

第一章

1. 试述信息技术发展史上的 3 次信息化浪潮及具体内容。

信息化浪潮	发生时间	标志	解决问题	代表公司
第一次浪潮	1980 年前后	个人计算机	信息解决	Intel、AMD、IBM、苹果、微软、联想、戴尔、惠普等
第二次浪潮	1995 年前后	互联网	信息传播	雅虎、google、阿里巴巴、百度、腾讯等
第三次浪潮	前后	物理网、云计算和大数据	信息爆炸	将涌现出一批新的市场标杆公司

2. 试述数据产生方式经历的几种阶段

答：运营式系统阶段，顾客原创内容阶段，感知式系统阶段。

3. 试述大数据的 4 个基本特性

答：数据量大、数据类型繁多、解决速度快和价值密度低。

4. 试述大数据时代的“数据爆炸”的特性

答：大数据时代的“数据爆炸”的特性是，人类社会产生的数据一致都以每年 50% 的速度增长，也就是说，每两年增长一倍。

5. 数据研究经历了哪4个阶段？

答：人类自古以来在科学研究上先后历经了实验、理论、计算、和数据四种范式。

6. 试述大数据对思维方式的重要影响

答：大数据时代对思维方式的重要影响是三种思维的转变：全样而非抽样，效率而非精确，有关而非因果。

7. 大数据决策与老式的基于数据仓库的决策有什么区别

答：数据仓库具有批量和周期性的数据加载以及数据变化的实时探测、传播和加载能力，能结合历史数据和实时数据实现查询分析和自动规则触发，从而提供对战略决策和战术决策。

大数据决策可以面向类型繁多的、非构造化的海量数据进行决策分析。

8. 举例阐明大数据的基本应用

答：

领域	大数据的应用
金融行业	大数据在高频交易、社区情绪分析和信贷风险分析三大金融创新领域发挥重要作用。
汽车行业	运用大数据和物联网技术的五人驾驶汽车，在不远的将来将走进我们的平常生活
互联网行业	借助于大数据技术，可以分析客户行为，进行商品推荐和有针对性广告投放
个人生活	大数据还可以应用于个人生活，运用与每个人有关联的“个人大数据”，分析个人生活行为习惯，为其提供更加周全的个性化服务。

9. 举例阐明大数据的核心技术

答：批解决计算，流计算，图计算，查询分析计算

10. 大数据产业涉及哪些核心技术。

答：IT 基本设施层、数据源层、数据管理层、数据分析层、数据平台层、数据应用层。

11. 定义并解释如下术语：云计算、物联网

答：云计算：云计算就是实现了通过网络提供可伸缩的、便宜的分布式计算机能力，顾客只需要在具有网络接入条件的地方，就可以随时随处获得所需的多种 IT 资源。

物联网是物物相连的互联网，是互联网的延伸，它运用局部网络或互联网等通信技术把传感器、控制器、机器、人类和物等通过新的方式连在一起，形成人与物、物与物相连，实现信息化和远程管理控制。

12. 具体论述大数据、云计算和物联网三者之间的区别与联系。

大数据、云计算和物联网的区别	大数据、云计算和物联网的联系
大数据侧重于海量数据的存储、解决与分析，海量数据中发现价值，服务于生产和生活；云计算本质上皆在整合和优化多种 IT 资源并通过网络已服务的措施，便宜地提供应顾客；物联网的发展目的是实现万物互联，应用创新是物联网的核心	

	<p>从整体来看，大数据、云计算和物联网这三者是相辅相成的。大数据根植于云计算，大数据分析的诸多技术都来自于云计算，云计算的分布式存储和管理系统提供了海量数据的存储和管理能力，没有这些云计算技术作为支撑，大数据分析就无从谈起。物联网的传感器源源不断的产生大量数据，构成了大数据的重要数据来源，物联网需要借助于云计算和大数据技术，实现物联网大数据的存储、分析和解决。</p>
--	--

第二章

1. 试述 hadoop 和 google 的 mapreduce、gfs 等技术之间的关系

答：Hadoop 的核心是分布式文件系统 HDFS 和 MapReduce，HDFS 是 google 文件系统 GFS 的开源实现，MapReduces 是针对 googleMapReduce 的开源实现。

2. 试述 Hadoop 具有哪些特性。

答：高可靠性，高效性，高可扩展性，高容错性，成本低，运营在 Linux 平台，支持多种编程语言

3. 试述 Hadoop 在各个领域的应用状况。

答：，雅虎在 Sunnyvale 总部建立了 M45——一种涉及了 4000 个解决器和 1.5PB 容量的 Hadoop 集群系统；

Facebook 重要将 Hadoop 平台用于日记解决，推荐系统和数据仓库等方面；

百度重要使用 Hadoop 于日记的存储和记录、网页数据的分析和挖掘、商业分析、在线数据反馈、网页聚类等。

4. 试述 Hadoop 的项目构造以及每个部分的具体功能。

答：

Pig	Chukwa	Hive	HBase
MapReduce	HDFS		Zookeeper
Common		Avro	

Common 是为 Hadoop 其他子项目提供支持的常用工具，重要涉及文献系统、RPC 和串行化库

Avro 是为 Hadoop 的子项目，用于数据序列化的系统，提供了丰富的数据构造类型、迅速可压缩的二进制数据格式、存储持续性数据的文献集、远程调用的功能和简朴的动态语言集成功能。

