MHBD 考试核心知识点 / Ключевые моменты для экзамена

> 8小时速成版 🚀 俄中双语+口语表达

---

1. Что такое большие данные. 3V, 4V

什么是大数据。3V, 4V模型

📝 核心知识点 / Ключевые моменты:

Большие данные (大数据) = данные, объем которых превышает возможности обычных баз данных.
大数据 = 数据量超过普通数据库处理能力的数据集

3V 模型 / Модель 3V:

V	Русский	中文	Пример / 例子
Volume	Объём	数据量	ТБ → ПБ → ЗБ
Variety	Разнообразие	多样性	структурированные(结构化), полуструктурированные(半结构化), неструктурированные(非结构化)
Velocity	Скорость	速度	обработка в реальном времени (实时处理)

4V = 3V + Veracity (Достоверность / 真实性)

- Veracity = точность и чистота данных (数据的准确性和洁净度)

🗣️ 口语回答 / Как отвечать:

Русский: > Большие данные — это очень большие объемы данных, которые нельзя обработать обычными базами данных. Для описания больших данных используется модель 3V: Volume — это объём данных, от терабайтов до петабайтов. Variety — это разнообразие данных: структурированные, полуструктурированные и неструктурированные. Velocity — это скорость

поступления данных, нужна обработка в реальном времени. В модели 4V добавляется Veracity — достоверность, то есть точность и качество данных.

中文: > 大数据是指数据量非常大，普通数据库无法处理的数据。我们用3V模型来描述大数据：Volume是数据量，从TB到PB级别。Variety是多样性，包括结构化、半结构化和非结构化数据。Velocity是速度，需要实时处理。4V模型增加了Veracity——真实性，就是数据的准确性和质量。

---

2. Сетевые и распределенные файловые системы

网络和分布式文件系统

📝 **核心知识点 / Ключевые моменты:**

Ключевая архитектура / 核心架构: Shared Nothing (SNA) = каждый узел независим (своя память, диски) 每个节点独立（有自己的内存、磁盘）

Примеры / 例子: AFS, GFS, **HDFS**

HDFS 核心 / Ключевые характеристики HDFS:

Параметр / 参数	Значение / 值
Блоки / 数据块	64 МБ (рекомендуется 128 МБ)
Репликация / 副本	3 копии (по умолчанию) / 默认3个副本
Отказоустойчивость / 容错	копии на разных серверах/стойках / 副本分布在不同服务器/机架

🗣️ **口语回答 / Как отвечать:**

Русский: > Распределенные файловые системы хранят данные на нескольких узлах. Главная архитектура — Shared Nothing, где каждый узел независим и имеет свою память и диски. Примеры таких систем: AFS, GFS и HDFS. HDFS — основная система в экосистеме больших данных. Файлы делятся на блоки по 128 мегабайт. Для надежности каждый блок копируется 3 раза на разные серверы. Если один сервер сломается, данные не потеряются.

中文: > 分布式文件系统把数据存储在多个节点上。主要架构是Shared Nothing，每个节点独立，有自己的内存和磁盘。例子有AFS、GFS和HDFS。HDFS是大数据生态系统的核心。文件被分成128MB的块。为了可靠性，每个块复制3份到不同服务器。如果一个服务器坏了，数据也不会丢失。

---

3. Распределенные базы данных

分布式数据库

📝 核心知识点 / Ключевые моменты:

Архитектура / 架构: Shared Nothing (SNA) — каждый узел самодостаточен 每个节点自给自足

HBase:

Характеристика / 特性	Описание / 描述
Модель / 模型	Разреженное многомерное отображение / 稀疏多维映射
Скорость / 速度	Миллионы запросов/сек / 每秒数百万请求
Объем / 容量	До **ПБ** данных / 可达PB级
Основа / 基础	Работает на **HDFS**

MongoDB:

Характеристика / 特性	Описание / 描述
Тип / 类型	Документоориентированная БД / 面向文档数据库
Схема / 模式	Без строгой схемы (schemaless) / 无固定模式
Формат / 格式	JSON/BSON

