Spark 생태계 강의 자료

Slide 1:

Slide 2

Slide 3 - 목차

오늘 강의 내용은 주로 네 부분으로 구성됩니다:

- **Spark 개요** - Spark의 기본 개념과 특징을 이해합니다
- **Spark 생태계** - Spark의 풍부한 컴포넌트 생태계를 중점적으로 소개합니다
- **Spark 실행 아키텍처** - Spark의 실행 메커니즘을 깊이 이해합니다
- **Spark 배포 방식** - 다양한 배포 모드를 학습합니다

Slide 4 - 2.2 Spark 생태계

다음으로 2.2절 - Spark 생태계를 중점적으로 학습하겠습니다.
이것은 Spark가 왜 이렇게 강력하고 인기 있는지를 이해하는 핵심 부분입니다.

Slide 1:

Slide 2:

Slide 3: 目录

今天的课程内容主要分为四个部分：

1. Spark概述 - 我们将了解Spark的基本概念和特点
2. Spark生态系统 - 重点介绍Spark丰富的组件生态
3. Spark运行架构 - 深入理解Spark的运行机制
4. Spark的部署方式 - 学习不同的部署模式

Slide 4: 2.2 Spark生态系统

接下来我们重点学习2.2节 - Spark生态系统。
这是理解Spark为什么如此强大和受欢迎的关键部分。

Slide 5 - 왜 생태계를 구축하는가

먼저 하나의 질문을 생각해봅시다:

왜 생태계를 구축해야 하는가?

이 질문은 매우 중요합니다. 왜냐하면 Spark의 핵심 설계 이념과 관련이 있기 때문입니다.

Spark가 단일 컴퓨팅 엔진에서 완전한 오픈소스 생태계로 발전한 것은 우연이 아닙니다.

Slide 6 - 다양한 응용 시나리오의 시간 범위

다양한 유형의 빅데이터 응용 시나리오를 살펴봅시다:

- **복잡한 배치 데이터 처리**: 일반적으로 수십 분에서 수 시간이 필요
- **히스토리 데이터 기반 대화형 쿼리**: 시간 범위는 수십 초에서 수 분
- **실시간 데이터 스트림 처리**: 수백 밀리초에서 수 초 이내 완료 필요

이러한 서로 다른 시간 범위 요구사항은 다양한 기술 솔루션을 필요로 합니다.

Slide 5: 为什么要构建生态系统

首先我们要思考一个问题：

为什么要构建一个生态系统？

这个问题非常重要，因为它关系到Spark的核心设计理念。

我们可以看到，Spark从一个单一的计算引擎发展成为一个完整的开源生态系统，这并不是偶然的。

Slide 6: 不同应用场景的时间跨度

让我们看看不同类型的大数据应用场景：

- 复杂的批量数据处理：通常需要数十分钟到数小时的时间
- 基于历史数据的交互式查询：时间跨度(**kuàdù**)在数十秒到数分钟
- 基于实时数据流的数据处理：要求在数百毫秒到数秒内完成

这些不同的时间跨度需求，促使我们需要不同的技术解决方案。

Slide 7 - 전통적인 분리형 솔루션

Spark가 출현하기 전에는 일반적으로 분리된 솔루션을 사용했습니다:

- 배치 처리: **Hadoop MapReduce**
- 대화형 쿼리: **Apache 프로젝트**
- 스트림 처리: **Apache Storm**

이러한 분리형 아키텍처는 몇 가지 문제점이 있습니다.

Slide 8 - 원활한 공유 불가

첫 번째 문제는 **원활한 공유가 불가능**하다는 것입니다.

서로 다른 기술 스택을 사용할 때, 데이터는 Storm, Apache, Hadoop MapReduce 간에 포맷 변환이 필요하며, 이는 시스템의 복잡성을 증가시킵니다.

Slide 7: 传统的分离式解决方案

在Spark出现之前，我们通常使用分离的解决方案：

- 批处理使用Hadoop MapReduce
- 交互查询使用Apache项目
- 流处理使用Apache Storm

这种分离式的架构存在一些问题。

Slide 8: 无法无缝共享

第一个问题是无法无缝共享。

当我们使用不同的技术栈时，
数据需要在Storm、Apache和Hadoop MapReduce之间进行格式转换，
这增加(**zēngjiā**)了系统的复杂性。

Slide 9 - 유지보수 비용이 높음

두 번째 문제는 **유지보수 비용이 높다**는 것입니다.