HDFS 是 Hadoop 项目的两个核心之一，它是针对 google 文献系统的开源实现。

HBase 是一种提高可靠性、高性能、可伸缩、实时读写、分布式的列式数据库，一般采用 HDFS 作为其底层数据存储。

MapReduce 是针对 googleMapReduce 的开源实现，用于大规模数据集的并行运算。

Zookeeper 是针对 googleChubby 的一种开源实现，是高效和可靠的协同工作系统，提供分布式锁之类的基本服务，用于构建分布式应用，减轻分布式应用程序所承担的协调任务。

Hive 是一种基于 Hadoop 的数据仓库工具，可以用于对 Hadoop 文献中的数据集进行数据整顿、特殊查询和分布存储。

Pig 是一种数据流语言和运营环境，适合于使用 Hadoop 和 MapReduce 平台上查询大型半结构化数据集。

Sqoop 可以改善数据的互操作性，重要用来在 Hadoop 和关系数据库之间互换数据。

Chukwa 是一种开源的、用于监控大型分布式系统的数据收集系统，可以将多种类型的数据收集成适合 Hadoop 解决的文献，并保存在 HDFS 中供 Hadoop 进行多种 MapReduce 操作。

第三章

1. 试述分布式文献系统设计的需求。

设计需求	含义	HDFS 的实现状况
------	----	------------

透明性		
-----	--	--

	具有访问透明性、位置透明性、性能、和伸缩透明性	只能提供一定限度的访问透明性，完全支持位置透明性、性能和伸缩透明性
并发控制	客户端对于文献的读写不应当影响其他客户端对同一种文献的读写	机制非常简朴，任何时候都只容许有一种程序写入某个文献
文献复制	一种文献可以拥有不同位置的多种副本	HDFS 采用了多副本机制
硬件和操作系统的异构性	可以在不同的操作系统和计算机上实现同样的客户端和服务端程序	采用 Java 语言开发，具有较好的跨平台能力
可伸缩性	支持节点的动态加入或退出	建立在大规模便宜机器上的分布式文献系统集群，具有较好的伸缩性
容错	保证文献服务在客户端或者服务端浮现问题的时候能正常使用	具有多副本机制和故障自动检测、恢复机制
安全	保证系统的安全性	安全性较弱

2· 分布式文献系统是如何实现较高水平扩展的？

分布式文献系统在物理构造上是由计算机集群中的多种节点构成的，这些节点分为两类，一类叫“主节点” (Master Node) 或者也被称为“名称结点” (NameNode)，另一类叫“从节点” (Slave Node) 或者也被称为“数据节点” (DataNode)

3· 试述 HDFS 中的块和一般文献系统中的块的区别。

答:在老式的文献系统中,为了提高磁盘读写效率,一般以数据块为单位,恶如不是以字节为单位。

HDFS 中的块,默认一种块大小为 64MB,而 HDFS 中的文献会被拆提成多种块,每个块作为独立的单元进行存储。HDFS 在块的大小的设计上明显要不小于一般文献系统。

4·试述 HDFS 中的名称节点和数据节点的具体功能。

答:名称节点负责管理分布式文献系统系统的命名空间,记录分布式文献系统中的每个文献中各个块所在的数据节点的位置信息;

数据节点是分布式文献系统 HDFS 的工作节点,负责数据的存储和读取,会根据客户端或者是名称节点的调度来进行数据的存储和检索,并向名称节点定期发送自己所存储的块的列表。

`hadoop fs -ls <path>` 显示<path>指定的文献的具体信息

`hadoop fs -cat <path>` 将<path>指定的文献的内容输出到原则输出

`hadoop fs -mkdir <path>` 创立<path>指定的文献夹

`hadoop fs -get [-ignorecrc] [-crc] <src><localdst>` 复制<src>指定的文献到本地文献系统<localdst>指定的文献或文献夹。`-ignorecrc` 选项复制 CRC 校验失败的文献。使用`-crc` 选项复制文献以及 CRC 信息。

hadoop fs -put <localsrc><dst> 从本地文献系统中复制<localsrc>指定的单个或多种源文献到<dst>指定的目的文献系统中。也支持从原则输入(stdin)中读取输入写入目的文献系统。

hadoop fs -rmr <path> 删除<path>指定的文献夹及其的所有文献

第四章

1. 试述在 Hadoop 体系架构中 HBase 与其他构成部分的互相关系。

答：HBase 运用 Hadoop MapReduce 来解决 HBase 中的海量数据，实现高性能计算；运用 Zookeeper 作为协同服务，实现稳定服务和失败恢复；使用 HDFS 作为高可靠的底层存储，运用便宜集群提供海量数据存储能力；Sqoop 为 HBase 的底层数据导入功能，Pig 和 Hive 为 HBase 提供了高层语言支持，HBase 是 BigTable 的开源实现。

