Spark 강의 자료

Slide 1:

Slide 2: 오늘의 학습 내용

먼저 오늘 학습할 내용을 살펴보겠습니다.
우리는 네 가지 측면에서 Spark를 깊이 이해할 것입니다:

- **첫째는 Spark 개요**로, Spark에 대한 전체적인 인식을 갖게 됩니다
- **그 다음은 Spark 생태계**로, Spark가 고립된 도구가 아님을 이해합니다
- **이어서 Spark 실행 아키텍처**입니다 - 이것이 핵심 내용입니다
- **마지막으로 Spark 배포 방식**으로, 실용성이 매우 강합니다.

운전을 배우는 것과 같습니다.
먼저 차의 기본 구조를 이해하고, 그 다음 조작법을 익히며, 마지막으로 도로에서 운전할 수 있게 됩니다!

Slide 1:

Slide 2:

让我们先看看今天的学习内容。
我们将从四个方面来深入了解Spark：

- 首先是Spark概述，让大家对Spark有个整体认识**(rèn shi)**；
- 然后是Spark生态系统，了解Spark不是一个孤立的工具；
- 接着是Spark运行架构，这是重点内容；
- 最后是Spark的部署方式，实用性很强。

就像学习开车一样，
我们先要了解车的基本构造，
然后学会操作，
最后才能上路驾驶(jià shǐ)！

Slide 3: 빅데이터의 시대

이 Slide는 아름다운 워드 클라우드를 보여줍니다. BIG DATA, ANALYTICS, INTERNET 등의 키워드가 포함되어 있습니다.

이러한 단어들이 오늘 우리가 배울 주제를 완벽하게 설명합니다.
빅데이터 시대에 데이터는 공기처럼 어디에나 존재합니다.
상상해보세요. 전 세계의 데이터를 바다에 비유한다면, Spark는 우리가 데이터 바다를 항해하는 슈퍼 쾌속정입니다!

Slide 4: 2.1 Spark 개요

이 아름다운 파란색 기하학적 도형이 보이시나요? 이것은 Spark의 DNA 구조도와 같아서, 복잡하면서도 정교합니다.

이 도형은 Spark의 다면성과 강력한 기능을 상징합니다.

이제 Spark의 세계로 공식적으로 들어가봅시다!
상상해보세요. 우리는 지금 하이테크 건물 입구에 서서 탐험을 준비하고 있습니다. 이 건물이 바로 Spark이며, 각 층마다 다양한 놀라움이 우리를 기다리고 있습니다!

Slide 3:

这张幻灯片展示了一个漂亮的词云图，里面包含了BIG DATA、ANALYTICS、INTERNET等关键词。

这些词汇完美诠释(**quán shì**)了我们今天要学习的主题。
在大数据时代，数据就像空气一样无处不在。
想象一下，如果把全世界的数据比作海洋，那么Spark就是我们遨游(**áo yóu**)数据海洋的超级快艇(**kuài tǐng**)！

Slide 4: 2.1 Spark概述
看到这个美丽的蓝色几何图形了吗？
这就像Spark的DNA结构图，复杂而精致。

这个图形象征着Spark的多面性和强大功能。

现在，让我们正式走进Spark的世界！

想象一下，我们现在站在一座高科技大厦的门口，准备进入探索。
这座大厦就是Spark，每一层都有不同的惊喜等着我们！

Slide 5: 미국 캘리포니아 대학교 버클리 캠퍼스

이 사진은 미국 캘리포니아 대학교 버클리의 아름다운 캠퍼스를 보여줍니다.
저 상징적인 종탑이 보이시나요?
그것이 바로 유명한 새더 타워로, 높이 307피트이며 버클리의 랜드마크 건물입니다.

바로 이런 학문적 분위기가 가득한 곳에서 Spark가 탄생했습니다!
상상해보세요. 이런 환경에서 똑똑한 컴퓨터 과학자들이 카페에 앉아 데이터 처리를 더 빠르고 강력하게 만드는 방법을 논의하고 있습니다.
갑자기 영감의 불꽃이 번쩍입니다 - Spark가 이렇게 탄생했습니다!
이것이 바로 "Spark"(불꽃)라고 불리는 이유입니다!

Slide 5: 美国加州大学伯克利分校

这张照片展示的是美国加州大学伯克利分校的美丽校园。
看到那座标志性的钟楼**(zhōng lóu)**了吗？
那就是著名的萨瑟塔**(Sà sè tǎ, sather tower)**，高307英尺，是伯克利的地标建筑。

就是在这个充满学术氛围的地方，Spark诞生了！
想象一下，在这样的环境中，聪明的计算机科学家们坐在咖啡厅里讨论着如何让数据处理变得更快、更强大。

突然，灵感的火花闪现**(shǎn xiàn)**——Spark就这样诞生了！
这就是为什么它叫"Spark"（火花）的原因！

Slide 6: AMP 연구소의 구성

이 그림은 매우 흥미롭습니다!
만화 스타일의 일러스트를 보세요.

연구자가 세 개의 버블에 둘러싸여 있으며, 각각 Algorithm(알고리즘), Machine(기계), People(사람)이라고 쓰여 있습니다.

이것은 마법의 삼각형과 같습니다:

- 알고리즘은 마법 주문
- 기계는 마법 도구
- 사람은 마법사

오직 이 세 가지가 완벽하게 결합될 때만 Spark와 같은 "데이터 마법"을 창조할 수 있습니다!

AMP 연구소는 바로 이런 신비로운 곳으로, 가장 똑똑한 두뇌, 가장 선진적인 장비, 그리고 가장 창의적인 알고리즘 사고가 모여 있습니다.

Slide 6: AMP实验室的构成

这张图非常有趣！
看这个卡通(cartoon)风格的插画((chā huà)：
一个研究者被三个气泡包围，
分别写着Algorithm（算法）、Machine（机器）、People（人）。

这就像一个魔法三角(**mó fǎ sān jiao)**：

1. 算法是魔法咒语(**zhòu yǔ)**，
2. 机器是魔法道具，
3. 人是魔法师。

只有三者完美结合，才能创造出Spark这样的"数据魔法"！

AMP实验室就是这样一个神奇的地方，汇聚了最聪明的头脑、最先进的设备，还有最富(**zuì fù)**创造力的算法思维。

Slide 7: 2009년 - Spark의 탄생

2009년! 금색 글씨로 기념할 가치가 있는 해입니다!
이 해에 많은 중요한 일들이 일어났습니다:

- iPhone 3GS 출시, 스마트폰 시대가 진정으로 도래
- 《아바타》 영화 개봉, 3D 기술이 세계를 놀라게 함
- 그리고 우리의 Spark가 버클리 대학에서 조용히 탄생했습니다!

당시 아무도 실험실에서 조용히 진행되던 이 작은 프로젝트가 십여 년 후 세계를 바꾸는 슈퍼스타가 될 것이라고는 생각하지 못했습니다!

씨앗과 같아서, 비록 작지만 엄청난 에너지를 담고 있었습니다.

Slide 7: 2009年 - Spark诞生

2009年！这是一个值得用金色字体纪念的年份！
这一年发生了很多重要事情：

1. iPhone 3GS 发布，
   智能手机时代真正到来
2. 《阿凡达》电影上映，3D技术震撼(**zhèn hàn)**世界
3. 还有，我们的Spark在伯克利大学悄悄(**qiāo qiāo)**诞生了！

当时谁也没想到，这个在实验室里默默无闻的小项目，十几年后会成为改变世界的超级明星！

就像一颗种子(**yī kē zhǒng zi)**，虽然小，但蕴含(yùn hán)着巨大的能量。

Slide 8: Spark의 탄생

이 초귀여운 병아리가 껍질을 깨고 나오는 장면을 보세요!
병아리 주위에는 따뜻한 오렌지색 빛이 있어 새 생명의 활력을 상징합니다.
이것은 Spark의 탄생 과정을 완벽하게 비유합니다.
병아리가 껍질 속에서 21일 동안 천천히 발육하는 것처럼, Spark도 버클리 대학이라는 "인큐베이터"에서 정성스럽게 육성되고 발전했습니다.
2009년, 모든 조건이 성숙했을 때, Spark는 마침내 "껍질을 깨고 나와" 전체 빅데이터 세계를 바꿀 준비를 했습니다!
이 순간은《해리 포터》의 마법 학교 기숙사 배정 의식처럼 흥분됩니다!

Slide 8: Spark破壳(**pò ké**)而出

看这个超可爱的小鸡破壳而出的画面！
小鸡周围还有温暖的橙色光芒(**chéng sè guāng máng**)，象征着新生命的活力。

这完美地比喻了Spark的诞生过程。

就像小鸡需要在蛋壳里慢慢发育21天，
Spark也在伯克利大学这个"孵化器(**fū huà qì**)"里经过了精心的孕育(**yùn yù**)和发展。

2009年，当所有条件都成熟时，Spark终于"破壳而出"，准备改变整个大数据世界！

这个时刻就像《哈利·波特》中魔法学院的分院仪式一样激动(**jī dòng**)人心！

Slide 9: **Spark는 Hadoop의 장점을 계승하고 단점을 개선**

이 그림은 기술 진화의 지혜를 보여줍니다!
Hadoop MapReduce에서 Spark로, 두 개의 중요한 화살표가 있습니다:
검은색 화살표는 "장점 계승"을, 노란색 구부러진 화살표는 "단점 개선"을 나타냅니다.