Компоненты / 组件:

- **Master** = управление регионами / 管理区域
- **ZooKeeper** = синхронизация и мониторинг / 同步和监控
- **Репликация** = защита от потерь / 防止数据丢失

🗣️ 口语回答 / Как отвечать:

Русский: > Распределенные базы данных используют архитектуру Shared Nothing, где каждый узел независим. Есть два главных примера: HBase и MongoDB. HBase — это разреженное многомерное хранилище, работает на HDFS, может хранить петабайты данных и обрабатывать миллионы запросов в секунду. MongoDB — документоориентированная база данных, хранит данные в формате JSON, не требует строгой схемы. Для управления используется Master-сервер и ZooKeeper для синхронизации.

中文: > 分布式数据库使用Shared Nothing架构，每个节点都是独立的。有两个主要例子：HBase和MongoDB。HBase是稀疏多维存储，运行在HDFS上，可以存储PB级数据，每秒处理数百万请求。MongoDB是面向文档的数据库，用JSON格式存储数据，不需要固定模式。系统用Master服务器管理，ZooKeeper负责同步。

---

4. HDFS (Hadoop Distributed File System)

Hadoop分布式文件系统

📝 核心知识点 / Ключевые моменты:

Компонент / 组件	Функция / 功能
NameNode	Главный узел, хранит метаданные / 主节点，存储元数据
DataNode	Хранит блоки данных / 存储数据块

Ключевые параметры / 关键参数:

Параметр / 参数	Значение / 值
Размер блока / 块大小	**128 МБ** (рекомендуется)
Репликация / 副本数	**3 копии** по умолчанию
Стратегия / 策略	минимум 1 копия в **другой стойке** / 至少1份在不同机架

Команды / 命令:

```
hadoop fs -ls          # список файлов / 列出文件
hadoop fs -put         # загрузка / 上传
hadoop fs -cat         # просмотр / 查看内容
hadoop fs -setrep      # установка репликации / 设置副本数
```

🗣️ 口语回答 / Как отвечать:

Русский: > HDFS — это распределенная файловая система Hadoop. В ней есть два типа узлов: NameNode — главный узел, который хранит метаданные, и DataNode — узлы, которые хранят сами данные. Файлы делятся на блоки размером 128 мегабайт. Каждый блок копируется 3 раза для надежности. Важно, что одна копия обязательно хранится на другой стойке. Так если одна стойка сломается, данные останутся. Для работы с файлами используются команды: hadoop fs -ls для списка, -put для загрузки, -cat для просмотра.

中文: > HDFS是Hadoop的分布式文件系统。有两种节点：NameNode是主节点，存储元数据；DataNode存储实际数据。文件被分成128MB的块。每个块复制3份保证可靠性。重要的是，至少有一份存在不同的机架上。这样即使一个机架坏了，数据还在。操作文件用命令：hadoop fs -ls列出文件，-put上传，-cat查看内容。

---

5. Отличия систем хранения больших данных

大数据存储系统的区别（可扩展性、容错性）

📝 **核心知识点 / Ключевые моменты:**

Масштабируемость / 可扩展性:

Система / 系统	Объем / 容量
РСУБД (传统数据库)	ТБ (терабайты)
Big Data (大数据)	**ПБ** (петабайты)

- Горизонтальное масштабирование = добавление узлов без перенастройки
- 水平扩展 = 添加节点不需要重新配置

Отказоустойчивость / 容错性:

Механизм / 机制	Описание / 描述
Репликация	3 копии блоков / 3个副本
Стойки	Копии на **разных стойках** / 副本在不同机架
Восстановление	**Автоматическое** при сбоях / 故障时自动恢复

🗣️ **口语回答 / Как отвечать:**

Русский: > Системы больших данных отличаются от обычных баз данных по двум главным параметрам. Первое — масштабируемость: обычные базы работают с терабайтами, а большие данные — с петабайтами. При этом используется горизонтальное масштабирование — можно добавлять новые серверы без перенастройки системы. Второе — отказоустойчивость: данные копируются 3 раза на разные серверы и разные стойки. Если сервер сломается, система автоматически восстановит копии на других серверах.