2. 请论述 HBase 和 BigTable 的底层技术的相应关系

答：

项目	BigTable	HBase
文献存储系统	GFS	HDFS
海量数据解决	MapReduce	Hadoop MapReduce

协同服务管理	Chubby	Zookeeper
--------	--------	-----------

3. 请论述 HBase 和老式关系数据库的区别

答：

区别	老式关系数据库	HBase
数据类型	关系模型	数据模型
数据操作	插入、删除、更新、查询、多表连接	插入、查询、删除、清空，无法实现表与表之间关联
存储模式	基于行模式存储，元组或行会被持续地存储在磁盘中也	基于列存储，每个列族都由几种文献保存，不同列族的文献是分离的
数据索引	针对不同列构建复杂的多种索引	只有一种行键索引
数据维护	用最新的目前值去替代记录中本来的旧值	更新操作不会删除数据旧的版本，而是生成一种新的版本
可伸缩性	很难实现横向扩展，纵向扩展的空间也比较有限	容易地通过在集群中增长或者减少硬件数量来实现性能的伸缩

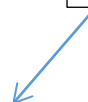
4. HBase 有哪些类型的访问接口？

答：HBase 提供了 Native Java API，HBase Shell，Thrift Gateway，REST GateWay，Pig，Hive 等访问接口。