이것은 중국 고대의 이야기를 떠올리게 합니다: 거인의 어깨 위에 서기.

Spark 창조자들은 매우 똑똑했습니다. 그들은 처음부터 시작하지 않고, "Hadoop 형, 당신의 장점은 배우고, 당신의 단점은 개선하겠습니다!"라고 말했습니다.

이것이 기술 진보의 올바른 자세입니다 - 과거를 계승하고 미래를 여는 것입니다!

Slide 9: Spark继承Hadoop优点，改进缺陷

这张图展示了技术进化的智慧！

从Hadoop MapReduce到Spark，有两个重要的箭头：
一个黑色箭头代表"优点继承(**jì chéng**)"，
-一个黄色弯曲箭头代表"缺陷(**quē xiàn**)改进"。

这让我想起了一个中国古代的故事：站在巨人的肩膀(**jiān bang**)上。

Spark的创造者们非常聪明，他们没有从零开始，而是说："Hadoop大哥，你的优点我们学习，你的缺点我们来改进！

"这就是技术进步的正确姿势(**zī shì**)——继往开来(**(jì wǎng kāi lái)** ，推陈出新((**tuī chén chū xīn)** ！

温故知新
法古创新
继往开来
推陈出新

Slide 10: Spark - 메모리 기반 컴퓨팅 프레임워크

이 그림은 Spark의 핵심 포지셔닝을
명확하게 보여줍니다.
화살표가 가리키는 키워드들을 보세요:
구축 → 대규모의, 저지연 → 데이터 분석
애플리케이션.

생생한 비유를 들어보겠습니다:
전통적인 데이터 처리는 주판으로
계산하는 것과 같아서,
각 단계마다 "딱딱" 소리를 내며 구슬을
옮겨야 합니다. 반면 Spark의 메모리
컴퓨팅은 슈퍼컴퓨터를 사용하는 것과
같아서, 두뇌 속에서 "휙휙" 번개처럼
계산합니다!

이것이 "메모리 기반 컴퓨팅
프레임워크"의 매력입니다 - 시간이 멈춘
것처럼 느껴질 정도로 빠릅니다!

Slide 10: Spark - 基于内存的计算框架

这张图清晰地展示了Spark的核心定位。

看那些箭头指向的关键词：
构建→大型的、
构建→低延迟
数据分析应用程序。

让我用一个生动的比喻(**bǐ yù**)：
传统的数据处理就像用算盘计算，
每一步都要"啪啪啪(**pā pā pā**)"地拨珠子
(**bō zhū zi**)；
而Spark的内存计算就像用超级计算机，
在大脑里"嗖嗖嗖(**sōu sōu sōu**)"地闪电
(**shǎn diàn**)计算！

这就是"基于内存的计算框架"的魅力——快
到让你怀疑(**huái yí**)时间停止了！

Slide 11: 2013년 - Apache 가입

2013년, Spark는 그 "인생"에서 중요한 이정표를 맞이했습니다 - Apache 인큐베이터 프로젝트에 가입했습니다!

이것은 재능있는 학생이 세계 최고의 대학에 합격한 것과 같습니다!

Apache 재단은 오픈소스 세계에서 "하버드 대학"처럼 명성이 높습니다. Apache 인큐베이터에 들어갈 수 있다는 것은 Spark가 이미 엄청난 잠재력을 보여주었고, 업계 전문가들의 인정을 받았다는 것을 의미합니다.

이것은 Spark가 "실험실의 총아"에서 "업계의 신예"로 전환하는 핵심적인 단계입니다!

Slide 11: 2013年 - 加入Apache

2013年，Spark迎来了它"人生"中的重要里程碑——加入Apache孵化器项目！

这就像一个有天赋**(tiān fù)**的学生考上了世界顶级大学！

Apache基金会在开源世界就像"哈佛大学"一样声名显赫**(xiǎn hè)**。
能够进入Apache孵化器，说明Spark已经展现出了巨大的潜力，得到了业界专家们的认可。

这是Spark从"实验室宠儿 (chǒng ér) "向"业界新星"转变的关键一步！

Slide 12: Apache 인큐베이터 프로젝트

이 그림은 매우 형상적입니다!
파이 차트가 보이시나요?
Apache 재단은 거대한 "보호 우산"과 같고, Spark는 작은 "노란색 조각"으로 이 대가족에 융합되었습니다.

Apache 재단은 따뜻한 대가족과 같아서, 우수한 오픈소스 프로젝트에 법적 보호, 기술 지도 및 커뮤니티 지원을 제공합니다.

Spark가 이 가족에 가입할 수 있었다는 것은, 재능있는 젊은이가 권위있는 기관의 인증과 양성을 받은 것과 같습니다!

Slide 12: Apache孵化器项目

这张图很形象！
看到那个饼状图(**bǐng zhuàng tú)**了吗？
Apache基金会就像一个巨大的"保护伞(**sǎn)**"，Spark作为一小块"黄色楔形(**xiē xíng)**"融入了这个大家庭。

Apache基金会就像一个温暖的大家庭，为优秀的开源项目提供法律保护、技术指导和社区支持。

Spark能够加入这个家庭，就像一个有才华的年轻人得到了权威机构的认证和培养！

Slide 13: 2018년 - 최상위 프로젝트

2018년! 축하할 만한 또 다른 해입니다!
Spark가 Apache 인큐베이터 프로젝트에서 "졸업"하여 최상위 프로젝트가 되었습니다!

이것은 대학 학부를 졸업하여 학사 학위증을 받은 것과 같습니다!
이제부터 Spark는 더 이상 "인턴"이 아니라 공식적인 "선임 엔지니어"가 되었습니다.
Spark는 이미 독립적으로 일할 수 있는 능력을 갖추었고, 더 중요한 책임을 맡을 수 있게 되었습니다!

이 "졸업식"은 Spark가 공식적으로 성인이 되었으며, 오픈소스 세계에서 자신의 지위와 명성을 갖게 되었음을 나타냅니다!

Slide 13: 2018年 - 顶级项目

2018年！又一个值得庆祝(**qìngzhù)**的年份！
Spark从Apache孵化器项目"毕业"，成为顶级(**dǐngjí)**项目！

这就像从大学本科毕业，拿到了学士学位证书！
从此，Spark不再是"实习生"，而是正式的"高级工程师"了。

它已经具备了独当一面(**dūdāngyīmiàn**)的能力，可以承担(**chéngdān**)更重要的责任！

这个"毕业典礼"标志着Spark正式成年，在开源世界拥有了自己的地位和声誉！

Slide 14: 분산 컴퓨팅 시스템 오픈소스 프로젝트

이 그림은 흥미로운 개념을 보여줍니다:
Spark가 "분산 컴퓨팅 시스템 오픈소스 프로젝트"라는 슈퍼 연합에 가입했습니다!
옆에는 Hadoop과 Apache Storm의 로고도 있습니다.
이것은《어벤져스》와 같습니다! 각 슈퍼히어로는 자신만의 특별한 능력을 가지고 있습니다:

- Hadoop은 "아이언맨"과 같아서, 방어력이 강하고 저장 능력이 일류입니다
- Storm은 "플래시"와 같아서, 실시간 처리 속도가 매우 빠릅니다
- Spark는 "슈퍼맨"과 같아서, 종합 능력이 가장 강하고, 빠르면서도 강력합니다!

그들은 함께 "데이터 세계"를 보호하는 슈퍼 팀을 구성합니다!

Slide 14: 分布式计算系统开源项目

这张图展示了一个有趣的概念：
Spark加入了"分布式计算系统开源项目"这个超级联盟(**lián méng)**！
旁边还有Hadoop和Apache Storm的logo。

这就像《复仇者联盟》！每个超级英雄都有自己的特殊能力：

1. Hadoop像"钢铁侠(Gāngtiěxiá)"，防御力(fángyù lì)强，存储能力一流
2. Storm像"闪电侠"，实时处理速度极快
3. 而Spark像"超人"，综合能力最强，既快又强！

他们一起组成了保护"数据世界"的超级团队！

Slide 15: 2014년 - 기록 경신

2014년, Spark는 업계를 놀라게 한 일을 해냈습니다 - Hadoop이 보유하고 있던 정렬 벤치마크 기록을 경신했습니다!
이것은 올림픽에서 새로운 선수가 경기에 참가했을 뿐만 아니라, 오랫동안 유지되어 온 세계 기록을 깼다는 것과 같습니다!

스포츠 해설자의 흥분된 목소리를 상상해보세요: "너무나 믿기지 않습니다! 버클리에서 온 이 신예가 우리의 속도에 대한 인식을 새롭게 했습니다!"
이 순간, 전 세계의 빅데이터 엔지니어들이 환호했습니다!

Slide 15: 2014年 - 打破纪录

2014年，Spark做了一件震惊业界的事情——打破了Hadoop保持的排序基准纪录！

这就像在奥运会上，一个新选手不仅参加了比赛，还打破了保持多年的世界纪录！

想象一下体育解说员激动(**jī dòng**)的声音："太不可思议了！
这个来自伯克利的新星刷新了我们对速度的认知！"
这一刻，全世界的大数据工程师们都沸腾(**fèi téng**)了！

Slide 16: 벤치마크 정렬 기록

그림을 보세요!
금색 트로피가 빛나고 있으며, Spark가 Hadoop을 이기고 "벤치마크 정렬 기록" 챔피언 타이틀을 획득했습니다!