中文: > 大数据系统和普通数据库有两个主要区别。第一是可扩展性：普通数据库处理TB级数据，大数据处理PB级。使用水平扩展——可以添加新服务器而不需要重新配置。第二是容错性：数据复制3份到不同服务

器和机架。如果服务器坏了，系统会自动在其他服务器上恢复副本。

---

6. Зачем нужны NoSQL

为什么需要NoSQL数据库

📝 核心知识点 / Ключевые моменты:

Проблема РСУБД / 传统DB问题	Решение NoSQL / NoSQL解决方案
Только ТБ данных / 只能处理TB	До **ПБ** данных / 可达PB级
Тысячи запросов/сек / 每秒千次	**Миллионы** запросов/сек / 每秒百万次
Строгая схема / 固定模式	**Гибкая** схема (schemaless) / 灵活模式
Вертикальное масштабирование / 垂直扩展	**Горизонтальное** масштабирование / 水平扩展

Компромисс / 权衡: NoSQL часто отказывается от полного ACID NoSQL通常放弃完整的ACID支持

🗣️ 口语回答 / Как отвечать:

Русский: > NoSQL базы данных нужны потому, что обычные реляционные базы не справляются с большими данными. У обычных баз есть ограничения: они работают только с терабайтами, обрабатывают тысячи запросов в секунду, требуют строгую схему. NoSQL решает эти проблемы: может хранить петабайты данных, обрабатывать миллионы запросов, использовать гибкую схему без строгого описания таблиц. Главное преимущество — горизонтальное масштабирование. Но есть компромисс: NoSQL обычно не поддерживает полный ACID.

中文: > 需要NoSQL数据库是因为传统关系型数据库无法处理大数据。传统数据库有限制：只能处理TB级数据，每秒处理几千个请求，需要固定模式。NoSQL解决了这些问题：可以存储PB级数据，每秒处理数百万请求，使用灵活模式不需要严格定义表结构。主要优势是水平扩展。但有个权衡：NoSQL通常不支持完整的ACID。

---

7. Виды NoSQL баз данных

NoSQL数据库的类型

📝 核心知识点 / Ключевые моменты:

Тип / 类型	Пример / 例子	Особенность / 特点
Column-family (列族)	HBase	Разреженное хранилище, ПБ данных / 稀疏存储，PB级
Document (文档)	MongoDB	JSON документы, без схемы / JSON文档，无模式
Graph (图)	Neo4j	Узлы + связи, язык Cypher / 节点+关系，Cypher语言

🗣️ 口语回答 / Как отвечать:

Русский: > Есть три основных типа NoSQL баз данных. Первый — Column-family, или столбцовые хранилища. Пример — HBase. Это разреженное многомерное хранилище, которое может работать с петабайтами данных. Второй тип — документоориентированные базы. Пример — MongoDB. Она хранит данные в формате JSON и не требует строгой схемы. Третий тип — графовые базы данных. Пример — Neo4j. Она хранит узлы и связи между ними, использует специальный язык запросов Cypher.

中文: > NoSQL数据库有三种主要类型。第一种是列族存储，例如HBase。这是稀疏多维存储，可以处理PB级数据。第二种是面向文档的数据库，例如MongoDB。它用JSON格式存储数据，不需要固定模式。第三种是图数据库，例如Neo4j。它存储节点和节点之间的关系，使用专门的查询语言Cypher。

---

8. Модель данных HBase

HBase数据模型

📝 核心知识点 / Ключевые моменты:

Определение / 定义: Разреженное, распределенное, многомерное отсортированное отображение
稀疏的、分布式的、多维有序映射

Структура / 结构:

```
Row Key → Column Family → Column Qualifier → Timestamp → Value
行键 → 列族 → 列修饰符 → 时间戳 → 值
```