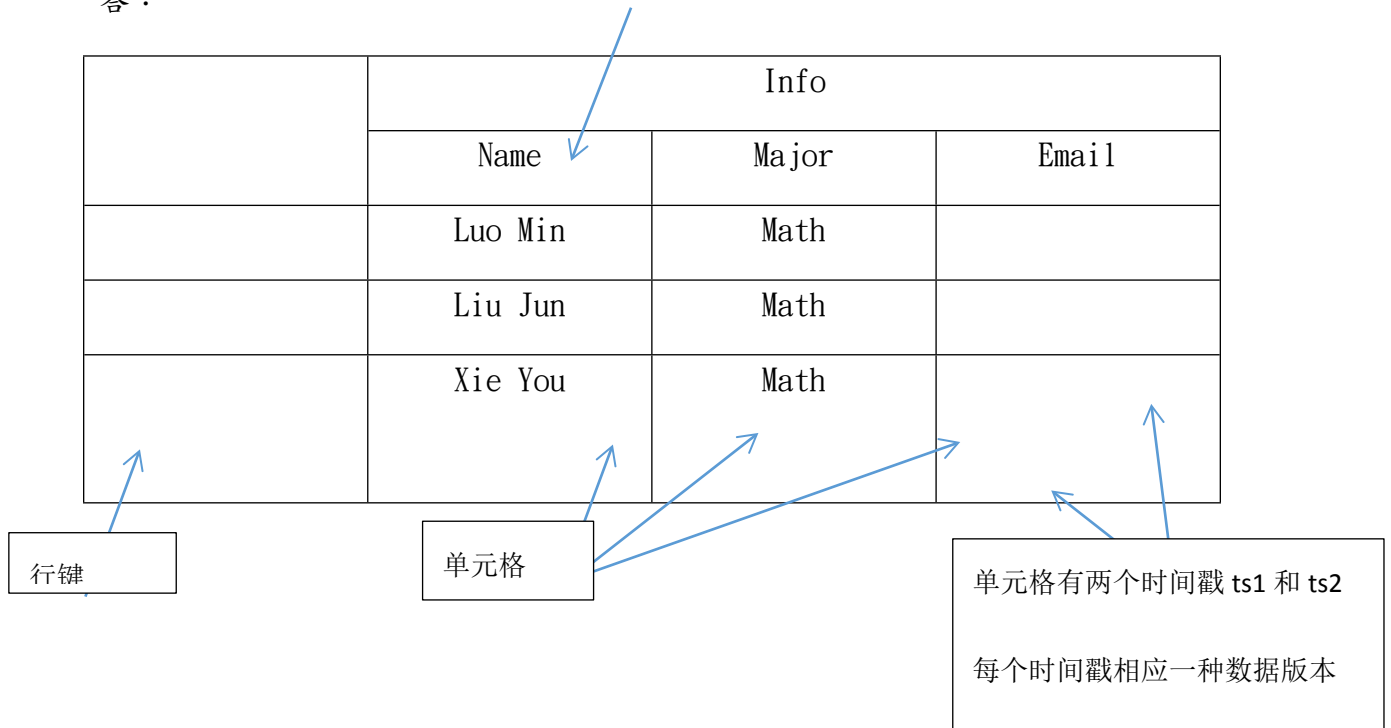
5. 请以实例阐明 HBase 数据模型。

列限定符

列族



答：



6. 分别解释 HBase 中行键、列键和时间戳的概念

(1) 行键是唯一的，在一种表里只浮现一次，否则就是在更新同一行，行键可以是任意的字节数组。

(2) 列族需要在创立表的时候就定义好，数量也不适宜过多。列族名必须由可打印字符构成，创立表的时候不需要定义好列。

(3) 时间戳，默认由系统指定，顾客也可以显示设立。使用不同的时间戳来辨别不同的版本。

7. 请举个实例来论述 HBase 的概念视图和物理视图的不同

HBase 数据概念视图

行键	时间戳	列族 contents	列族 anchor
"com. cnn. www"	T5		Anchor:cnnsi.com=" CNN"
	T3		

			Anchor:my. look. ca=" CNN"
"com. cnn. www"	T3	Content:html=" <html>... "	
	T2	Content:html=" <html>... "	
	T1	Content:html=" <html>... "	

HBase 数据物理视图

行键	时间戳	列族 anchor
"com. cnn. www"	T5	Anchor:cnnsi. com=" CNN"
	T4	Anchor:my. look. ca=" CNN"

行键	时间戳	列族 contents
"com. cnn. www"	T3	Content:html=" <html>... "
	T2	Content:html=" <html>... "
	T1	Content:html=" <html>... "

在 HBase 的概念视图中，一种表可以视为一种稀疏、多维的映射关系。

在物理视图中，一种表会按照属于同一列族的数据保存在一起

8. 试述 HBase 各功能组建及其作用

(1) 库函数：链接到每个客户端；

(2) 一种 Master 主服务器：主服务器 Master 重要负责表和 Region 的管理工作；

(3) 许多种 Region 服务器：Region 服务器是 HBase 中最核心的模块，负责维护分派给自己的 Region，并响应顾客的读写祈求

9. 请论述 HBase 的数据分区机制。

答：HBase 采用分区存储，一种大的表会被分拆许多种 Region，这些 Region 会被分发到不同的服务器上实现分布式存储。

10. HBase 中的分区是如何定位的。

通过构建的映射表的每个条目涉及两项内容，一种是 Regionde 标记符，另一种是 Region 服务器标记，这个条目就标记 Region 和 Region 服务器之间的相应关系，从而就可以懂得某个 Region 被保存在哪个 Region 服务器中。

11. 试述 HBase 的三层构造中各层次的名称和作用。

层次	名称	作用
第一层	Zookeeper 文献	记录了-ROOT-表的位置信息
第二层	-ROOT-表	记录了.META.表的 Region 位置信息 -ROOT-表只能有一种 Region。通过-ROOT-表，就可以访问.META.表中的数据
第三层	.META.表	记录了顾客数据表的 Region 位置信息，.META.表可以有多种 Region，保存了 HBase 中所有顾客数据表的 Region 位置信息

12. 请论述 HBase 的三层构造下，客户端是如何访问到数据的。

答：一方面访问 Zookeeper，获取 -ROOT 表的位置信息，然后访问 -Root- 表，获得 .META. 表的信息，接着访问 .META. 表，找到所需的 Region 具体位于哪个 Region 服务器，最后才会到该 Region 服务器读取数据。

13. 试述 HBase 系统基本架构以及每个构成部分的作用。

(1) 客户端

客户端涉及访问 HBase 的接口，同步在缓存中维护着已经访问过的 Region 位置信息，用来加快后续数据访问过程

(2) Zookeeper 服务器

Zookeeper 可以协助选举出一种 Master 作为集群的总管，并保证在任何时刻总有唯一一种 Master 在运营，这就避免了 Master 的“单点失效”问题

(3) Master

主服务器 Master 重要负责表和 Region 的管理工作：管理顾客对表的增长、删除、修改、查询等操作；实现不同 Region 服务器之间的负载均衡；在 Region 分裂或合并后，负责重新调节 Region 的分布；对发生故障失效的 Region 服务器上的 Region 进行迁移

(4) Region 服务器

Region 服务器是 HBase 中最核心的模块，负责维护分派给自己的 Region，并响应顾客的读写祈求

14. 请论述 Region 服务器向 HDFS 文档系统中读写数据的基本原理

Region 服务器内部管理一系列 Region 对象和一种 HLog 文献，其中，HLog 是磁盘上面的记录文献，它记录着所有的更新操作。每个 Region 对象又是由多种 Store 构成的，每个 Store 对象了表中的一种列族的存储。每个 Store 又涉及了 MemStore 和若干个 StoreFile，其中，MemStore 是在内存中的缓存。