이 성과는 류샹이 2004년 아테네 올림픽에서 12.91초의 성적으로 110m 허들 금메달을 딴 것만큼 감동적입니다!
이때부터 사람들은 알게 되었습니다:
데이터 처리 경주에서 Spark라는 "육상 선수"가 있다는 것을!

Slide 16: 基准排序纪录

看这张图！
金色的奖杯闪闪发光，Spark战胜(zhànshèng)了Hadoop，赢得了"基准排序纪录"的冠军(guànjūn)称号(chēnghào)！

这个成就就像刘翔在2004年雅典奥运会上以12.91秒的成绩夺得110米跨栏金牌一样激动人心！
从此，人们知道了：在数据处理的赛道上，有一个叫Spark的"飞人"！

Slide 17: 성능 비교 - 컴퓨팅 자원 1/10

여기서 구체적인 비교 데이터를 볼 수 있는데, 이 숫자 뒤에는 놀라운 이야기가 숨어 있습니다:

- **Spark**: 206개 노드, 23분, 100TB 데이터
- **Hadoop**: 2000개 노드, 72분, 100TB 데이터

계산해봅시다:
Spark는 Hadoop의 1/10 기계만 사용했지만, Hadoop의 1/3 시간만 사용했습니다!

이 비교는 아마도 Spark (특수부대) 가 Hadoop (전통 군대) 에 비해 성능에서 얼마나 도약했는지를 보여주는 유명한 사례일 것입니다. 이것은 효율성을 높이고 비용을 절약하기 위해 올바른 기술 프레임워크를 선택하는 것이 얼마나 중요한지를 우리에게 가르쳐 줍니다.

Slide 17: 性能对比 - 计算资源1/10

这里我们看到了具体的对比数据，这些数字背后(bèi hòu)藏着(cáng)惊人(jīng)的故事：

1. Spark：206个节点，23分钟，100TB数据
2. Hadoop：2000个节点，72分钟，100TB数据

让我们算一下：
Spark用了Hadoop 1/10的机器，却只用了Hadoop 1/3的时间！

这个对比很可能是在展示 Spark（特种部队） 相比 Hadoop（传统军队） 在性能上的巨大飞跃。它告诉我们，选择正确的技术框架，对于提升效率、节约成本至关重要。

Slide 18: 성능 비교 - 3배 속도

"3배 속도"라는 네 글자가 특히 눈에 띕니다! 이 숫자 뒤의 의미는 매우 중대합니다.
상상해보세요: 매일 데이터 분석 작업을 완료하는 데 3시간이 필요하다면, Spark를 사용하면 단 1시간만 필요합니다!
남은 2시간은 커피를 마시거나, 인생을 사색하거나, 더 많은 데이터를 처리할 수 있습니다!

이것이 기술 진보가 가져온 혜택입니다 - 우리에게 기다리는 대신 창조할 수 있는 더 많은 시간을 줍니다!

Slide 18: 性能对比 - 3倍速度

"3倍速度"这四个大字特别醒目(**xǐng mù)**！这个数字背后的意义非常重大。

想象一下：如果你每天需要3小时完成数据分析工作，
使用Spark后只需要1小时！
多出来的2小时你可以喝咖啡、思考人生，或者处理更多的数据！

这就是技术进步带来的红利——让我们有更多时间去创造，而不是等待！

Slide 19: 빠른 실행 속도의 이유

이 그림은 Spark 속도가 빠른 비밀 무기를 밝힙니다: 메모리 컴퓨팅 + 순환 데이터 흐름.

요리로 비유해보겠습니다:

- **전통적 방법**: 매번 요리할 때마다 냉장고에서 재료를 꺼내고, 사용 후 다시 넣어야 하므로 효율이 매우 낮습니다
- **Spark 방법**: 자주 사용하는 재료를 모두 가스레인지 옆에 두고, 필요할 때마다 바로 사용하며, 순환적으로 재활용할 수 있습니다!

메모리는 작업대와 같고, 순환 데이터 흐름은 조립 라인 작업과 같습니다.
두 가지가 결합하면, 효율이 자연스럽게 급상승합니다!

Slide 19: 运行速度快的原因

这张图揭示了Spark速度快的秘密武器：
内存计算 + 循环数据流。

让我用做菜来比喻(**bǐ yù**)：

1. 传统方法：每做一道菜都要从冰箱拿食材，用完放回去，效率很低
2. Spark方法：把常用食材都放在灶台边(**zào tái biān**)，随用随取，还能循环利用！

内存就像你的工作台，循环数据流就像流水线作业。
两者结合，效率自然飞起来！

Slide 20: DAG 유향 비순환 그래프 설계 메커니즘

이제 Spark의 핵심 기술인 DAG(유향 비순환 그래프)에 접근했습니다!
오른쪽 그림은 DAG 구조를 보여줍니다: A→B→C→E, B→D→C, F→G, 그리고 독립적인 X.
이것은 무엇처럼 보이나요?
지하철 노선도와 같습니다!
DAG는 Spark의 "스마트 내비게이션 시스템"입니다. 최적의 경로를 계획하여 데이터가 가장 효율적인 경로로 흐르도록 보장하며, 절대로 되돌아가지 않습니다!
바이두 지도보다 더 똑똑합니다!

Slide 21: 데이터셋에 대한 모든 작업

이 그림은 중요한 개념을 보여줍니다: 데이터셋에 대한 모든 작업이 유향 비순환 그래프를 형성합니다.

왼쪽은 데이터베이스 작업의 개략도이고, 오른쪽은 복잡한 DAG 구조도입니다.
이것은 다음과 같습니다:

- 왼쪽은 재료 준비
- 오른쪽은 완전한 레시피 흐름도

Spark는 데이터 작업의 모든 단계를 기록한 다음, 실행 순서를 지능적으로 배치하여 효율을 최대화합니다!

Slide 20: DAG有向无环图设计机制

现在我们接触到了Spark的核心技术：DAG（有向无环图）！
右边的图展示了DAG的结构：
A→B→C→E，B→D→C，F→G，还有一个独立的X。
这看起来像什么？
像一张地铁线路图！
DAG就是Spark的"智能导航系统"，它会规划出最优路线，确保数据按照最高效的路径流动，而且绝对不会走回头路！
比百度地图还聪明！

Slide 21: 所有针对数据集的操作

这张图展示了一个重要概念：所有针对数据集的操作都会形成有向无环图。

左边是数据库操作的示意图，右边是复杂的DAG结构图。
这就像：

1. 左边是食材准备
2. 右边是完整的菜谱(**cài pǔ**)流程图

Spark把你的每一步数据操作都记录下来，
然后智能地安排执行顺序，
确保效率最高！

Slide 22: 유향 비순환 그래프 구조

이 더 큰 DAG 그림은 복잡한 데이터 처리 흐름을 보여줍니다.
빨간색 상자들을 보세요, 다양한 계층을 표시하고 있습니다: External clients, Internal clients, Middle Tier, Data Tier.
이것은 대형 쇼핑몰의 층별 분포도와 같습니다:

- 최상층: 외부 고객
- 중간층: 내부 처리
- 하위층: 데이터 저장

각 계층은 특정 기능을 가지고 있으며, 조화롭게 협력하여 질서정연합니다!

Slide 23: 실행 엔진 기반 스케줄링

"실행 엔진 기반 스케줄링" - 이 몇 글자가 Spark의 지혜를 말해줍니다!

Spark는 초똑똑한 프로젝트 매니저와 같아서, 다음과 같은 일을 합니다:

1. 전체 프로젝트 분석 (DAG 그래프)
2. 최적의 실행 방안 찾기
3. 다른 작업자들에게 작업 할당
4. 각 부서의 협력 조정

이것은 전통적인 "한 단계씩 천천히 진행하는" 방법보다 훨씬 똑똑합니다!

Slide 22: 有向无环图结构

这张更大的DAG图展示了复杂的数据处理流程。
看那些红色框框，标出了不同的层级：External clients、Internal clients、Middle Tier、Data Tier。

这就像一个大型购物中心的楼层分布图：

1. 顶层：外部客户
2. 中层：内部处理
3. 底层：数据存储

每个层级(céng jí)都有特定的功能，协调(xié tiáo)配合，井然有序(jǐng rán yǒu xù)！

Slide 23: 基于执行引擎调度

"基于执行引擎调度"——这几个字道出了Spark的智慧所在！

Spark就像一个超级聪明的项目经理，它会：

1. 分析整个项目（DAG图）
2. 找出最优执行方案
3. 分配任务给不同的工人
4. 协调各部门的配合(pèi hé)

这比传统的"一步一步慢慢来"的方法聪明太多了！

Slide 24: 작업 분할

이 그림은 Spark가 복잡한 작업을 어떻게 분해하는지 명확하게 보여줍니다:
DAG → 분할 → 단계1, 단계2, 단계3 → 재분할 → 구체적 작업

이것은 대형 콘서트 조직과 같습니다:

- 총괄 기획 (DAG)
- 무대, 음향, 보안 등 단계로 분할
- 각 단계를 다시 다른 작업자들에게 구체적인 작업 할당

정연하고 효율이 극히 높습니다!

Slide 25: DAG 방식 채택

이 그림은 작업이 다른 장치에서 분산 실행되는 것을 보여줍니다. 그 연결선들을 보면, 정밀한 협력 네트워크와 같습니다!