Элемент / 元素	Описание / 描述
Row Key (行键)	Первичный ключ, сортировка лексикографическая / 主键，字典序排序
Column Family (列族)	Группа столбцов (задается при создании) / 列的组（创建时定义）
Column Qualifier (列修饰符)	Конкретное поле (динамическое) / 具体字段（动态的）
Timestamp (时间戳)	Версионирование данных / 数据版本控制

HBase vs РСУБД / HBase对比传统数据库:

Параметр / 参数	HBase	РСУБД
Транзакции / 事务	ACID только для 1 строки / 仅单行ACID	Полный ACID / 完整ACID
Запросы / 查询	get/put/scan	SQL
Индексы / 索引	Только Row Key / 仅行键	Множество / 多个

Хранение / 存储: HDFS → формат **HFile Запись / 写入:** WAL → MemStore → Диск

🗣️ 口语回答 / Как отвечать:

Русский: > Модель данных HBase — это разреженное, распределенное, многомерное отсортированное отображение. Структура такая: Row Key — это первичный ключ, данные сортируются по нему. Column Family — группа столбцов, задается при создании таблицы. Column Qualifier — конкретное поле внутри семейства. Timestamp позволяет хранить несколько версий одних данных. В отличие от обычных баз, HBase поддерживает ACID только для одной строки, вместо SQL использует операции get, put и scan. Данные хранятся на HDFS в формате HFile.

中文: > HBase的数据模型是稀疏的、分布式的、多维有序映射。结构是这样的：Row Key是主键，数据按它排序。Column Family是列族，在创建表时定义。Column Qualifier是列族里的具体字段。Timestamp让我们可以存储同一数据的多个版本。和普通数据库不同，HBase只支持单行的ACID，用get、put、scan操作代替SQL。数据存储在HDFS上，格式是HFile。

---

9. MapReduce

MapReduce分布式计算模型

📝 核心知识点 / Ключевые моменты:

Определение / 定义: Парадигма распределенных вычислений на основе пар ключ/значение 基于键值对的分布式计算范式

3 этапа / 三个阶段:

```
Map:    <k1,v1> → list(<k2,v2>)        # фильтрация/преобразование 过滤/转换
Shuffle: группировка по ключу          # автоматически 自动按键分组
Reduce: <k2,list(v2)> → list(<k3,v3>)  # агрегация 聚合
```

Важно / 重要: Reduce запускается ПОСЛЕ завершения всех Map Reduce只有在所有Map完成后才启动

Combiner = локальная агрегация на узле (оптимизация) Combiner = 节点上的本地聚合（优化）

Пример Word Count / 词频统计例子:

```
Map:    "hello world" → [(hello,1), (world,1)]
Reduce: (hello, [1,1,1]) → (hello, 3)
```

🗣️ 口语回答 / Как отвечать:

Русский: > MapReduce — это модель распределенных вычислений, которая работает с парами ключ-значение. Есть три этапа. Первый — Map: получает входные данные и создает промежуточные пары ключ-значение. Например, для подсчета слов слово “hello” превращается в пару (hello, 1). Второй этап — Shuffle: система автоматически группирует все значения с одинаковым ключом. Третий — Reduce: получает ключ и список значений, делает агрегацию. Важно помнить: Reduce запускается только после завершения всех Map задач. Для оптимизации можно использовать Combiner — это локальная агрегация на узле.

中文: > MapReduce是一种分布式计算模型，用键值对工作。有三个阶段。第一个是Map：接收输入数据，产生中间键值对。比如统计词频时，“hello”变成(hello, 1)。第二阶段是Shuffle：系统自动把相同键的值分组。第三个是Reduce：接收键和值列表，进行聚合。重要的是：Reduce只在所有Map任务完成后才启动。为了优化可以用Combiner——这是节点上的本地聚合。

---

10. Spark

Spark技术

📝 核心知识点 / Ключевые моменты:

Позиция / 定位: Часть экосистемы Hadoop / Hadoop生态系统的一部分

Главное преимущество / 主要优势: In-memory computing (вычисления в памяти / 内存计算)