각 시스템은 독립적인 팀의 유지보수가 필요합니다:

- Hadoop MapReduce는 전문 팀 필요
- Apache Storm은 전문 팀 필요
- Apache 프로젝트도 전문 팀 필요

이는 인건비와 유지보수 난이도를 크게 증가시킵니다.

Slide 9: 维护成本较高

第二个问题是维护成本较高。

每个系统都需要独立的团队来维护：

- Hadoop MapReduce 需要专门的团队
- Apache Storm需要专门的团队
- Apache项目也需要专门的团队

这大大增加了人力成本和维护难度。

Slide 10 - 자원 활용 불충분

세 번째 문제는 **자원 활용이 불충분**하다는 것입니다.

전통적인 아키텍처에서는 클러스터 자원을 서로 다른 시스템 간에 동적으로 할당할 수 없으며, 클러스터 자원을 통합 관리하고 자원 활용 효율을 높이기 위해 공통 자원 관리자가 필요합니다.

Slide 10: 资源利用不充分

第三个问题是资源利用不充分。

在传统架构中，
集群资源无法在不同系统间动态分配，
需要一个公共资源管家来统一管理集群资源，提高资源利用效率。

Slide 11 - 결함 요약

전통적인 분리형 아키텍처의 세 가지 주요 결함을 요약하면:

1. 원활한 공유 불가
2. 사용 비용이 높음
3. 자원 활용 불충분

이러한 문제들이 Spark 생태계의 탄생을 촉진했습니다.

Slide 12 - Spark의 원스톱 솔루션

Spark는 **"하나의 소프트웨어 스택으로 다양한 응용 시나리오 요구사항을 충족"**한다는 이념을 제시했습니다.

Spark 생태계는 다음을 포함합니다:

- **상위 컴포넌트**: Spark SQL, Spark Streaming, MLlib, GraphX 등
- **통합 컴퓨팅 엔진**: Apache Spark 기반
- **분산 스토리지 시스템**: Tachyon
- **스토리지 지원**: HDFS, S3 등
- **자원 관리자**: Apache Mesos, YARN 등

Slide 11: 缺陷(**quēxiàn**)总结

总结一下传统分离式架构的三个主要缺陷：

1. 无法无缝共享
2. 使用成本较高
3. 资源利用不充分

这些问题促使(**cùshǐ**)了Spark生态系统的诞生。

Slide 12: Spark的一站式解决方案

Spark提出了"一个软件栈满足不同应用场景需求"的理念。

我们可以看到Spark生态系统包含了：

- Spark SQL、Spark Streaming、MLlib、GraphX等上层组件
- 基于Apache Spark的统一计算引擎
- Tachyon分布式存储系统
- HDFS、S3等存储支持
- Apache Mesos和Yarn等资源管理器

Tach-": "타크 "빠르다"는 뜻).
"-yon": "온" (입자 의미의 접미사

Slide 13 - Spark 생태계 아키텍처

Spark 생태계는 계층형 아키텍처를 채택합니다:

- **최하위 계층**: Spark Core - 핵심 컴퓨팅 능력 제공
- **상위 계층**: 네 개의 주요 컴포넌트
  - Spark SQL
  - Spark Streaming
  - MLlib
  - GraphX

이러한 설계는 모든 컴포넌트가 통합된 Spark Core 기반으로 구축되어, 컴포넌트 간의 원활한 통합을 보장합니다.

Slide 14 - 하나의 소프트웨어 스택으로 다양한 응용 시나리오 충족

이것이 바로 Spark의 핵심 가치입니다: **하나의 소프트웨어 스택으로 다양한 응용 시나리오를 충족**

전통적인 분리형 아키텍처와 비교하여, Spark는 통합된 솔루션을 제공하며, 빅데이터 처리의 복잡성을 크게 단순화합니다.