DAG 방식을 채택하는 것은 다음과 같습니다:

- **전통적 방법**: 한 사람이 처음부터 끝까지 모든 일을 합니다
- **Spark 방법**: 여러 사람이 협동 작업하며, 각자 가장 잘하는 부분을 합니다

결과는 당연히 Spark가 더 빠르고, 더 안정적이며, 더 강력합니다!

Slide 24: 任务拆分

这张图清晰地展示了Spark如何将复杂任务分解：
DAG → 拆分 → 阶段1、阶段2、阶段3 → 再拆分 → 具体任务

这就像组织一场大型演唱会：

- 总策划（DAG）
- 分成舞台、音响、安保等阶段
- 每个阶段再分配具体任务给不同的工作人员

井井有条，效率极高！

Slide 25: 采用DAG的方式

这张图展示了任务在不同设备上的分布执行。
看那些连线，就像一张精密的协作网络！

采用DAG的方式就像：

1. 传统方法：一个人从头到尾做完所有事情
2. Spark方法：一群人协同作业，每个人做自己最擅长(shànchán)的部分

结果当然是Spark更快、更稳、更强！

Slide 26: 파이프라인 최적화

이 정교한 산업 파이프라인 아이콘을 보세요! 파란색 로봇이 파이프라인에서 작업하고 있으며, 이것은 Spark의 파이프라인 최적화 이념을 완벽하게 해석합니다.

자동차 제조 공장을 생각해보세요:

- 한 작업자가 타이어를 설치
- 다른 작업자가 좌석을 설치
- 세 번째 작업자가 핸들을 설치

각 단계가 병렬로 진행되어 효율을 최대화합니다! Spark의 데이터 처리도 같은 원리입니다 - 파이프라인 최적화로 모든 것이 질서정연해집니다!

Slide 26: 流水线优化

看这个精美(**jīng měi)**的工业流水线图标！
蓝色的机器人正在流水线上工作，
这完美诠释(**quán shì)**了Spark的流水线优化理念。

想想汽车制造厂：

1. 一个工人安装轮胎(lúntāi)
2. 另一个工人安装座椅(zuòyǐ)
3. 第三个工人安装方向盘
   (fāngxiàngpán)

每个环节并行进行，效率最大化！
Spark处理数据也是这个道理——
流水线优化让一切都变得井然有序！

Slide 27:
(노트 없음)

Slide 27: 클릭하면서 애니메이션 따라감
(노트 없음)

Slide 28: 사용하기 쉬움 - 다양한 언어 지원

이 그림은 Spark가 지원하는 프로그래밍 언어들을 보여줍니다: Scala, Java, Python, R 언어.

"언어 천재"처럼, 네 가지 다른 "프로그래밍 방언"을 구사합니다!

이것은 큰 문제를 해결합니다:

- Java 프로그래머는 Java로 Spark를 작성할 수 있습니다
- Python 데이터 과학자는 Python을 사용할 수 있습니다
- R 언어 통계학자는 R 언어를 사용할 수 있습니다
- Scala 애호가는 Scala를 사용할 수 있습니다

모두가 자신에게 가장 익숙한 언어를 사용할 수 있습니다, 이것이 Spark의 "포용성"입니다!

Slide 28: 容易使用 - 多种语言支持

这张图展示了Spark支持的编程语言：Scala、Java、Python、R语言。

就像一个"语言天才"，会说四种不同的"编程方言"！

这就解决了一个大问题：

1. Java程序员可以用Java写Spark
2. Python数据科学家可以用Python
3. R语言统计学家可以用R语言
4. Scala爱好者可以用Scala

每个人都能用自己最熟悉的语言，
这就是Spark的"包容性"！

Slide 29: 효율성 높음

이 아이콘을 보세요: 빛나는 전구, 주위에 파란색 광환이 있어 "고효율" 개념을 상징합니다!

Spark는 금색 왕관을 쓰고 있어 "효율의 왕"임을 나타냅니다!
이 왕관은 그냥 쓴 것이 아니라, 실력과 데이터로 증명하여 얻은 것입니다!

Slide 30: Java 프로그래밍의 번거로움

이 그림은 현실적인 문제를 지적합니다:

Spark 프로그램을 작성할 때, Java 언어는 확실히 비교적 번거롭고, 문법이 복잡하며, 코드가 장황합니다.

붓으로 글씨 쓰기와 타자기로 타이핑하는 차이와 같습니다:

- 붓글씨는 아름답지만 쓰기가 느립니다
- 타자기는 우아하지 않지만 효율이 높습니다

Java는 붓과 같아서, 기능은 강력하지만 확실히 좀 "무겁습니다"!

Slide 29: 效率高

看这个图标：一个发光的灯泡(**dēng pào)**，周围有蓝色的光圈(**quān)**，象征着"高效"的概念！

Spark戴上(**dài shang)**了一顶金色的皇冠，说明它是"效率之王"！
这个皇冠可不是白戴的，是用实力和数据证明赢**(yíng)**得的！

Slide 30: Java编程的繁琐性

这张图指出了一个现实问题：

编写Spark程序时，Java语言确实比较繁琐(fánsuǒ)，语法琐碎(suǒsuì)，代码冗长(rǒngcháng)。

就像用毛笔写字和用打字机打字的区别：

1. 毛笔字虽然漂亮，但写起来慢
2. 打字机虽然不够优雅，但效率高

Java就像毛笔，功能强大但确实有点"重"！

Slide 31: Spark Shell 대화형 프로그래밍

좋은 소식이 왔습니다! Spark는 Shell 대화형 프로그래밍 기능을 제공합니다! 이것은 "즉시 번역기"를 가진 것과 같습니다:

- 한 마디 하면, 즉시 결과를 줍니다
- 완전한 프로그램을 작성할 필요 없이, 생각나는 대로 시도할 수 있습니다
- Spark와 직접 대화하는 것처럼 편리합니다!

학습과 디버깅에 너무나 친절합니다!

Slide 32: Python의 특징

Python은 2025년 언어 순위 상위 1위! 이 그림은 Python의 특징을 보여줍니다:

장점:

- ✅ 데이터 마이닝 능력 강함
- ✅ 문법이 간단하고 입문이 쉬움

단점:

- ❌ 동시성이 좋지 않음
- ❌ 소규모 데이터 분석에 사용됨
- ❌ 기업급 대규모 데이터 분석 프로그램에 적합하지 않음

똑똑하지만 체력이 일반적인 학생과 같습니다: 머리는 좋지만, 육체적 노동은 많이 할 수 없습니다!

Slide 31: Spark Shell交互式编程

好消息来了！Spark提供了Shell交互式编程功能！

这就像有了一个"即时翻译器"：

- 你说一句，它立刻给你结果
- 不用写完整的程序，想到什么试什么
- 就像和Spark直接对话一样方便！

这对学习和调试来说太友好了！

Slide 32: Python의 특징

Python在2025年语言排榜位居第一！这张图展示了

Python的特点：

- ✅ 数据挖掘能力强
- ✅ 语法简单，易入门

- ❌ 并发性不好
- ❌ 用于小规模数据分析
- ❌ 不适合企业级大规模数据分析程序

就像一个聪明但体力一般的学生：脑子好使，但体力活干不了太多！

Slide 33: Hadoop 플랫폼

이 그림은 Scala에서 Python으로, 다시 Hadoop 플랫폼으로 가는 기술 경로를 보여줍니다.
Scala는 Hadoop 플랫폼에서 자신의 "홈 그라운드"를 찾은 것과 같습니다!
그들의 조합은 다음과 같습니다:

- Scala는 우수한 운동선수
- Hadoop은 전문 경기장

둘이 결합하면 최고의 효과를 발휘합니다!

Slide 34: Spark와 Hadoop의 관계

이 그림은 분업을 명확하게 보여줍니다:

- **Spark**: 계산 문제 해결
- **Hadoop**: 저장 문제 해결

황금 파트너와 같습니다:

- 한 명은 "생각"을 책임짐 (Spark 계산)
- 한 명은 "기억"을 책임짐 (Hadoop 저장)

완벽한 상호 보완, 천생연분입니다!

Slide 33:: Hadoop 플랫폼

这张图展示了Scala到Python，
再到Hadoop平台的技术路线。

Scala在Hadoop平台上就像找到了自己的"主场"！
它们的配合就像：

- Scala是优秀的运动员
- Hadoop是专业的体育场

两者结合，发挥出最佳效果！

Slide 34: Spark와 Hadoop의 관계

这张图清楚地展示了分工：

1. Spark：解决计算问题
2. Hadoop：解决存储问题

就像一对黄金搭档**(dā dàng)**：

1. 一个负责"思考"（Spark计算）
2. 一个负责"记忆"（Hadoop存储）

完美互补，天作之合！

Slide 35: 접근 효율 문제

이 그림은 문제를 지적합니다: Python 등 제3자 도구가 Hadoop에 접근하는 효율이 낮습니다.
다음과 같습니다:

- Python이 데이터를 원하지만 많은 "환승역"을 거쳐야 합니다
- 기차를 타고 자주 환승해야 하는 것과 같아서, 효율이 자연히 낮습니다

반면 Spark는 직행 고속철과 같아서, 한 번에 도착합니다!

Slide 36: Spark 애플리케이션 개발

이 그림은 해결책을 보여줍니다:
Scala와 Python으로 직접 Spark 애플리케이션을 개발하세요!

이것은 다음과 같습니다:

- **이전**: Python → 제3자 도구 → Hadoop (먼 길)
- **현재**: Python → Spark (직행 급행)

경로가 더 짧고 효율이 더 높습니다!