15. 试述 HStore 的工作原理

每个 Store 相应了表中的一种列族的存储。每个 Store 涉及一种 MemStore 缓存和若干个 StoreFile 文献。MemStore 是排序的内存缓冲区，当顾客写入数据时，系统一方面把数据放入 MemStore 缓存，当 MemStore 缓存满时，就会刷新到磁盘中的一种 StoreFile 文献中，当单个 StoreFile 文献大小超过一定阈值时，就会触发文献分裂操作。

16. 试述 HLog 的工作原理

答：HBase 系统为每个 Region 服务器配备了一种 HLog 文献，它是一种预写式日记（Write Ahead Log），顾客更新数据必须一方面写入日记后，才干写入 MemStore 缓存，并且，直到 MemStore 缓存内容相应的日记已经写入磁盘，该缓存内容才干被刷写到磁盘。

17. 在 HBase 中，每个 Region 服务器维护一种 HLog，而不是为每个 Region 都单独维护一种 HLog。请阐明这种做法的优缺陷。

长处：多种 Region 对象的更新操作所发生的日记修改，只需要不断把日记记录追加到单个日记文献中，不需要同步打开、写入到多种日记文献中。

缺陷：如果一种 Region 服务器发生故障，为了恢复其上次 Region 对象，需要将 Region 服务器上的对象，需要将 Region 服务器上的 HLog 按照其所属的 Region 对象进行拆分，然后分发到其他 Region 服务器上执行恢复操作。

18. 当一台 Region 服务器意外终结时，Master 如何发现这种意外终结状况？为了恢复这台发生意外的 Region 服务器上的 Region, Master 应当做出哪些解决（涉及如何使用 HLog 进行恢复）？

Zookeeper 会实时监测每个 Region 服务器的状态，当某个 Region 服务器发生故障时，Zookeeper 会告知 Master。

Master 一方面会解决该故障 Region 服务器上面遗留的 HLog 文献，这个遗留的 HLog 文献中涉及了来自多种 Region 对象的日记记录。

系统会根据每条日记记录所属的 Region 对象对 HLog 数据进行拆分，分别放到相应 Region 对象的目录下，然后，再将失效的 Region 重新分派到可用的 Region 服务器中，并把与该 Region 对象有关的 HLog 日记记录也发送给相应的 Region 服务器。

Region 服务器领取到分派给自己的 Region 对象以及与之有关的 HLog 日记记录后来，会重新做一遍日记记录中的多种操作，把日记记录中的数据写入到 MemStore 缓存中，然后，刷新到磁盘的 StoreFile 文献中，完毕数据恢复。

第五章

1. 如何精确理解 NoSQL 的含义？

NoSQL 是一种不同于关系数据库的数据库管理系统设计方式，是对非关系型数据库的一类统称，它采用的数据模型并非老式关系数据库的关系模型，而是类似键/值、列族、文档等非关系模型。

2. 试述关系数据库在哪些方面无法满族 Web2.0 应用的需求。

关系数据库已经无法满足 Web2.0 的需求。重要表目前如下几种方面：

- (1) 无法满足海量数据的管理需求
- (2) 无法满足数据高并发的需求
- (3) 无法满足高可扩展性和高可用性的需求

3. 请比较 NoSQL 数据库和关系数据库的优缺陷。

比较原则	RDBMS	NoSQL	备注
数据库原理	完全支持	部分支持	RDBMS 有关系代数理论作为基本 NoSQL 没有统一的理论基本
数据规模	大	超大	RDBMS 很难实现横向扩展，纵向扩展的空间也比较有限，性能会随着数据规模的增大而减少 NoSQL 可以很容易通过添加更多设备来支持更大规模的数据
数据库模式	固定	灵活	RDBMS 需要定义数据库模式，严格遵守数据定义和有关约束条件

			NoSQL 不存在数据库模式，可以自由灵活定义并存储多种不同类型的数据
查询效率	快	可以实现高效的简朴查询，但是不具有高度构造化查询等特性，复杂查询的性能不尽人意	RDBMS 借助于索引机制可以实现迅速查询（涉及记录查询和范畴查询） 诸多 NoSQL 数据库没有面向复杂查询的索引，虽然 NoSQL 可以使用 MapReduce 来加速查询，但是，在复杂查询方面的性能仍然不如 RDBMS
一致性	强一致性	弱一致性	RDBMS 严格遵守事务 ACID 模型，可以保证事务强一致性 诸多 NoSQL 数据库放松了对事务 ACID 四性的规定，而是遵守 BASE 模型，只能保证最后一致性
数据完整性	容易实现	很难实现	