Использование / 应用场景: | Сценарий / 场景 | Описание / 描述 | |---|---| | Потоковая обработка | Streaming / 流处理 | | Real-time аналитика | Анализ в реальном времени / 实时分析 | | Итеративные задачи | Быстрее MapReduce / 比MapReduce快 |

🗣️ 口语回答 / Как отвечать:

Русский: > Spark — это технология обработки данных, которая является частью экосистемы Hadoop. Главное преимущество Spark — вычисления в оперативной памяти, то есть In-memory computing. Это делает Spark намного быстрее, чем MapReduce, особенно для итеративных задач. Spark используется для потоковой обработки данных, аналитики в реальном времени и продвинутого анализа. Он может работать вместе с HDFS, Hive и другими компонентами Hadoop.

中文: > Spark是Hadoop生态系统中的数据处理技术。Spark的主要优势是内存计算（In-memory computing）。这使得Spark比MapReduce快得多，特别是对于迭代任务。Spark用于流数据处理、实时分析和高级分析。它可以和HDFS、Hive等Hadoop组件一起工作。

---

11. Языки высокого уровня над Hadoop

Hadoop上的高级编程语言

📝 核心知识点 / Ключевые моменты:

Язык / 语言	Описание / 描述
Pig	Скриптовый язык для MapReduce / MapReduce的脚本语言
Hive	SQL-подобный язык (HiveQL) / 类SQL语言
Spark	Продвинутая аналитика / 高级分析

Суть / 本质: Все транслируются в MapReduce задачи 所有这些最终都转换成MapReduce任务

🗣️ 口语回答 / Как отвечать:

Русский: > Над Hadoop существуют языки высокого уровня, которые упрощают работу с большими данными. Pig — это скриптовый язык, который позволяет писать логику обработки данных проще, чем на чистом MapReduce. Hive предоставляет SQL-подобный интерфейс — язык HiveQL. Это удобно для тех, кто знает SQL. Spark используется для продвинутой аналитики. Главное — все эти языки в итоге транслируются в задачи MapReduce или подобные распределенные процессы.

中文: > Hadoop上有高级语言，简化大数据处理工作。Pig是脚本语言，比直接写MapReduce更简单。Hive提供类SQL接口——HiveQL语言。对于懂SQL的人很方便。Spark用于高级分析。关键是：这些语言最终都转换成MapReduce任务或类似的分布式处理。

---

12. Что такое Hadoop. Компоненты

什么是Hadoop。主要组件

📝 核心知识点 / Ключевые моменты:

Hadoop = стек технологий для распределенного хранения и обработки данных Hadoop = 分布式存储和处理数据的技术栈

Архитектура / 架构: Shared Nothing (SNA)

Основные компоненты / 主要组件:

Компонент / 组件	Функция / 功能
HDFS	Распределенная файловая система / 分布式文件系统
MapReduce	Модель вычислений / 计算模型
YARN	Управление ресурсами / 资源管理

🗣️ 口语回答 / Как отвечать:

Русский: > Hadoop — это технологический стек для распределенного хранения и обработки больших объемов данных. Он основан на архитектуре Shared Nothing, где каждый узел независим. У Hadoop три основных компонента. Первый — HDFS, распределенная файловая система для хранения данных. Второй — MapReduce, модель распределенных вычислений на основе пар ключ-значение. Третий — YARN, система управления ресурсами кластера и планирования задач. Эти компоненты работают вместе для обработки петабайтов данных.

中文: > Hadoop是用于分布式存储和处理大数据的技术栈。它基于Shared Nothing架构，每个节点都是独立的。Hadoop有三个主要组件。第一个是HDFS，分布式文件系统，用于存储数据。第二个是MapReduce，

基于键值对的分布式计算模型。第三个是YARN，管理集群资源和任务调度的系统。这些组件一起工作，可以处理PB级数据。

---

13. Узлы Hadoop кластера

Hadoop集群的节点

📝 核心知识点 / Ключевые моменты:

Для хранения (HDFS) / 存储节点:

Узел / 节点	Роль / 角色
NameNode	Главный, метаданные, расположение блоков / 主节点，元数据，块位置
DataNode	Физическое хранение блоков / 物理存储数据块