Slide 15:

Slide 13: Spark生态系统架构

Spark生态系统采用分层架构：

- 最底层是Spark Core，提供核心计算能力
- 上层是四个主要组件：
  - ✓ Spark SQL、
  - ✓ Spark Streaming、
  - ✓ MLlib、
  - ✓ GraphX

这种设计使得所有组件都基于统一的Spark Core，确保了组件间的无缝集成。

Slide 14: 一个软件栈满足不同应用场景

这就是Spark的核心价值：
一个软件栈满足不同应用场景。

相比传统的分离式架构，
Spark提供了统一的解决方案，
大大简化了大数据处理的复杂性。

Slide 15: 가볍게 설명

SQL 即席查询 jíxí cháxún
实时流式计算
机器学习
图计算

Slide 16 - 원스톱 빅데이터 솔루션

YARN 자원 관리자를 통해, Spark는 Storm, Hadoop MapReduce 등 다른 빅데이터 기술과 공존할 수 있으며, 진정한 원스톱 빅데이터 솔루션을 형성합니다.

Slide 17 - 생태계 장점

Spark 생태계를 사용한 후, 서로 다른 빅데이터 처리 작업을 전문 팀이 담당할 수 있지만, 모두 통합된 Spark 플랫폼을 사용하여 시스템 유지보수의 복잡성을 크게 낮출 수 있습니다.

Slide 18 - 통합된 Spark 솔루션

최종적으로 모든 빅데이터 처리 요구사항을 통합된 Spark 플랫폼에서 해결할 수 있습니다. 이것이 바로 Spark 생태계의 강력한 점입니다.

Slide 19 - 버클리 데이터 분석 소프트웨어 스택(BDAS)

Spark는 원래 버클리 대학교에서 개발한 BDAS(Berkeley Data Analytics Stack) 프로젝트의 핵심 컴포넌트였습니다. 이 프로젝트의 목표는 완전한 빅데이터 분석 소프트웨어 스택을 만드는 것이었습니다.

Slide 16: 一站式大数据解决方案

通过YARN资源管理器，Spark可以与Storm、Hadoop MapReduce等 其他大数据技术共存，形成真正的一站式大数据解决方案。

Slide 17: 生态系统优势

可以看到，使用Spark生态系统后，不同的大数据处理任务可以由专门的团队负责，但都使用统一的Spark平台，大大降低了系统维护的复杂性。

Slide 18: 统一的Spark解决方案

最终，所有的大数据处理需求都可以在统一的Spark平台上解决，这就是Spark生态系统的强大之处。

Slide 19: 伯克利数据分析软件栈(BDAS)

Spark最初是伯克利大学开发的BDAS（Berkeley Data Analytics Stack） 项目的核心组件。
这个项目的目标是创建一个完整的大数据分析软件栈。

Slide 20 - BDAS 아키텍처 다이어그램

이 그림은 완전한 BDAS 아키텍처를 보여줍니다:

- **진한 파란색 부분**: 이미 릴리스된 컴포넌트
- **연한 파란색 부분**: 개발 중인 컴포넌트
- **회색 부분**: 관련 외부 프로젝트

Spark는 핵심 엔진으로서 다양한 프로그래밍 언어 API를 지원합니다.

Slide 20: BDAS架构图

这张图展示了完整的BDAS架构。
我们可以看到：

- 深蓝色部分是已经发布的组件
- 浅蓝色部分是正在开发的组件
- 灰色部分是相关的外部项目

Spark作为核心引擎，支持多种编程语言API。

Slide 21 - BDAS 컴포넌트 상세 설명

각 컴포넌트의 기능을 자세히 설명하겠습니다:

- **Spark Streaming**: 스트림 컴퓨팅 요구사항 충족
- **BlinkDB**: 대화형 쿼리 분석 요구사항 충족
- **GraphX**: 그래프 컴퓨팅 요구사항 충족
- **MLlib**: 머신러닝 요구사항 충족
- **Spark Core**: 핵심 컴퓨팅 능력 제공
- **Tachyon**: 메모리 기반 분산 파일 시스템
- **HDFS**: 디스크 기반 분산 파일 시스템
- **S3**: 아마존 클라우드 심플 스토리지 서비스
- **Mesos/Hadoop YARN**: 자원 스케줄링 관리자

Slide 22 - MLlib 머신러닝 라이브러리

MLlib는 Spark의 머신러닝 알고리즘 라이브러리로, 분류와 클러스터링 등을 포함한 풍부한 머신러닝 알고리즘을 제공합니다.