			<p>任何一种 RDBMS 都可以很容易实现数据完整性，例如通过主键或者非空约束来实现实体完整性，通过主键、外键来实现参照完整性，通过约束或者触发器来实现顾客自定义完整性</p> <p>但是，在 NoSQL 数据库却无法实现</p>
扩展性	一般	好	<p>RDBMS 很难实现横向扩展，纵向扩展的空间也比较有限</p> <p>NoSQL 在设计之初就充分考虑了横向扩展的需求，可以很容易通过添加便宜设备实现扩展</p>
可用性	好	较好	<p>RDBMS 在任何时候都以保证数据一致性为优先目的，另一方面才是优化系统性能，随着数据规模的增大，RDBMS 为了保证严格的一致性，只能提供相对较弱的可用性</p> <p>大多数 NoSQL 都能提供较高的可用性</p>

原则化	是	否	RDBMS 已经原则化 (SQL) NoSQL 还没有行业原则，不同的 NoSQL 数据库均有自己的查询语言，很难规范应用程序接口

			StoneBraker 觉得： NoSQL 缺少统一查询语言，将会拖慢 NoSQL 发展
技术支持	高	低	RDBMS 通过几十年的发展，已经非常成熟，Oracle 等大型厂商都可以提供较好的技术支持 NoSQL 在技术支持方面仍然处在起步阶段，还不成熟，缺少有力的技术支持
可维护性	复杂	复杂	RDBMS 需要专门的数据库管理员 (DBA) 维护 NoSQL 数据库虽然没有 DBMS 复杂，也难以维护

5. 试述 NoSQL 数据库的四大类型

答：键值数据库、列族数据库、文档数据库和图数据库

6. 试述键值数据库、列族数据库、文档数据库和图数据库的合用场合和优缺陷。

数据库	合用场合	长处	缺陷
键值数据库	通过键而是通过值来查的业务	扩展性好，灵活性好，大量写操作时性能高	无法存储结构化信息，条件查询效率较低

列族数据库	不需要 ACID 事务支持的情形	查找速度快，可扩展性强，容易进行分布式扩展，复杂性低	功能较少，大都不支持强事务一致性
文档数据库	只在相似的文档上添加事务	性能好（高并发），灵活性高，复杂性低，数据构造灵活 提供嵌入式文档功能，将常常查询的数据存储在同一种文档中 既可以根据键来构建索引，也可以根据内容构建索引	缺少统一的查询语法
图形数据库	具有高度互相关联关系的数据	灵活性高，支持复杂的图形算法，可用于构建复杂的关系图谱	复杂性高，只能支持一定的数据规模