Для вычислений (Hadoop 1) / 计算节点:

Узел / 节点	Роль / 角色
JobTracker	Прием заданий, распределение задач / 接收作业，分配任务
TaskTracker	Выполнение Map/Reduce задач / 执行Map/Reduce任务

🗣️ 口语回答 / Как отвечать:

Русский: > В кластере Hadoop есть несколько типов узлов. Для хранения данных используются NameNode и DataNode. NameNode — главный узел, который хранит метаданные и информацию о расположении блоков данных. DataNode — узлы, на которых физически хранятся блоки данных. Для вычислений в Hadoop 1 использовались JobTracker и TaskTracker. JobTracker принимает задания от пользователей и распределяет задачи. TaskTracker выполняет конкретные операции Map или Reduce на узлах.

中文: > Hadoop集群有几种类型的节点。存储数据用NameNode和DataNode。NameNode是主节点，存储元数据和数据块的位置信息。DataNode是物理存储数据块的节点。计算方面，Hadoop 1用JobTracker和TaskTracker。JobTracker接收用户的作业并分配任务。TaskTracker在节点上执行具体的Map或Reduce操作。

---

14. Отказоустойчивость узлов

节点的容错性（节点故障处理）

📝 核心知识点 / Ключевые моменты:

DataNode упал / DataNode故障:

Механизм / 机制	Описание / 描述
✅ Данные НЕ теряются	3 реплики / 3个副本，数据不丢失
✅ Автовосстановление	Автоматическое восстановление копий / 自动恢复副本
🔍 Обнаружение	Через **heartbeat** / 通过心跳信号发现

NameNode/JobTracker упал / NameNode/JobTracker故障:

Механизм / 机制	Описание / 描述
✅ **Backup Master**	Резервные серверы / 备用服务器
✅ **ZooKeeper**	Обнаруживает сбой / 检测故障
✅ Переключение	Резервный становится активным / 备用变主用

TaskTracker упал / TaskTracker故障:

Механизм / 机制	Описание / 描述
✅ Переназначение	Задача переходит другому узлу / 任务转给其他节点
🔍 Обнаружение	Через **heartbeat** / 通过心跳信号

🗣️ 口语回答 / Как отвечать:

Русский: > Система Hadoop устойчива к сбоям узлов. Если упадет DataNode — данные не потеряются, потому что есть 3 копии каждого блока. Система обнаруживает сбой через сигналы heartbeat и автоматически восстанавливает копии на других серверах. Если упадет NameNode или JobTracker — есть резервные Master-серверы. ZooKeeper следит за состоянием и при сбое переключает на резервный сервер. Если упадет TaskTracker — задача просто переназначается другому узлу. Так система продолжает работать даже при сбоях.

中文: > Hadoop系统对节点故障有容错能力。如果DataNode坏了——数据不会丢失，因为每个块有3个副本。系统通过heartbeat信号发现故障，自动在其他服务器上恢复副本。如果NameNode或JobTracker坏了——有备用Master服务器。ZooKeeper监控状态，故障时切换到备用服务器。如果TaskTracker坏了——任务直接分配给其他节点。这样即使有故障，系统也能继续工作。

---

15. Проблемы Hadoop 1 → Отличия Hadoop 2

Hadoop 1的问题 → Hadoop 2的区别

📝 **核心知识点 / Ключевые моменты:**

Проблемы Hadoop 1 / Hadoop 1的问题:

Проблема / 问题	Описание / 描述
SPOF	NameNode = единая точка отказа / 单点故障
Перегрузка	JobTracker перегружен / JobTracker负载过重
Ограничения	Только MapReduce / 只支持MapReduce

Решения в Hadoop 2 / Hadoop 2的解决方案:

Проблема / 问题	Решение / 解决
SPOF	**HA** (High Availability) + резервные Master / 高可用+备用Master
Перегрузка	**YARN** для управления ресурсами / YARN资源管理
Ограничения	Поддержка **Spark, Hive, Pig** / 支持多引擎