MLlib 사용은 매우 간단합니다: **인터페이스 호출 → 파라미터 전달 → 학습 진행 → 결과 출력**

Slide 21: BDAS组件详解

让我详细解释各个组件的功能：

1. Spark Streaming：满足流计算需求
2. BlinkDB：满足交互式查询分析需求
3. GraphX：满足图计算需求
4. MLlib：满足机器学习需求
5. Spark Core：提供核心计算能力
6. Tachyon：基于内存的分布式文件系统
7. HDFS：基于磁盘的分布式文件系统
8. S3：亚马逊云端简单存储服务
9. Mesos/Hadoop YARN：资源调度管理器

Slide 22: MLlib机器学习库

MLlib是Spark的机器学习算法库，提供了丰富的机器学习算法，包括分类和聚类等。

使用MLlib非常简单：调用接口 → 传参数 → 进行训练 → 输出结果。

Slide 23 - Structured Streaming의 도입

이 그림에서 볼 수 있듯이, Spark 생태계는 지속적으로 진화하고 있으며, Structured Streaming 컴포넌트가 새로 추가되어 스트림 처리 능력을 더욱 강화했습니다.

Slide 24 - Spark 2.0의 중요 업데이트

Spark 2.0는 중요한 새 기능을 도입했습니다: **Structured Streaming** 이것은 스트림 처리 기능의 중대한 개선입니다.

Slide 25 - 두 가지 스트림 컴퓨팅 방안

Spark는 두 가지 스트림 컴퓨팅 솔루션을 제공합니다:

- **Spark Streaming**: RDD 데이터 추상화 기반, 특정 응용 시나리오에 적합
- **Structured Streaming**: DataFrame 데이터 추상화 기반, 권장되는 스트림 컴퓨팅 방안

Slide 23: - Structured Streaming的引入

从这张图可以看到，Spark生态系统在不断演进，新增了Structured Streaming组件，进一步增强了流处理能力。

Slide 24: Spark 2.0的重要更新

Spark 2.0引入了一个重要的新特性：Structured Streaming。
这是对流处理功能的重大改进。

Slide 25: - 两种流计算方案

Spark提供了两种流计算解决方案：

1. Spark Streaming：
   基于RDD数据抽象，适用于特定应用场景
2. Structured Streaming：
   基于DataFrame数据抽象，是推荐的流计算方案

这次课程中，我们不学Spark Streaming，而是直接学习Structured Streaming。

Slide 26 - Structured Streaming의 발전 과정

Structured Streaming의 발전 과정:

- **Spark 2.0**: 시험 버전
- **Spark 2.1**: 지속 개선
- **Spark 2.2**: 정식 버전

Slide 27 - Spark 2.3 릴리스

2018년 2월 28일, Spark 2.3이 정식으로 릴리스되었으며, Structured Streaming은 실시간 스트림 컴퓨팅의 중요한 컴포넌트가 되었습니다.

Slide 28 - Spark 2.3의 중요 특징

Spark 2.3은 밀리초 수준의 스트림 컴퓨팅을 지원하며, Spark 생태계의 컴포넌트는 매우 다양하여 다른 컴포넌트들을 잘 대체할 수 있어, Spark 생태계의 강력한 실력을 보여줍니다.

Slide 26: Structured Streaming的发展历程

Structured Streaming的发展历程：

- Spark 2.0：试用版
- Spark 2.1：继续改进
- Spark 2.2：正式版

Slide 27: Spark 2.3的发布

2018年2月28日，Spark 2.3正式发布，Structured Streaming成为实时流计算的重要组件。

Slide 28: Spark 2.3的重要特性

Spark 2.3支持毫秒级的流计算，Spark生态系统当中组件非常多，可以很好地去取代一些其他的组件，展现了Spark生态系统的强大实力。

Slide 29 - 응용 시나리오 비교 표

이 표는 다양한 응용 시나리오, 시간 범위, 다른 프레임워크와 Spark 생태계 컴포넌트의 대응 관계를 요약합니다:

응용 시나리오	시간 범위	다른 프레임워크	Spark 생태계
복잡한 배치 데이터 처리	시간 단위	MapReduce/Hive	Spark Core
대화형 쿼리	분/초 단위	Impala/Dremel/Drill	Spark SQL
실시간 데이터 스트림 처리	밀리초/초 단위	Storm/S4	Spark Streaming/Structured Streaming
데이터 마이닝	-	Mahout	MLlib
그래프 구조 데이터 처리	-	Pregel/Hama	GraphX

Slide 29: 应用场景对比表

这张表格总结了不同应用场景、时间跨度、其他框架与Spark生态系统组件的对应关系：

复杂批量数据处理：小时级，对应MapReduce/Hive，使用Spark Core

交互式查询：分钟级/秒级，对应Impala/Dremel Drill，使用Spark SQL

实时数据流处理：毫秒/秒级，对应Storm/S4，使用Spark Streaming/Structured Streaming

数据挖掘：对应Mahout，使用MLlib

图结构数据处理：对应Pregel/Hama，使用GraphX

Slide 30 - Mahout의 변화

흥미롭게도, Mahout는 이전에 MapReduce 기반의 머신러닝 알고리즘 라이브러리였습니다. 이는 중요한 기술 진화 트렌드를 설명합니다.

Slide 31 - 기술 진화 트렌드

우리는 명확한 기술 진화 트렌드를 볼 수 있습니다:
Hadoop MapReduce는 저물고, Spark가 부상하고 있습니다.
이는 빅데이터 기술의 발전 방향을 반영합니다.

Slide 32 - Mahout 팀의 선택

더 흥미로운 것은, Mahout 팀이 완전히 Spark 쪽으로 기울어, Spark를 하위 컴퓨팅 엔진으로 사용하기 시작했다는 것입니다.

Slide 33 - 기술 지원의 변화

Mahout 팀은 MapReduce의 후속 업데이트를 더 이상 지원하지 않겠다고 선언했습니다. 이는 중요한 기술 전환점입니다.

Slide 30: Mahout的转变

有趣的是，Mahout以前是基于MapReduce的机器学习算法库，
这说明了一个重要的技术演进趋势。

Slide 31: - 技术演进趋势

我们可以看到明显的技术演进趋势：Hadoop MapReduce 日落西山，
而Spark正在崛起(**juéqǐ)**，

这反映了大数据技术的发展方向。

Slide 32: - Mahout团队的选择

更有意思的是，Mahout团队完全倒向(**dǎoxiàng)**了Spark，开始倒戈(**dǎogē,**
전향)使用Spark作为底层计算引擎。

Slide 33:- 技术支持的变化

Mahout团队宣布不再支持MapReduce的后续更新，这是一个重要的技术转折点
(**zhuǎnzhédiǎn)**。

Slide 34 - Mahout의 현재 상황

현재 Mahout는 Spark 기반의 머신러닝 알고리즘 라이브러리가 되었으며, 이는 Spark 생태계의 매력과 기술적 우위를 완벽하게 증명합니다.

Slide 35 - 요약 표

마지막으로, 이 비교 표를 다시 한번 검토해봅시다.
이는 Spark 생태계가 어떻게 전통적인 분리형 빅데이터 처리 아키텍처를 전면적으로 대체했는지를 명확하게 보여줍니다.

연습 문제 및 해설

Slide 36:

문제 1

정답: D

해설: 전통 아키텍처의 문제는 시스템 통합과 운영 복잡성에 있으며, 개별 컴포넌트의 성능 자체가 문제는 아닙니다. 전통 아키텍처의 세 가지 결함은 원활한 공유 불가, 유지보수 비용 높음, 자원 활용 불충분이며, 컴포넌트 성능 문제가 아닙니다.