7. 试述 CAP 理论的具体含义。

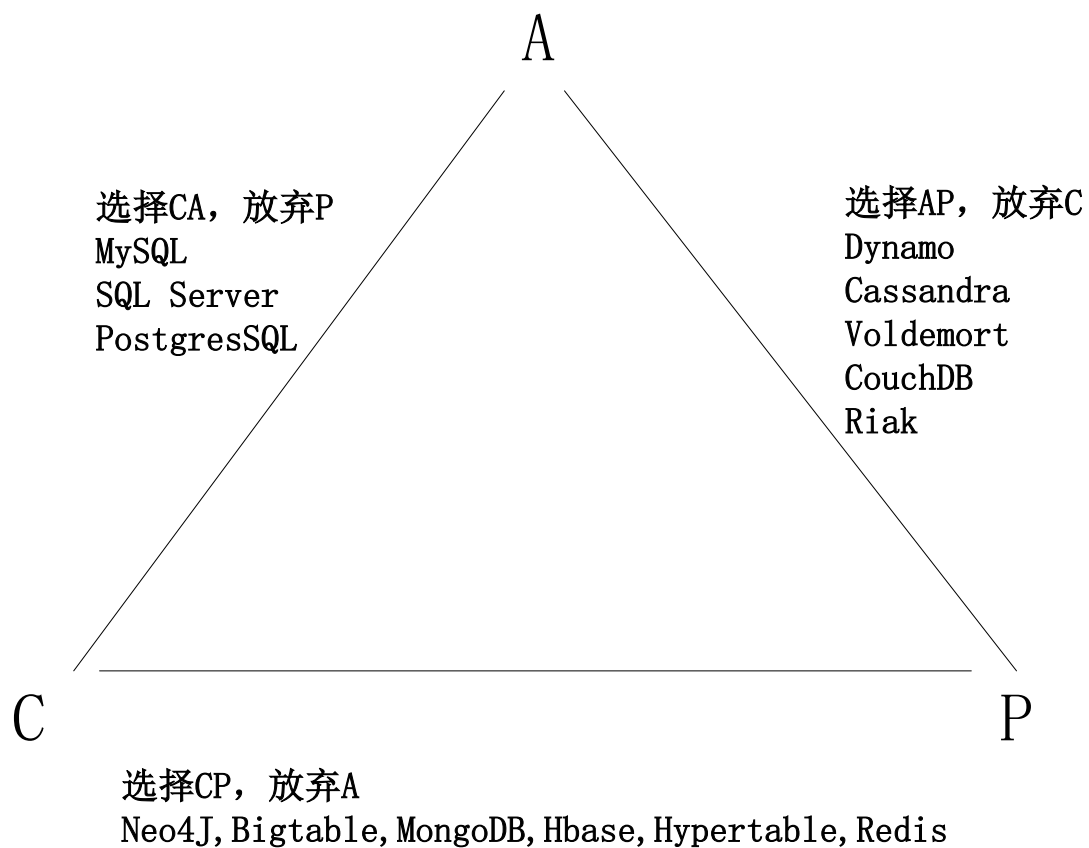
答：所谓的 CAP 指的是：

C (Consistency)：一致性，是指任何一种读操作总是可以读到之前完毕的写操作的成果，也就是在分布式环境中，多点的数据是一致的，或者说，所有节点在同一时间具有相似的数据

A: (Availability)：可用性，是指迅速获取数据，可以在拟定的时间内返回操作成果，保证每个祈求不管成功或者失败均有响应；

P (Tolerance of Network Partition)：分区容忍性，是指当浮现网络分区的状况时（即系统中的一部分节点无法和其他节点进行通信），分离的系统也可以正常运营，也就是说，系统中任意信息的丢失或失败不会影响系统的继续运作。

8. 请举例阐明不同产品在设计时是如何运用 CAP 理论的。



9. 试述数据库的 ACID 四性的含义

1. 原子性 (Atomicity)

指事务必须是原子工作单元，对于其数据修改，要么全都执行，要么全都不执行。

2. 一致性 (consistency)

指事务在完毕时，必须使所有的数据都保持一致状态。

4. 隔离性 (Isolation)

指并发事务所做的修改必须与其他并发事务所做的修改隔离。

5. 持久性 (Durability)

指事务完毕之后，它对于系统的影响是永久性的，该修改虽然浮现致命的系统故障也将始终保持。

10. 试述 BASE 的具体含义

BASE 的基本含义是基本可用 (Basically Available)、软状态 (Soft-state) 和最后一致性 (Eventual consistency)

11. 请解释软状态、无状态、硬状态的具体含义。

“软状态 (soft-state)”是与“硬状态 (hard-state)”相相应的一种提法。数据库保存的数据是“硬状态”时，可以保证数据一致性，即保证数据始终是对的。“软状态”是指状态可以有一段时间不同步，具有一定的滞后性。

12. 什么是最后一致性？

最后一致性根据更新数据后各进程访问到数据的时间和方式的不同，又可以辨别为：

(1) 会话一致性：它把访问存储系统的进程放到会话 (session) 的上下文中，只要会话还存在，系统就保证“读己之所写”一致性。如果由于某些失败情形令会话终结，就要建立新的会话，并且系统保证不会延续到新的会话；

(2) 单调写一致性：系统保证来自同一种进程的写操作顺序执行。系统必须保证这种限度的一致性，否则就非常难以编程了

(3) 单调读一致性：如果进程已经看到过数据对象的某个值，那么任何后续访问都不会返回在那个值之前的值

(4) 因果一致性：如果进程 A 告知进程 B 它已更新了一种数据项，那么进程 B 的后续访问将获得 A 写入的最新值。而与进程 A 无因果关系的进程 C 的访问，仍然遵守一般的最后一致性规则

(5) “读己之所写”一致性：可以视为因果一致性的一种特例。当进程 A 自己执行一种更新操作之后，它自己总是可以访问到更新过的值，绝不会看到旧值

13. 试述不一致性窗口的含义。

所有后续的访问都可以读取到操作 OP 写入的最新值。从 OP 操作完毕到后续访问可以最后读取到 OP 写入的最新值，这之间的时间间隔称为“不一致性窗口”。

14 最后一致性根据更新数据后各进程访问到数据的时间和方式的不同，又可以分为哪些不同类型的一致性？

会话一致性、单调写一致性、单调读一致性、因果一致性和“读己之所写”一致性。

14. 什么是 NewSQL 数据库？

NewSQL 是对多种新的可扩展、高性能数据库的简称，此类数据库不仅具有 NoSQL 对海量数据的存储管理能力，还保持了老式数据库支持 ACID 和 SQL 特性。

第六章

1. 试述云数据库的概念。

答:云数据库是部署和虚拟化在云计算环境中的数据库。云数据库是在云计算的大背景下发展起来的一种新兴的共享基本架构的措施,它极大地增强了数据库的存储能力,消除了人员、硬件、软件的反复配备,让软、硬件升级变得更加容易,同步,也虚拟化了许多后端功能。云数据库具有高可扩展性、高可用性、采用多租形式和支持资源有效分发等特点。

2. 与老式的软件使用方式相比,云计算这种模式具有哪些明显的优势?