Slide 35: 访问效率问题

这张图指出了一个问题：Python等第三方工具访问Hadoop效率低。

就像：

1. Python想要数据，但要通过很多"中转站"
2. 就像坐火车需要频繁(**pín fán**)换乘，效率自然低下

而Spark就像直达高铁，一站到底！

Slide 36: Spark应用程序开发

这张图展示了解决方案：用Scala和Python直接开发Spark应用程序！

这就像：

1. 以前：Python → 第三方工具 → Hadoop（绕 **rào** 远路）
2. 现在：Python → Spark（直达快车）

径更短，效率更高！

Slide 37: 범용성 - 완전한 솔루션

Spark의 범용성은 "완전한 솔루션"과 "기술 소프트웨어 스택"에 반영됩니다.

전능형 선수와 같습니다:

- 단일 종목 챔피언이 아닙니다
- 10종 경기 챔피언입니다
- 하나의 도구로 여러 문제를 해결합니다!

Slide 38: Spark는 단일 소프트웨어가 아님

중요한 관점: Spark는 단일 소프트웨어가 아닙니다!

"스마트폰은 단순한 전화기가 아니다"라고 말하는 것과 같습니다:

- 통신 도구입니다
- 카메라이기도 합니다
- 게임기이기도 합니다
- 학습 플랫폼이기도 합니다

Spark도 이처럼 다면적입니다!

Slide 37: 通用性 - 完整的解决方案

Spark的通用性体现在它是一个"完整的解决方案"和"技术软件栈(**zhàn**)"。

就像一个全能型选手：

1. 不只是单项冠军
2. 而是十项全能冠军
3. 一个工具解决多种问题！

Slide 38:Spark不是一个单一的软件

重要观点：Spark不是一个单一的软件！

这就像说"智能手机不只是电话"一样：

1. 它是通信工具
2. 也是照相机
3. 还是游戏机
4. 更是学习平台

Spark也是如此多面！

Slide 39: 여러 구성 요소

파란색 톱니바퀴 아이콘들을 보세요! Spark는 여러 구성 요소로 이루어져 있으며, 정밀한 기계와 같아서, 각 톱니바퀴가 중요한 역할을 합니다! 톱니바퀴들이 서로 완벽하게 맞물려야만 전체 기계가 효율적으로 작동할 수 있습니다.

Slide 40: 생태계

Spark 생태계 VS Hadoop 생태계!

이것은 두 개의 다른 "기술 단지"와 같습니다:

- 각자의 특색이 있습니다
- 각자의 장점이 있습니다
- 하지만 모두 빅데이터 서비스를 위해 노력하고 있습니다!

Slide 39: 多个组件

看这些蓝色的齿轮图标！(**chǐ lún tú biāo)**
Spark由多个组件组成，
就像一台精密(**jīng mì)**的机器，
每个齿轮都有重要作用！
齿轮与齿轮之间完美配合，
才能让整台机器高效运转

Slide 40: 生态系统

Spark生态系统 VS Hadoop生态系统！

这就像两个不同的"科技园区"：

1. 各有各的特色
2. 各有各的优势
3. 但都在为大数据服务努力！

Slide 41: 범용성의 구체적 구성

Spark의 범용성은 네 가지 핵심 구성 요소로 반영됩니다:

- 🔵 **Spark SQL** - SQL 쿼리 전문가
- 🟢 **Spark Streaming** - 스트리밍 계산 고수
- 🔵 **Spark MLlib** - 머신러닝 마스터
- 🟢 **Spark GraphX** - 그래프 알고리즘 구성 요소 전문가

《서유기》의 사제사인(师徒四人)과 같아서, 각자 신통력을 가지고 팀을 이뤄 몬스터를 퇴치합니다!

《서유기》의 사제사인:

1. **당삼장(唐僧)**: 스승, 불경을 구하기 위해 서천으로 여정을 떠남
2. **손오공(孙悟空)**: "제천대성", 72변과 근두운, 여의봉 등 신통력 보유
3. **저팔계(猪八戒)**: 전직 천봉원수, 돼지 요괴 모습, 식탐과 나태함
4. **사오정(沙悟净)**: 전직 권장대장, 성실하고 충직한 성격

Slide 41:b通用性的具体组成

Spark的通用性体现在四个核心组件：

🔵 Spark SQL - SQL查询专家

🟢 Spark Streaming - 流式计算高手

🔵 Spark MLlib - 机器学习大师

🟢 Spark GraphX - 图算法组件专家

就像《西游记》的师徒四人，
各有神通，组队打怪！

《西游记》의 주요 등장인물인 사제사인(师徒四人, 스승과 제자 네 명)은 다음과 같습니다:
당삼장(唐僧, Tang Sanzang)
본명은 현장(玄奘)으로, 역사적 인물인 중국의 고승 현장을 모델로 함.

Slide 42: 실행 모드 다양성

Spark는 다양한 모드에서 실행할 수 있습니다: 독립 클러스터 모드, 그리고 다양한 플랫폼에서도 실행 가능합니다.

"만능 어댑터"와 같습니다:

- 어디에 꽂아도 사용 가능
- 다양한 환경에 적응
- 호환성이 초강력!

Slide 42: 运行模式多样性

Spark可以运行在多种模式：

独立集群模式，

还可以运行于各种平台。

就像一个"万能适配器(**shì pèi qì**)"：

1. 插在哪里都能用
2. 适应各种环境
3. 兼容性超强！

Slide 43: YARN 자원 관리 프레임워크

Hadoop은 자원 관리 프레임워크 YARN을 포함하며, 이것은 자원 스케줄링 관리자입니다.

YARN은 슈퍼 관리인과 같습니다:

- 모든 계산 자원을 관리
- 다른 작업들에게 합리적으로 할당
- 각 작업이 충분한 자원으로 실행되도록 보장

Slide 43: YARN资源管理框架

Hadoop包含资源管理框架YARN，

它是资源调度管理器。

YARN就像一个超级管家：

1. 管理所有的计算资源
2. 合理分配给不同的任务
3. 确保每个任务都有足够的资源运行

Slide 44: 클러스터 자원 스케줄링 관리 서비스

YARN은 Spark에게 클러스터 자원의 스케줄링 관리 서비스를 제공합니다.

이것은 다음과 같습니다:

- YARN은 물업관리 회사입니다
- Spark는 세입자입니다
- 물업이 수도, 전기, 가스 분배와 관리를 책임집니다
- 세입자가 자신의 일에 집중할 수 있게 합니다

Slide 44: 集群资源调度管理服务

YARN为Spark提供集群资源的调度管理服务。

这就像：

1. YARN是物业管理公司
2. Spark是租户(**zù hù)**
3. 物业负责水电煤气的分配和管理
4. 让租户专心做自己的事情

Slide 45: 클라우드 플랫폼 지원

Spark는 다양한 클라우드 플랫폼에서도 실행할 수 있습니다: 알리바바 클라우드, 텐센트 클라우드, AWS EC2 등.

"세계 여행가"와 같습니다:

- 어디를 가든 적응할 수 있습니다
- 어떤 "클라우드"에서든 정착할 수 있습니다
- 진정한 "클라우드 네이티브" 애플리케이션입니다!

Slide 45: 云平台支持

Spark还可以运行在各种云平台：阿里云、腾讯云、AWS EC2等。

这就像一个"环球旅行家"：

1. 到哪里都能适应
2. 在任何"云端"都能安家
3. 真正的"云原生"应用！

Slide 46: 다양한 데이터 소스 접근

Spark는 다양한 데이터 소스에 접근할 수 있습니다: HDFS, Cassandra, HBase, Hive 등.
"만능 열쇠"와 같습니다:

- HDFS 문을 엽니다 🚪
- Cassandra 문을 엽니다 🚪
- HBase 문을 엽니다 🚪
- Hive 문을 엽니다 🚪

모든 데이터 웨어하우스의 문이 Spark에게 열려 있습니다!

Slide 46: 多种数据源访问

Spark可以访问多种数据源：HDFS、Cassandra、HBase、Hive等。

就像一个"万能钥匙"：

1. 打开HDFS这扇门 (**shàn mén**) 🚪
2. 打开Cassandra这扇门 🚪
3. 打开HBase这扇门 🚪
4. 打开Hive这扇门 🚪

所有数据仓库的大门都向Spark敞开(**chǎng kāi**)！

Slide 47: Spark의 4대 핵심 우위

이 3D 입체 그림은 Spark의 네 가지 핵심 우위를 완벽하게 요약합니다:

- 🔵 **사용하기 쉬움** - 사용자 친화적
- 🟢 **빠른 실행 속도** - 뛰어난 성능
- 🟠 **범용성** - 포괄적인 기능
- 🔴 **다양한 실행 모드** - 강한 적응력

완벽한 정사각형과 같아서, 네 모서리가 모두 견고합니다!

Slide 47: Spark的四大核心优势

这张3D立体图完美总结了Spark的四大核心优势：

🔵 容易使用 - 用户友好
🟢 运行速度快 - 性能卓越 (**zhuó yuè**)
🟠 通用性 - 功能全面
🔴 运行模式多样 - 适应性强

就像一个完美的正方形，四个角都很稳固(wěn gù)

Slide 48: Google 트렌드 비교

이 트렌드 그래프는 매우 흥미진진합니다! Spark와 Hadoop의 Google 검색 열도 비교를 보여줍니다:

- 📈 2014년 Hadoop 정렬 기록을 경신한 후, Spark 열도가 급상승!
- 📈 2015년 발전 정점에 도달!
- 📈 파란선(Spark)이 최종적으로 빨간선(Hadoop)을 초월!