Slide 34:- Mahout的现状

现在，Mahout已经是基于Spark的机器学习算法库，这完美地证明了Spark生态系统的吸引力和技术优势。

Slide 35:- 总结表格

最后，让我们再次回顾这张对比表格，它清晰地展示了Spark生态系统如何全面替代了传统的分离式大数据处理架构。

Slide 36:

문제 1

정답: D

解析: 传统架构的问题在于系统集成和运维复杂性，而不是单个组件的性能本身。
传统架构的三大缺陷是无法无缝共享、维护成本高、资源利用不充分，并非组件性能问题。

Slide 37:**문제 2**

정답: B

해설: 계층형 아키텍처의 핵심은 통합된 Spark Core를 통해 컴포넌트 간 원활한 통합을 보장하는 것입니다. 이렇게 하면 모든 상위 컴포넌트가 동일한 컴퓨팅 엔진을 기반으로 하여, 전통적인 분리형 아키텍처의 데이터 포맷 변환 및 시스템 통합 문제를 피할 수 있습니다.

Slide 38:

문제 3

정답: B

해설: 실시간 모니터링(Structured Streaming), 배치 분석(Spark Core), 대화형 쿼리(Spark SQL)가 최적의 조합입니다. 시간 범위 요구사항에 따라: 실시간 모니터링(밀리초 단위)은 Structured Streaming이 적합하고, 일일 배치 분석(시간 단위)은 Spark Core가 적합하며, 대화형 쿼리(초 단위)는 Spark SQL이 적합합니다. Structured Streaming은 더 진보된 스트림 처리 방안입니다.

Slide 37:

정답: B

解析: 分层架构的核心价值在于通过统一的Spark Core确保各组件间的无缝集成。这样所有上层组件都基于同一个计算引擎，避免了传统分离式架构中数据格式转换和系统整合的问题。

Slide 38:

정답: B .

解析:

根据时间跨度需求：实时监控(毫秒级)适合用Structured Streaming，每日批量分析(小时级)适合用Spark Core，交互式查询(秒级)适合用Spark SQL。Structured Streaming是更先进的流处理方案。

문제 4

정답: B

해설: 이는 통합 컴퓨팅 엔진이 분리형 아키텍처보다 빅데이터 발전 요구에 더 부합한다는 트렌드를 반영합니다. Mahout 팀의 전환은 빅데이터 기술 발전의 중요한 트렌드를 반영합니다: 통합 컴퓨팅 엔진이 분리형 아키텍처보다 현대 빅데이터 처리 요구사항에 더 부합합니다. 이는 "하나의 소프트웨어 스택으로 다양한 응용 시나리오 충족"이라는 기술 발전 방향을 구현합니다.

문제 5

정답: C

해설: 이 설계 이념은 서로 다른 빅데이터 처리 시나리오 간의 기술적 장벽을 제거하는 것이 목적입니다. 이 설계 이념의 핵심 목적은 서로 다른 빅데이터 처리 시나리오(배치 처리, 스트림 처리, 머신러닝, 그래프 컴퓨팅) 간의 기술적 장벽을 제거하여, 사용자가 통합 플랫폼에서 모든 빅데이터 처리 작업을 완료할 수 있도록 하고, 데이터 포맷 변환 및 시스템 통합의 복잡성을 피하는 것입니다.

Slide 39:

정답: B

解析: Mahout团队的转向反映了大数据技术发展的重要趋势：统一计算引擎比分离式架构更符合现代大数据处理需求。这体现了"一个软件栈满足不同应用场景"的技术发展方向。

Slide 40:

정답: C

解析: 这一设计理念的核心目的是消除不同大数据处理场景(批处理、流处理、机器学习、图计算)之间的技术壁垒，让用户在统一平台上完成所有大数据处理任务，避免数据格式转换和系统集成的复杂性。

문제 6

정답: B

해설: Structured Streaming의 주요 개선점은 DataFrame 추상화를 기반으로 한 더 고급 API 제공입니다. Structured Streaming의 주요 개선점은 DataFrame 데이터 추상화를 기반으로 더 고급이고 사용하기 쉬운 API를 제공한다는 것이며, 전통적인 Spark Streaming은 RDD 추상화를 기반으로 합니다. DataFrame 추상화는 스트림 처리를 더 간단하고 강력하게 만듭니다.

