列表。NameNode 通过文件名在其内存中查找该文件的 Block 列表和 DataNode 列表，并根据指定的偏移量找到该文件对应的 Block ID。NameNode 将所有存储该 Block 的 DataNode 的地址返回给客户端。

② 这些返回的 DataNode 地址，会按照集群拓扑结构得出 DataNode 与客户端的距离，然后进行排序，排序遵循两个规则：网络拓扑结构中距离 Client 近的排在前面；心跳机制中超时汇报的 DataNode 状态为 STALE，排在后面。

③ 读取块的内容时，客户端首先尝试最接近的副本。如果读取尝试失败，则客户端将依次尝试下一个副本。如果目标 DataNode 不可用，该节点不再托管该块的副本，或在测试校验和时发现该副本已损坏，则读取可能会失败。

2. 客户端向 HDFS 系统中指定文件追加写入数据的工作流程是什么?

① 打开写入文件的 HDFS 客户端被授予该文件的租约，其他客户端无法写入该文件。写入客户端通过向 NameNode 发送心跳包来定期更新租约。当文件被关闭时，租约将被撤销。

② 当需要产生新 Block 时，NameNode 分配该块一个 Block ID，然后决定若干存储该块副本的 DataNode。这些 DataNode 形成一个 Pipeline。字节流作为一个数据包的序列被推送到 Pipeline 中。应用程序要写入的字节流先在客户端接受缓冲。当缓冲区（通常为 64KB）被填满后，数据才被推入 Pipeline。在接收到一个包的确认之前，可以继续将下一个包推入 Pipeline 中。未完成的数据包的数量受到客户端未完成的数据包窗口大小的限制。

③ 在将数据写入 HDFS 文件之后直到文件关闭前，HDFS 不保证数据对新 Reader 可见。如果用户应用程序需要可见性保证，它可以显式地调用 hflush 操作，使得当前数据包被立即推送到管道。hfush 操作将持续等待直到管道中的所有

DataNode 都确认成功传输了数据包。这样一来，在进行 hflush 操作之前写入的所有数据都会对 Reader 可见。

3. 新增加一个数据块时，HDFS 如何选择存储该数据块的物理节点?

创建新块时，HDFS 将第一个副本放置在 Writer 所在的物理节点上，将第二个和第三个副本放置在不同机架服务器中的两个不同物理节点上，其余放置在随机节点上。放置在随机节点上时需要满足以下两点限制：

(a) 一个节点上至多放置一个副本。

(b) 当副本总数小于机架服务器数量的两倍时，同个机架服务器上至多放置两个副本。

将第二个和第三个副本放置在不同机架服务器上的选择可以更好地在整个群集中分配单个文件的块副本。因为如果前两个副本放在同一个机架服务器上，那么会导致对于任何文件，其三分之二的块副本将放在同一个机架服务器上。

4. HDFS 采用了哪些措施应对数据块损坏或丢失问题?

① 每个数据节点运行一个**块扫描器（Block Scanner）**。该扫描器定期扫描其块副本，并验证存储的校验和是否与块数据匹配。在每个扫描周期中，块扫描器调整读取带宽，以便在一个可配置的周期内完成验证。如果客户端读取一个完整的块并进行校验和验证成功，它会通知数据节点。DataNode 将其作为对复制副本的验证。

② 当读取客户端或块扫描器检测到损坏的块时，它就会通知 NameNode。NameNode 将副本标记为已损坏，但不会立即删除该副本。相反，它开始复制该块的完好副本。只有当完好的副本计数达到块的复制因子时，才会计划删除损坏的副本。这样的策略旨在尽可能长时间地保存数据。**因此，即使一个块的所有副本都已损坏，该策略也允许用户从已损坏的副本中检索数据。**

③ HDFS 为 HDFS 文件中的每个数据块生成并存储**校验和**。校验和在读取时由 HDFS 客户端验证校验和，以帮助检测由客户端、数据节点或网络引起的任何损坏。当 HDFS 读一个文件时，每个块的数据和校验和都会被发送到客户端。客户端计算接收到的数据的校验和，并验证新计算的校验和是否与它接收到的校

验和相匹配。如果没有，客户端将通知 NameNode 该副本已损坏，然后客户端尝试从另一个数据节点获取块的未损坏副本。

④ **HDFS 允许应用程序设置文件的复制因子**。默认情况下，一个文件的复制因子为 3。对于关键文件或经常被访问的文件，具有更高的复制系数可以提高其对故障的容错能力，并增加其读取带宽。

⑤ **DataNode 向 NameNode 发送心跳包**，以确认数据节点正在运行，其主机的块副本可用。如果 NameNode 在十分钟内没有接收到来自某一 DataNode 的心跳，它将认为该 DataNode 已经挂掉，然后计划在其他 DataNode 上创建这些块的新副本。

5. HDFS 采用了什么措施应对主节点失效问题?

① **建立检查点和日志的备份**。映像文件和事务日志是 NameNode 的核心数据，可以配置为拥有多个副本。HDFS 被配置为将检查点和日志存储在多个存储目录中。如果 NameNode 在将日志写入其中一个存储目录时遇到错误，则它会自动将该目录从存储目录列表中排除。如果没有可用的存储目录，NameNode 会将自己关闭。

② **设置 CheckpointNode**。CheckpointNode **定期组合现有的检查点和日志**，以创建一个新的检查点和一个空的日志。CheckpointNode 通常运行在与 NameNo-de 不同的主机上，因为它具有与 NameNode 相同的内存需求。它从 NameNode 下载当前检查点和日志文件，在本地合并它们，并将新检查点返回给 NameNode。创建定期检查点是保护文件系统元数据的一种方法。**如果命名空间映像或日志的所有其他持久性副本都不可用，则系统可以从最近的检查点启动。**

③ **设置 BackupNode**。BackupNode 除了能够创建定期的检查点之外，还维护文件系统名称空间的内存中最新映像，该映像始终与 NameNode 的状态同步。**如果 NameNode 失效，则内存中的 BackupNode 的映像和磁盘上的检查点是最新名称空间状态的记录。**

④ **建立快照（SnapShot）。**集群管理员可以选择在重新启动系统时将 HDFS 回滚到快照状态。NameNode 可恢复创建快照时保存的检查点。

6. NameNode 维护的“数据块物理节点对应表”需不需要在硬盘中备份？为什么？

不需要。

HDFS 将整个命名空间存储在 RAM 中，并不需要永久保存 DataNode 和其存储的数据块对应的信息，即不需要硬盘备份。当新的 DataNode 加入集群的时候，NameNode 会询问 DataNode 有关该 DataNode 存储了哪些 Block 的信息，并间断地更新“数据块物理节点对应表”。