주식 차트처럼 흥분됩니다!

Slide 49: 유명 기업들의 적용

이 빛나는 회사 로고들을 보세요: 타오바오, 바이두, 텐센트, 아마존...

이것은 Spark가 이미 업계 거물들의 인정을 받았다는 것을 보여줍니다!

배우가 오스카상을 받아 자신의 실력을 증명한 것과 같습니다!

Slide 48: Google趋势对比

这张趋势图太精彩了！
显示了Spark与Hadoop在Google搜索热度上的对比：

📈 2014年打破Hadoop排序纪录后，Spark热度飙升！

📈 2015年达到发展高峰！

📈 蓝线(Spark)最终超越了红线(Hadoop)！

这就像股票图表一样激动人心！

Slide 49: 知名公司应用

看这些闪亮的公司logo：
淘宝、百度、腾讯、亚马逊...

这说明Spark已经得到了业界巨头们的认可！

就像一个演员获得了奥斯卡奖(**Àosīkǎ Jiǎng**)，证明了自己的实力！

Slide 50: Spark vs Hadoop 대비

이제 하이라이트에 진입합니다: Spark와 Hadoop의 상세 비교!

이것은 두 무림 고수의 최종 대결과 같습니다:

- 🔵 왼쪽: Spark (신세대 대표)
- 🟡 오른쪽: Hadoop (노련한 강자)

이 멋진 "화산논검"을 봅시다!

Slide 51: Hadoop의 구성 - MapReduce와 HDFS

Hadoop은 주로 두 부분으로 구성됩니다:

- ❌ **MapReduce** - 계산 엔진 (문제 있음)
- ✅ **HDFS** - 저장 시스템 (매우 좋음)

차와 같습니다:

- 엔진(MapReduce)에 문제가 있어 느리게 달립니다
- 하지만 연료탱크(HDFS)가 매우 커서 많은 연료를 담을 수 있습니다

Slide 50: Spark vs Hadoop 대비

现在我们进入重头戏(**zhòng tóu xì**)：Spark与Hadoop的详细对比！

这就像两个武林高手的终极对决：

🔵 左边：Spark（新生代代表）

🟡 右边：Hadoop（老牌劲旅**lǎo pái jìnglǚ**）

让我们看看这场精彩的"华山论剑"！

Slide 51: Hadoop的构成 - MapReduce与HDFS

Hadoop主要由两部分组成：

❌ MapReduce - 计算引擎（有问题）

✅ HDFS - 存储系统（很好用）

就像一辆车：

1. 发动机(MapReduce)有问题，跑得慢
2. 但油箱(HDFS)很大，能装很多油

Slide 52: MapReduce의 한계 - 표현력 제한

MapReduce의 첫 번째 문제: 표현력이 제한적입니다!
Map 함수와 Reduce 함수만 있어서, 다음과 같습니다:

- "덧셈"과 "뺄셈"만 있음
- 복잡한 수학 문제를 풀기 어려움
- 복잡한 비즈니스 로직을 표현하기 어려움

분산 애플리케이션 개발의 복잡성을 낮췄지만, 표현력도 제한했습니다!

Slide 52: MapReduce的局限 - 表达能力限制

MapReduce的第一个问题：表达能力有限！

只有Map函数和Reduce函数，就像：

1. 只有"加法"和"减法"
2. 要做复杂数学题就很困难
3. 复杂业务逻辑难以表达

降低了分布式应用开发的复杂性，但也限制了表达能力！

Slide 53: 복잡한 비즈니스 로직의 제한

복잡한 비즈니스 로직이 Map 함수와 Reduce 함수로 제한되어, 표현력이 실제로 제한됩니다.

"예"와 "아니오"로 모든 질문에 답하는 것과 같습니다:

- 간단한 질문은 가능
- 복잡한 질문은 명확하게 답할 수 없음
- 더 풍부한 "어휘 목록"이 필요!

Slide 53: 复杂业务逻辑的限制

复杂业务逻辑被限制在Map函数和Reduce函数中，这确实限制了表达能力。

就像用"是"和"否"来回答所有问题：

1. 简单问题可以
2. 复杂问题就说不清楚了
3. 需要更丰富的"词汇表"！

Slide 54: 디스크 IO 오버헤드

MapReduce의 두 번째 문제: 디스크 IO 오버헤드가 큽니다!
중간 결과를 반드시 디스크에 써야 하므로, 다음과 같습니다:

- 요리할 때마다 채소를 자를 때마다 손을 씻어야 함
- 숙제를 할 때마다 글자를 쓸 때마다 펜을 다시 잡아야 함

효율이 자연히 매우 낮습니다!

Slide 54: 磁盘IO开销

MapReduce的第二个问题：磁盘IO开销大！

中间结果必须写入磁盘，就像：

1. 做菜时每切一个菜都要洗一次手
2. 写作业时每写一个字都要重新拿笔

效率自然很低！

Slide 55: 높은 지연

MapReduce는 두 단계로 나뉩니다: Map 단계와 Reduce 단계.

릴레이 경주와 같습니다:

- 첫 번째 주자가 끝나면 두 번째 주자를 기다려야 합니다
- 두 번째 주자가 시작하기 전에 첫 번째 주자가 모두 완료되기를 기다려야 합니다
- 중간의 대기 시간이 바로 지연입니다!

Slide 55: 高延迟

MapReduce分为两个阶段：Map阶段和Reduce阶段。

就像接力赛：

1. 第一棒跑完要等第二棒 **bàng**
2. 第二棒开始前要等第一棒全部完成
3. 中间的等待时间就是延迟！

Slide 56: 단계 간 대기 시간

이 그림은 Map 단계와 Reduce 단계 사이의 대기 관계를 보여줍니다.

Map 단계는 학생 시험과 같고, Reduce 단계는 선생님의 채점과 같습니다:

- 모든 학생이 시험을 마치고 시험지를 수거
- 그런 다음 선생님이 채점을 시작할 수 있음
- 중간 대기 시간이 매우 깁니다!

Slide 56: 阶段间等待时间

这张图展示了Map阶段和Reduce阶段之间的等待关系。

Map阶段就像学生考试，Reduce阶段就像老师改卷(**gǎi juàn**)：

1. 所有学生考完试，试卷收齐(shōu qí)
2. 老师才能开始改卷
3. 中间的等待时间很长！

Slide 57: Map과 Reduce의 처리 흐름

이 그림은 Map 함수와 Reduce 함수 사이의 작업 흐름을 보여줍니다.

작업 → Map 함수 → 출력 → 중간 결과 ← 처리 ← Reduce 함수

공장의 조립 라인과 같습니다:

- Map은 "부품 가공" 단계
- Reduce는 "최종 조립" 단계
- 중간에 반드시 "창고 저장" 과정이 있어야 함!

Slide 57: Map和Reduce的处理流程

这张图展示了Map函数和Reduce函数之间的工作流程。

任务 → Map函数 → 输出 → 中间结果 ← 处理 ← Reduce函数

就像工厂的流水线：

1. Map是"零件加工"阶段
2. Reduce是"最终组装"阶段
3. 中间必须有"仓库存储"过程！

Slide 58: Map과 Reduce의 병렬 처리

이 그림은 여러 Map과 Reduce가 병렬 처리되는 상황을 보여줍니다:

- **Map 단계**: Map1, Map2, Map3, Map... (병렬 처리)
- **Reduce 단계**: Reduce1, Reduce2, Reduce3, Reduce... (병렬 처리)

여러 요리사가 동시에 요리하는 것과 같지만, 두 단계 사이에서는 반드시 모든 Map이 완료되기를 기다려야 합니다!

Slide 58: 전체 Map 완료 대기

이 그림의 핵심은 "전체 Map 완료"와 "대기" 개념입니다!

학교 반 활동과 같습니다:

- 모든 학생이 도착해야 출발할 수 있습니다
- 99%의 학생이 도착해도 마지막 한 명을 기다려야 합니다

Map도 마찬가지로, 전체가 완료되어야만 Reduce 단계로 들어갈 수 있습니다!
돋보기가 "전체 Map 완료"의 중요성을 강조하고 있습니다.

Slide 58: Map和Reduce的并行处理

这张图展示了多个Map和Reduce并行处理的情况：

1. Map阶段: Map1, Map2, Map3, Map... (并行处理)
2. Reduce阶段: Reduce1, Reduce2, Reduce3, Reduce... (并行处理)

就像多个厨师同时做菜，但是两个阶段之间必须等待所有Map完成！

Slide 58: 全部Map完成等待(계속 설명)

这张图的8心是"完成全部Map"和"等待"的概念！

就像学校的班级活动

1. 所有同学到齐才能出发
2. 即使99%的同学都到了，也要等最后一个

Map也是这样，必须全部完成才能进入Reduce阶段！
中间这个放大镜(**fàng dà jìng**)突出了"完成全部Map"的重要性。

Slide 59: 연결 IO 오버헤드

MapReduce의 세 번째 문제: 연결의 IO 오버헤드가 큽니다!