문제 7

정답: C

해설: YARN과 Mesos는 클러스터 리소스의 동적 할당과 효율적 활용을 위한 자원 관리자입니다. YARN과 Mesos는 클러스터 자원 관리자로서, 전통적인 분리형 아키텍처에서 각 시스템이 독립적으로 자원을 점유하고 동적으로 할당할 수 없는 문제를 해결합니다. 이들은 "공공 자원 관리자"로서 자원의 통합 스케줄링과 충분한 활용을 실현합니다.

Slide 41:

정답: B

解析: Structured Streaming的主要改进在于基于DataFrame数据抽象，提供了更高级、更易用的API，而传统Spark Streaming基于RDD抽象。DataFrame抽象使得流处理更加简单和强大。

Slide 42:

정답: C

解析:

YARN和Mesos作为集群资源管理器，解决了传统分离式架构中各系统独立占用资源、无法动态分配的问题。它们作为"公共资源管家"，实现了资源的统一调度和充分利用。

문제 8

정답: B

해설: Tachyon은 고속 데이터 액세스를 제공하여 상위 계산 컴포넌트의 성능 요구를 지원하는 것이 핵심입니다. Tachyon은 메모리 기반 분산 파일 시스템으로서, 상위 계층 컴퓨팅 컴포넌트에 고속 데이터 액세스 능력을 제공하여 Spark 생태계의 고성능 요구사항을 지원하는 것이 전략적 의의입니다. 스토리지 계층과 컴퓨팅 계층을 연결하는 중요한 다리입니다.

문제 9

정답: A

해설: 사용자 행동 그래프(GraphX), 실시간 추천(Spark Streaming), 배치 처리(Spark Core), 쿼리(Spark SQL)가 최적의 할당입니다. 합리적인 작업 할당은: GraphX가 사용자 구매 행동 그래프 분석을 처리하고, Spark Streaming이 실시간 상품 추천을 처리하며, Spark Core가 배치 주문 데이터를 처리하고, Spark SQL이 상품 판매 보고서 쿼리를 처리하는 것입니다. 이렇게 하면 각 컴포넌트의 장점을 충분히 발휘할 수 있습니다.

Slide 43:

답: B

解析: Tachyon作为基于内存的分布式文件系统，其战略意义在于为上层计算组件提供高速数据访问能力，支撑Spark生态系统的高性能需求。它是连接存储层和计算层的重要桥梁

Slide 44:

답: A

解析: 合理的任务分配应该是：GraphX处理用户购买行为图谱分析，Spark Streaming处理实时商品推荐，Spark Core处理批量订单数据，Spark SQL处理商品销售报表查询。这样能充分发挥各组件的优势。

문제 10

정답: D

해설: 스타트업의 제약된 자원 상황에서 통일된 기술 스택으로 유지보수 부담을 줄이고 다양한 요구를 해결할 수 있는 것이 최대 장점입니다. 스타트업 기업에게 가장 중요한 것은 통일된 기술 스택이 가져오는 운영 간소화입니다. 팀은 Spark 기술 한 세트만 마스터하면 배치 처리, 스트림 처리, 머신러닝, 그래프 컴퓨팅 등 다양한 요구사항을 처리할 수 있어, 인건비와 유지보수 부담을 크게 줄일 수 있습니다.

Slide 47 - 강의 종료

경청해 주셔서 감사합니다! 오늘 우리는 Spark 생태계의 설계 이념, 아키텍처 구성 및 기술적 장점을 깊이 이해했습니다. Spark 생태계는 빅데이터 기술의 발전 방향을 대표하며, 여러분이 깊이 학습하고 실습하시기를 바랍니다. 궁금한 점이 있으시면 언제든지 연락 주시기 바랍니다. 감사합니다!

Slide 45:

정답: D

解析: 对于初创公司而言，最重要的是统一技术栈带来的运维简化。团队只需掌握一套Spark技术，就能处理批处理、流处理、机器学习、图计算等多种需求，大大减少了人力成本和维护负担。

Slide 46:

正确答案：B

Slide 47:- 课程结束

谢谢大家的聆听！今天我们深入了解了Spark生态系统的设计理念、架构组成和技术优势。Spark生态系统代表了大数据技术的发展方向，希望大家能够深入学习和实践。

如果有任何问题，欢迎随时联系我。谢谢！