🗣️ **口语回答 / Как отвечать:**

Русский: > В Hadoop 1 было три главные проблемы. Первая — SPOF, единая точка отказа: если NameNode падает, весь кластер останавливается. Вторая — JobTracker был перегружен, потому что отвечал и за ресурсы, и за задачи. Третья — поддерживался только MapReduce. Hadoop 2 решил эти проблемы. Для SPOF добавили High Availability — резервные Master-серверы с ZooKeeper. Для перегрузки создали YARN — отдельную систему управления ресурсами. Теперь можно запускать не только MapReduce, но и Spark, Hive, Pig на одном кластере.

中文: > Hadoop 1有三个主要问题。第一是SPOF单点故障：如果NameNode坏了，整个集群停止工作。第二是JobTracker负载过重，因为它既管资源又管任务。第三是只支持MapReduce。Hadoop 2解决了这些问

题。对于SPOF，增加了高可用——备用Master服务器和ZooKeeper。对于负载问题，创建了YARN——独立的资源管理系统。现在可以在同一个集群上运行MapReduce、Spark、Hive、Pig等多种引擎。

---

16. Безопасность: CIA (Конфиденциальность, Целостность, Доступность)

大数据安全：CIA（机密性、完整性、可用性）

📝 **核心知识点 / Ключевые моменты:**

Конфиденциальность (机密性) / Confidentiality:

Аспект / 方面	Описание / 描述
Цель / 目标	Защита от несанкционированного доступа / 防止未授权访问
Механизм / 机制	Аутентификация + Авторизация / 身份验证+授权
Данные / 数据	Оборона, медицина, финансы / 国防、医疗、金融

Целостность (完整性) / Integrity:

Аспект / 方面	Описание / 描述
Цель / 目标	Защита от изменений / 防止数据被篡改
HBase	ACID для одной строки / 单行ACID
WAL	Лог для восстановления / 恢复日志

Доступность (可用性) / Availability:

Аспект / 方面	Описание / 描述
Репликация / 副本	3 копии / 3个副本
Восстановление / 恢复	Автоматическое / 自动恢复
Производительность / 性能	Миллионы запросов/сек / 每秒百万请求

🗣️ 口语回答 / Как отвечать:

Русский: > Информационная безопасность больших данных основана на трех принципах — CIA. Конфиденциальность — защита данных от несанкционированного доступа. Это важно для оборонных, медицинских и финансовых данных. Используются аутентификация и авторизация. Целостность — защита от несанкционированных изменений. В HBase ACID поддерживается для одной строки, WAL-лог защищает данные при сбоях. Доступность — данные должны быть доступны когда нужно. Это обеспечивается репликацией — 3 копии данных, автоматическим восстановлением при сбоях.

中文: > 大数据信息安全基于三个原则——CIA。机密性是防止未授权访问数据。这对国防、医疗、金融数据很重要。使用身份验证和授权。完整性是防止数据被非法修改。HBase对单行支持ACID，WAL日志在故障时保护数据。可用性是数据在需要时必须可用。通过3个副本复制和故障时自动恢复来保证。

---

17. Безопасность: AAA (Аутентификация, Авторизация, Аудит)

大数据安全：AAA（身份验证、授权、审计）

📝 核心知识点 / Ключевые моменты:

Аутентификация (身份验证) / Authentication:

Вопрос / 问题	Описание / 描述
“Кто ты?”	Проверка личности пользователя / 验证用户身份
“你是谁？”	Подтверждение идентичности / 确认身份

Авторизация (授权) / Authorization:

Вопрос / 问题	Описание / 描述
“Что тебе разрешено?”	Проверка прав доступа / 检查访问权限
“你能做什么？”	Контроль доступа к ресурсам / 控制资源访问

Аудит (审计) / Audit:

Механизм / 机制	Описание / 描述
WAL	Лог всех операций / 所有操作的日志
ZooKeeper	Мониторинг состояния / 状态监控
Цель / 目标	Отслеживание всех действий / 跟踪所有活动

🗣️ 口语回答 / Как отвечать:

Русский: > Безопасность больших данных также включает три компонента — AAA. Аутентификация отвечает на вопрос “кто ты?” — это проверка личности пользователя. Авторизация отвечает на вопрос “что тебе разрешено?” — это проверка прав доступа к конкретным данным. Аудит — это отслеживание всех действий в системе. В HBase для аудита используется WAL-лог, который записывает все операции. ZooKeeper мониторит состояние всех серверов. Это позволяет узнать, кто, когда и что делал с данными.