Map 후 결과 → 쓰기 → 디스크 → 읽기 → Reduce

이것은 다음과 같습니다:

- 편지를 쓴 후 우체통에 넣음
- 우체부가 와서 가져감
- 다시 수신자에게 전달

직접 전달할 수 있는데, 이렇게 큰 원을 그려야 합니다!

Slide 60: MapReduce의 세 가지 결함

이 원형 차트는 MapReduce의 세 가지 주요 문제를 요약합니다:

- 🔵 **디스크 IO 오버헤드 큼** - 빈번한 디스크 읽기/쓰기
- 🟠 **높은 지연** - 단계 간 긴 대기 시간
- 🟢 **연결 IO 오버헤드 큼** - 중간 단계가 너무 많음

세 가지 문제가 서로 연관되어 성능 병목을 형성합니다!

Slide 59: 衔接IO开销

MapReduce的第三个问题：衔接的IO开销大！

Map后结果 → 写入 → 磁盘 → 读取 → Reduce

这就像：

1. 写信后放进邮箱
2. 邮递员来取走
3. 再送到收件人手中

明明可以直接传递，却要绕这么大一圈！

Slide 60: MapReduce的三大缺陷

这个圆形图表总结了MapReduce的三个主要问题：

- 🔵 磁盘的IO开销大 - 频繁读写磁盘
- 🟠 延迟高 - 阶段间等待时间长
- 🟢 衔接的IO开销大 - 中间环节太多

三个问题相互关联，形成了性能瓶颈(**píng jǐng)**！

Slide 61: Spark의 Hadoop 결함 개선과 계승

이 그림은 기술 진화의 아름다운 과정을 보여줍니다!
Hadoop → 결함 개선과 계승 → Spark

휴대폰 발전과 같습니다: 피처폰에서 스마트폰으로, 통화 기능은 유지하면서 수많은 새로운 기능을 추가했습니다!

Spark는 빅데이터 분야의 "스마트폰"입니다!

Slide 61: Spark对Hadoop进行缺陷改进与继承

这张图展示了技术进化的美妙过程！
Hadoop → 缺陷改进与继承 → Spark

就像手机的发展：从功能机到智能机，保留了通话功能，但增加了无数新特性！

Spark就是大数据领域的"智能手机"！

Slide 62: Spark의 풍부한 함수

Spark는 브라우저 창에서 풍부한 함수 라이브러리를 보여줍니다:

Map 함수, Reduce 함수, 그리고:

- **filter** (필터링)
- **join** (조인)
- **groupBy** (그룹화)

칼만 있던 것에서 전체 주방 도구 세트를 갖춘 것과 같습니다! 요리할 수 있는 가능성이 크게 증가했습니다!

Slide 62: Spark的丰富函数

Spark在浏览器(**liú lǎn qì)**窗口(**chuāng kǒu)**中展示了丰富的函数库：

Map函数、Reduce函数，还有：

1. filter (过滤)
2. join (连接)
3. groupBy (分组)

就像从只有菜刀变成了拥有整套厨具！做菜的可能性大大增加了！

Slide 63: Spark의 풍부한 함수

이 아름다운 별 모양 퍼즐은 Spark의 다양한 작업을 보여줍니다:

- ⭐ **Map** (매핑)
- ⭐ **Reduce** (리듀스)
- ⭐ **filter** (필터링)
- ⭐ **join** (조인)
- ⭐ **groupBy** (그룹화)

각 퍼즐 조각이 완벽하게 맞물려 강력한 Spark 도구 상자를 구성합니다!

Slide 64: 더 강한 표현력

"더 많은 작업 유형 → 더 강한 표현력"
이것은 간단하면서도 심오한 진보입니다:

- 어휘가 풍부할수록 표현이 정확
- 도구가 다양할수록 솔루션이 유연
- Spark는 더 복잡한 데이터 처리 요구사항을 "표현"할 수 있게 합니다!

Slide 63: Spark的丰富函数

这个美丽的星形拼图展示了Spark的各种操作：

⭐ Map (映射)
⭐ Reduce (归约)
⭐ filter (过滤)
⭐ join (连接)
⭐ groupBy (分组)

每个拼图块都完美契合(**qì hé**)，组成了强大的Spark工具箱！

Slide 64: 表达能力更强

"操作类型更多 → 表达能力更强"
这是一个简单而深刻的进步：

- 词汇越丰富，表达越精确
- 工具越多样，解决方案越灵活
- Spark让我们能够"说出"更复杂的数据处理需求！

Slide 65: 메모리 계산의 우위

Spark의 핵심 혁신: 메모리 계산!

계산 과정 중 → 중간 결과 → 메모리에 저장
이것은 다음과 같습니다:

- 모든 도구를 작업대 위에 놓아 손쉽게 가져갈 수 있음
- 매번 도구 상자에서 찾을 필요가 없음

효율 향상이 분명합니다!

Slide 66: 재접근 시 낮은 지연

메모리의 중간 결과를 재접근할 때 낮은 지연을 실현합니다!

자주 사용하는 앱을 휴대폰 홈 화면에 두는 것과 같습니다:

- 클릭하면 즉시 사용 가능
- 앱 목록에서 검색할 필요 없음

이것이 메모리 계산의 위력입니다! 🚀

Slide 65: 内存计算的优势

Spark的核心创新：内存计算！

计算过程中 → 中间结果 → 存储在内存中

这就像：

1. 把所有工具放在工作台上，随手可得
2. 而不是每次都去工具箱里翻找

效率提升是显而易见(xiǎn ér yì jiàn)的！

Slide 66: 再次访问时的低延迟

内存中的中间结果再次访问时实现了低延迟！

就像把常用的App放在手机桌面：

1. 点击就能立即使用
2. 不需要在应用列表中搜索

这就是内存计算的威力！🚀

Slide 67: 반복 연산 효율 향상

실행 효율 향상 + 반복 연산의 효율적 개선
이 순환 차트는 지속적인 개선 과정을 보여줍니다.

머신러닝과 데이터 마이닝에서 반복 계산은 매우 일반적입니다:

- 각 반복마다 이전 결과를 활용할 수 있음
- 메모리 저장으로 이 과정이 매우 빨라짐
- 학습과 같아서, 옛 것을 익혀 새것을 앎!

Slide 67: 迭代运算效率提升

运行效率提升 + 高效提高迭代运算

这个循环图表展示了持续(**chí xù**)改进的过程。

在机器学习和数据挖掘中，迭代计算非常常见：

1. 每次迭代都能利用上次的结果
2. 内存存储让这个过程变得飞快
3. 就像学习一样，温故而知新！

Slide 68: DAG 작업 스케줄링 실행 메커니즘

DAG 작업 스케줄링 실행 메커니즘 + DAG - 유향 비순환 그래프

오른쪽 그래프는 복잡한 의존 관계를 보여줍니다:

- 각 노드는 계산 작업을 나타냄
- 화살표는 데이터 흐름과 의존 관계를 나타냄
- 순환이 없어 실행의 결정성을 보장

정교하게 설계된 도시 교통망과 같아서, 차량이 질서정연하게 흐르고 막히지 않습니다!

Slide 68: DAG任务调度执行机制

DAG任务调度执行机制 + DAG—有向无环图

右侧的图形展示了复杂的依赖关系：

1. 每个节点代表一个计算任务
2. 箭头(**jiàn tóu**)表示数据流向和依赖关系
3. 没有循环，确保了执行的确定性

这就像一张精心设计的城市交通网络，车流有序，不会堵塞(**dǔ sè**)！

Slide 69: MapReduce의 반복 처리 문제

MapReduce를 사용한 반복 처리의 첫 단계:

반복1 → Hadoop MapReduce → 결과1 → 디스크에 저장

매번 숙제할 때마다:

- 책장에서 책을 꺼냄
- 숙제 완료
- 책을 다시 책장에 넣음
- 다음에 다시 처음부터 시작

이런 방식은 여러 번 반복이 필요한 알고리즘에 대해 효율이 너무 낮습니다!

Slide 70: 첫 번째 반복의 상세 과정

반복1 → Map + Reduce → 결과1
전체 과정에서 중간 결과를 반드시 디스크에 써야 합니다.
연속적인 작업에 너무 번거롭습니다!

Slide 69: MapReduce的迭代处理问题

使用MapReduce进行反复迭代的第一步：

迭代1 → Hadoop MapReduce → 结果1 → 存储到磁盘

就像每次做作业都要：

1. 从书架拿书
2. 做完作业
3. 把书放回书架
4. 下次再从头开始

这种方式对于需要多次迭代的算法来说效率太低！

Slide 70: 第一次迭代的详细过程

迭代1 → Map + Reduce → 结果1
整个过程中，中间结果必须写入磁盘。

对于连续的工作来说太麻烦了！

Slide 71: 반복이 필요한 알고리즘

로지스틱 회귀와 시뮬레이티드 어닐링 알고리즘은 모두 반복이 필요합니다.

이러한 알고리즘은 다음과 같습니다:

- 산을 올라 가장 높은 봉우리 찾기
- 각 단계가 이전 단계의 위치를 기반으로 함
- 최적 해를 찾을 때까지 계속 조정해야 함

하지만 MapReduce는 매번 "산에서 내려와 시작점으로 돌아가야" 하므로 효율이 자연히 매우 낮습니다!