项目	传统方式	云计算方式
获得软件的方式	自己投资建设机房,搭建硬件平台,购买软件在本地安装	直接购买云计算厂商的软件服务
使用方式	本地安装,本地使用	软件运行在云计算厂商服务器上,用户在任何有网络接入的地方都可以通过网络使用软件服务
付费方式	需要一次性支付较大的初期投入成本,包括建设机房、配置硬件、购买各种软件(操作系统、杀毒、业务软件等)	零成本投入就可以立即获得所需的 IT 资源,只需要为所使用的资源付费,多用多付,少用少付,极其廉价
维护成本	需要自己花钱聘请专业技术人员维护	零成本,所有维护工作由云计算厂商负责
获得 IT 资源的速度	需要耗费较长时间建设机房、购买和安装调试设备系统	随时可用,购买服务后立即可用
共享方式	自己建设,自给自足	云计算厂商建设好云计算服务平台后,同时为众多用户提供服务
维修速度	出现病毒、系统崩溃等问题时,需要自己聘请 IT 人员维护,很多普通企业的 IT 人员技术能力有限,碰到一些问题甚至需要寻找外援,通常不能立即解决	出现任何系统问题时,云计算厂商都会凭借其专业化团队给出及时响应,确保云服务的正常使用
资源利用率	利用率较低,投入大量资金建设的 IT 系统,往往只供企业自己使用,当企业不需要那么多 IT 资源时,就会产生资源浪费	利用率较高,每天都可以为大量用户提供服务;当存在闲置资源时,云计算管理系统会自动关闭和退出多余资源;当需要增加资源时,又会自动启动和加入相关资源
用户搬迁时的成本	当企业搬家时,原来的机房设施就要作废,需要在新地方重新投入较大成本建设机房	企业无论搬迁到哪里,都可以通过网络重新零成本立即获得云计算服务,因为,资源在云端,不在用户端,用户搬迁不会影响到 IT 资源的分布
资源可拓展性	企业自己建设的 IT 基础设施的服务能力通常是有上限的,当企业业务量突然增加时,现有的 IT 基础设施无法立即满足需求,就需要花费时间和金钱购买和安装新设备;当业务高峰过去时,多余的设备就会闲置,造成资源浪费	云计算厂商可以为企业提供近乎无限的 IT 资源(存储和计算等资源),用户想用多少都可以立即获得,当用户不使用时,只需退订多余资源,不存在任何资源闲置问题

3. 云数据库有哪些特性?

答:1) 动态可扩展 2) 高可用性 3) 较低的使用代价
 4) 易用性 5) 高性能 6) 免维护 7) 安全

4. 试述云数据库的影响。

答：在大数据时代，每个公司几乎每天都在不断产生大量的数据。公司类型不同，对于存储的需求也千差万别，而云数据库可以较好地满足不同公司的个性化存储需求。

一方面，云数据库可以满足大公司的海量数据存储需求。云数据库在目前数据爆炸的大数据时代具有广阔的应用前景。老式的关系数据库难以水平扩展，根本无法存储如此海量的数据。因此，具有高可扩展性的云数据库就成为公司海量数据存储管理的较好选择。

另一方面，云数据库可以满足中小公司的低成本数据存储需求。中小公司在 IT 基本设施方面的投入比较有限，非常渴望从第三方以便、快捷、便宜地获得数据库服务。云数据库采用多租户方式同步为多种顾客提供服务，减少了单个顾客的使用成本，并且顾客使用云数据库服务一般按需付费，不会挥霍资源导致额外支出，因此，云数据库使用成本很低，对于中小公司而言可以大大减少公司的信息化门槛，让公司在付出较低成本的同时，获得优质的专业级数据库服务，从而有效提高公司信息化水平。

此外，云数据库可以满足公司动态变化的数据存储需求。公司在不同时期需要存储的数据量是不断变化的，有时增长，有时减少。在小规模应用的状况下，系统负载的变化可以由系统空闲的多余资源来解决，但是，在大规模应用的状况下，老式的关系数据库由于其伸缩性较差，不仅无法满足应用需求，并且会给公司带来高昂的存储成本和管理开销。而云数据库的良好伸缩性，可以让公司在需求增长时立即获得数据库能力的提高，在需求减少时立即释放多余的数据库能力，较好地满足公司的动态数据存储需求。

5. 举例阐明云数据库厂商及其代表性产品。

答：云数据库供应商重要分为三类。

1) 老式的数据库厂商，如 Teradata、Oracle、IBM DB2 和 Microsoft SQL Server 等。

2) 涉足数据库市场的云供应商，如 Amazon、Google、Yahoo!、阿里、百度、腾讯等。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/977151001161006123>

3)