中文: > 大数据安全还包括三个组件——AAA。身份验证回答”你是谁？“——验证用户身份。授权回答”你能做什么？“——检查对特定数据的访问权限。审计是跟踪系统中的所有活动。HBase用WAL日志记录所有操作来审计。ZooKeeper监控所有服务器的状态。这样可以知道谁、什么时候、对数据做了什么。

---

18. Безопасность в Hadoop

Hadoop中的安全

📝 核心知识点 / Ключевые моменты:

Механизмы защиты / 保护机制:

Механизм / 机制	Описание / 描述
Аутентификация	Проверка пользователя / 验证用户
Авторизация	Контроль доступа / 控制访问
Репликация	3 копии данных / 3份数据副本
WAL	Лог для восстановления / 恢复日志
ZooKeeper	Координация и мониторинг / 协调和监控
Backup Master	Резервные узлы / 备用节点

Применение / 应用:

Область / 领域	Описание / 描述
Антифрод	Выявление мошенничества / 检测欺诈
Защита данных / 数据保护	Оборона, медицина, финансы / 国防、医疗、金融

🗣️ 口语回答 / Как отвечать:

Русский: > Безопасность в Hadoop обеспечивается несколькими механизмами. Аутентификация и авторизация проверяют пользователей и их права доступа. Репликация — 3 копии данных защищают от потери при сбоях. WAL-лог записывает все изменения для восстановления данных. ZooKeeper координирует работу кластера и следит за состоянием узлов. Backup Master-серверы обеспечивают работу при сбое главного узла. Hadoop используется для защиты важных данных — оборонных, медицинских, финансовых. Также применяется для антифрода — выявления мошенничества в транзакциях.

中文: > Hadoop的安全通过几种机制保证。身份验证和授权检查用户及其访问权限。副本机制——3份数据副本防止故障时丢失。WAL日志记录所有更改用于数据恢复。ZooKeeper协调集群工作并监控节点状态。备用Master服务器保证主节点故障时系统继续工作。Hadoop用于保护重要数据——国防、医疗、金融数据。还用于反欺诈——检测交易中的欺诈行为。

🎯 快速记忆表 / Быстрая шпаргалка

Термин / 术语	Значение / 含义
SNA	Shared Nothing Architecture / 无共享架构
HDFS	Hadoop Distributed File System / Hadoop分布式文件系统
3V	Volume, Variety, Velocity / 数据量、多样性、速度
4V	3V + Veracity / 3V + 真实性
NameNode	Главный узел (метаданные) / 主节点（元数据）
DataNode	Хранение данных / 数据存储节点
YARN	Управление ресурсами (Hadoop 2) / 资源管理
WAL	Write-Ahead Log / 预写日志
HFile	Формат хранения HBase / HBase存储格式
HA	High Availability / 高可用性
SPOF	Single Point of Failure / 单点故障
CIA	Конфиденциальность, Целостность, Доступность / 机密性、完整性、可用性
AAA	Аутентификация, Авторизация, Аудит / 身份验证、授权、审计

📊 必记数字 / Числа для запоминания

Параметр / 参数	Значение / 值
Размер блока HDFS / HDFS块大小	**128 МБ**
Репликация / 副本数	**3 копии / 3份**
HBase пропускная способность / HBase吞吐量	**миллионы** запросов/сек / 每秒**百万**请求
HBase объем / HBase容量	до **ПБ** / 可达**PB**级
РСУБД объем / 传统DB容量	до **ТБ** / 可达**TB**级

Удачи на экзамене! 祝考试顺利！ 💪