Slide 71: 需要反复迭代的算法

逻辑斯蒂回归和模拟退火算法都需要反复迭代。

这些算法就像：

1. 爬山寻找最高峰
2. 每一步都基于前一步的位置
3. 需要不断调整直到找到最优解

但MapReduce每次都要"下山回到起点"，效率自然很低！

Slide 72: 수렴하지 않았을 때의 낮은 효율

반복 → 수렴하지 않음 → 계속 반복 구현 → Hadoop MapReduce → 효율 매우 낮음

알고리즘이 수렴하기 위해 많은 반복이 필요할 때, MapReduce의 단점이 명확히 드러납니다. 매번 다시 시작해야 하는 마라톤 경기와 같습니다!

Slide 72: 未收敛时的低效率

迭代 → 没有收敛 (**shōu liǎn**)→ 继续迭代 实现 → Hadoop MapReduce → 效率非常低

当算法需要很多次迭代才能收敛时，MapReduce的劣势(**liè shì**)就暴露无遗了(**bào lù wú yí le**)。就像每次都要重新开始的马拉松比赛！

Slide 73: 첫 번째 반복의 상세 과정

반복1 → Map + Reduce → 결과1
전체 과정에서 중간 결과를 반드시 디스크에 써야 합니다.

이것은 다음과 같습니다:

- 문단을 작성
- 파일에 저장
- 문서를 닫음
- 다음에 다시 열기

연속적인 작업에 너무 번거롭습니다!

Slide 74: 두 번의 반복 완전한 흐름

반복1 → Hadoop MapReduce → 결과1(디스크) → 반복2 → Hadoop MapReduce → 결과2

각 반복 사이에 디스크 읽기/쓰기 작업이 있어 효율을 크게 떨어뜨립니다. 두 사람이 대화할 때 한 문장마다 쪽지에 써서 전달하는 것과 같습니다!

Slide 73: 第一次迭代的详细过程

迭代1 → Map + Reduce → 结果1
整个过程中，中间结果必须写入磁盘。

这就像：

1. 写一段话
2. 保存到文件
3. 关闭文档
4. 下次再重新打开

对于连续的工作来说太麻烦了！

Slide 74: 两次迭代的完整流程

迭代1 → Hadoop MapReduce → 结果1(磁盘) → 迭代2 → Hadoop MapReduce → 结果2

可以看到，每次迭代之间都有磁盘读写操作，这大大降低了效率。
就像两个人对话，每说一句话都要写纸条传递！

Slide 75: 데이터 낙하 방지

Spark의 해결책: 데이터 낙하 방지 + 디스크에 쓰지 않음!

저 금지 표시가 보이시나요?
Spark가 말합니다: "우리는 디스크가 필요 없습니다!"

- ❌ 빈번한 디스크 작업 금지
- ✅ 모두 메모리에서 처리

두뇌로 생각하는 것과 같아서, 펜을 들 필요가 없습니다!

Slide 76: 파이프라인 최적화의 데이터 흐름

파이프라인 최적화는 작업 간 데이터 흐름을 보여줍니다:

작업1 → 데이터 생성 → 데이터 교환(메모리 통해) → 데이터 사용 → 작업2

중간의 메모리 모듈이 빠른 데이터 전달을 보장하여, 컨베이어 벨트처럼 부드럽습니다!

Slide 75: 避免数据落地

Spark的解决方案：
避免数据落地 + 不写磁盘！

看到那个禁止标志吗？
Spark说："我们不需要磁盘！"

- ❌ 禁止频繁的磁盘操作
- ✅ 全部在内存中处理

就像在大脑中思考，不需要动笔！

Slide 76: 流水线优化的数据流

流水线优化展示了数据在操作间的流动：

操作1 → 数据生成 → 数据交换(通过内存) → 使用数据 → 操作2

中间的内存模块确保了快速的数据传递，就像传送带一样流畅(**liú chàng)**！

Slide 77: Hadoop과 Spark 실행 흐름 비교(전체)

이 비교 그림은 매우 중요합니다!
상단(Hadoop MapReduce 실행 흐름):
HDFS 읽기 → 반복1 → HDFS 쓰기 → HDFS 읽기 → 반복2 → HDFS 쓰기...

- 각 단계마다 디스크 작업 포함

하단(Spark 실행 흐름): 읽기 → 반복1 → 메모리에 저장 → 읽기 → 반복2 → 메모리에 저장...

- 주로 메모리에서 진행

차이가 한눈에 보입니다!

Slide 77: Hadoop和Spark执行流程对比(完整)

这个对比图非常重要！
上部分(Hadoop MapReduce执行流程)：
HDFS读取 → 迭代1 → HDFS写入 → HDFS读取 → 迭代2 → HDFS写入...

- 每个步骤都涉及磁盘操作

下部分(Spark执行流程)：读取 → 迭代1 → 存储在内存中 → 读取 → 迭代2 → 存储在内存中...

- 主要在内存中进行

差别一目了然！

Slide 78: Hadoop MapReduce 상세 흐름

상단은 Hadoop의 실행 과정을 상세히 보여줍니다:
입력 → HDFS 읽기 → 반복1 → HDFS 쓰기 → HDFS 읽기 → 반복2 → HDFS 쓰기 → ...
하단은 쿼리 처리를 보여줍니다: 입력 → 쿼리1 → 결과1, 쿼리2 → 결과2...
각 단계마다 "디스크 환승역"을 거쳐야 합니다!

Slide 78: Hadoop MapReduce详细流程

上半部分详细展示了Hadoop的执行过程：
输入 → HDFS读取 → 迭代1 → HDFS写入 → HDFS读取 → 迭代2 → HDFS写入 → ...

下半部分显示查询处理： 输入 → 查询1 → 结果1, 查询2 → 结果2...

每一步都要经过"磁盘中转站"！

Slide 79: Spark 상세 실행 흐름

Spark의 개선된 흐름:
입력 → 읽기 → 반복1 → 메모리에 저장 → 읽기 → 반복2 → 메모리에 저장 → ...
하단: 입력 → 메모리에 저장 → 쿼리1 → 결과1, 쿼리2 → 결과2...

모든 중간 과정이 "메모리 고속도로"에서 진행됩니다!

Slide 79: Spark详细执行流程

Spark的改进版流程：
输入 → 读取 → 迭代1 → 存储在内存中 → 读取 → 迭代2 → 存储在内存中 → ...
下半部分： 输入 → 存储在内存中 → 查询1 → 结果1, 查询2 → 结果2...

所有中间过程都在"内存高速公路"上进行！

Slide 80: 놀라운 성능 비교 결과

이 막대 차트는 충격적인 진실을 밝힙니다!

- **Hadoop**: 110초 (거대한 파란색 막대)
- **Spark**: 0.9초 (거의 보이지 않는 작은 막대)

110초 vs 0.9초! 약 122배의 성능 차이입니다!

이것은 걷기와 고속철을 비교하는 것과 같습니다:

- 걸어서 1시간 걸리는 거리
- 고속철로는 단 30초!

이것이 메모리 컴퓨팅 혁명의 위력입니다!

Slide 80: 惊人的性能对比结果

这个柱状图(**zhù zhuàng tú**)揭示了震撼(**zhèn hàn**)的真相！

- Hadoop: 110秒 (巨大的蓝色柱子)
- Spark: 0.9秒 (几乎看不见的小柱子)

110秒 vs 0.9秒！这是约122倍的性能差异！

这就像比较步行和坐高铁的区别：

- 步行1小时的路程
- 高铁只需要30秒！

这就是内存计算革命的威力！

퀴즈 정답

- **Slide 81**: ✅ 정답: C
- **Slide 82**: ✅ 정답: B
- **Slide 83**: ✅ 정답: B
- **Slide 84**: ✅ 정답: B
- **Slide 85**: ✅ 정답: C (110초 vs 0.9초, 약 122배)
- **Slide 86**: ✅ 정답: B
- **Slide 87**: ✅ 정답: D
- **Slide 88**: ✅ 정답: D (MapReduce의 문제는 과도한 디스크 의존이며, 메모리 사용 효율 문제가 아님)
- **Slide 89**: ✅ 정답: B
- **Slide 90**: ✅ 정답: B

퀴즈 정답

Slide 81:✅ 正确答案：C

Slide 82:正确答案：B

Slide 83:正确答案：B

Slide 84:正确答案：B

Slide 85: 正确答案：C（110秒 vs 0.9秒，约122倍）

Slide 86:正确答案：B

Slide 87:正确答案：D

Slide 88:

正确答案：D（MapReduce的问题是过度依赖磁盘，而不是内存使用效率问题）

Slide 89:正确答案：B

Slide 90:正确答案：B

Slide 91: 마무리

좋습니다, 이번 수업은 여기서 마치겠습니다!
오늘 우리는 "빅데이터 시대"의 전말을 알아보았고, 정보화 물결, 세 가지 지원 기술부터 데이터가 어떻게 생성되고 어떻게 진화하는지까지 살펴보았습니다.
다음 수업에서는 "빅데이터의 핵심 기술"에 대해 계속 탐구하겠습니다.
마지막으로 질문 하나: 오늘 여러분은 자신이 매일 사실 "데이터를 생산"하고 있다는 것을 발견하셨나요? 😉

감사합니다, 수업 마치겠습니다! 👏👏👏

Slide 91:

好了，我们本节课到这里就结束啦！
今天我们认识了“大数据时代”的来龙去脉，从信息化浪潮、三大支撑技术，到数据如何产生、如何演变。

下节课我们会继续探讨“大数据的关键技术”。

最后提问一下：你今天有没有发现，自己每天其实也在“生产数据”呢？😉

谢谢大家，下课！👏👏👏