



東南大學  
SOUTHEAST UNIVERSITY

# IT 新技术讲座

## 报告

学生姓名: 白丰硕

学生学号: 71116233

东南大学计算机科学与工程学院、软件学院

School of Computer Science & Engineering

College of Software Engineering

Southeast University

二 0 一 九 年 五 月

# 《计算机新技术讲座》第一次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**开源技术在机器学习中的应用

**报告人信息：**

刘石彬，IBM 资深研发工程师。2007 年加入 SPSS,长期从事统计分析，数据挖掘，机器学习类的软件研发工作。在工作中接触大量客户和使用场景，对统计分析、关联规则、时间序列方面有比较深的了解。

**企业信息：**

IBM 是国际商业机器股份有限公司的缩写，是美国一家跨国科技公司及咨询公司，总部位于纽约州阿蒙克市。IBM 主要客户是政府和企业。IBM 生产并销售计算机硬件及软件，并且为系统架构和网络托管提供咨询服务。截止 2013 年，IBM 已在全球拥有 12 个研究实验室和大量的软件开发基地。IBM 虽然是一家商业公司，但在材料、化学、物理等科学领域却也有很高的成就，利用这些学术研究为基础，发明很多产品。

**讲座内容：**

主要介绍了面对一个机器学习问题，解决问题的步骤和过程。可能会遇到的问题和对数据集的处理，此外还有对 scikit-learn 等机器学习的库使用的介绍。

## 二、收获和扩展学习

**报告收获：**

通过这次报告，让我们了解到了机器学习这个领域的一个概况。面对一个问题的时候，从哪里作为问题的剖析点和着眼点，遇到问题的时候如何进行理解和分析有了一个简单的认识。还了解到一些机器学习领域中的基本的算法，比如聚类算法 k-means 等。机器学习的很多算法涉及到一些艰涩难懂的数学知识，让不少人望而却步。但是对于新手学习而言，直接复现这些算法似乎还是比较有困难还挑战的一件事，但是很多开源的工具和库可以有现成的函数让我们直接使用，这样的话，算法的本身变成了一个黑盒的过程。编程者可以通过直接调用函数完成一些数据的训练，这样很有利于学习者从直观的角度去学习这些艰涩难懂的算法，而且更容易产生学习的兴趣。

**扩展学习：**

对于其他开源的工具，大家耳熟能详的是 Tensorflow、Keras、PyTorch 和另一些作为顶级框架的机器学习和深度学习库。在近一年中，还涌现出很多优秀的开源框架。如：亚马逊 Sagemaker 这种新的框架简化了构建机器学习模型并部署到云端的任务。这项服务对于并不深入了解机器学习的开发人员来说非常有用，因为它为开发人员提供了一系列预先构建的开发环境，基于流行的 Jupyter 笔记本格式。如果数据科学家不希望花费大量时间，就可以在 AWS 上构建有效的机器学习系统，并对性能进行微调，就会发现这项服务大有用处。还有微软 Azure 机器学习工具台是一个跨平台客户软件，它在 Windows 机器和苹果机器上都可以运行。它是为想要执行数据操纵和处理任务的数据科学家和机器学习开发人员量身打造的。它为确保可扩展性而构建，用户可以从一系列广泛的数据源获得直观的洞察力，并用于数据建模任务。很多大公司的机器学习平台都开始注重云端的学习，而不仅仅是用户单机的行为。这可能也是现代人工智能学习框架的一种发展的趋势。

# 《计算机新技术讲座》第二次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**区块链服务平台构建简介

**报告人信息：**

孙宏伟，IBM 中国系统实验室顾问软件工程师。主要从事 IBM 主机 z/OS 系统的系应用开发，系统测试，及相关工具的开发工作。

**企业信息：**

IBM 是国际商业机器股份有限公司的缩写，是美国一家跨国科技公司及咨询公司，总部位于纽约州阿蒙克市。IBM 主要客户是政府和企业。IBM 生产并销售计算机硬件及软件，并且为系统架构和网络托管提供咨询服务。截止 2013 年，IBM 已在全球拥有 12 个研究实验室和大量的软件开发基地。IBM 虽然是一家商业公司，但在材料、化学、物理等科学领域却也有很高的成就，利用这些学术研究为基础，发明很多产品。比较有名的 IBM 发明的产品包括硬盘、自动柜员机、通用产品代码、SQL、关系数据库管理系统、DRAM 及沃森。

**讲座内容：**

讲解了区块链的平台如何构建，以及区块链的服务如何在平台上构建，区块链技术在实际生产中的使用。

## 二、收获和扩展学习

**报告收获：**

通过这次报告，让我们了解到了区块链服务平台如何进行快速的搭建。在此之前，在实训期间，做过关于区块链的项目，对区块链的有一定的认识。对于使用以太坊进行开发的区块链平台，需要使用 solidity 来编写底层的执行合约。其中有一个很好的框架叫 truffle，使用这个框架可以快速构建我们的区块链平台。

对于区块链服务，是指利用区块链产生的数据，提供基于区块链的搜索查询、任务提交，等一系列操作服务。目前在区块链领域，只有各大公链的区块浏览器能称之为区块链服务，以及部分公链衍生应用：存证型-Factom，数字身份型-uPort 等。这些应用都有个特点，基于已存在的公链，发挥并强化公链已有的功能，让公链为大家提供更好的区块链服务。也就是说，如果要开发一个区块链服务型应用，一般指在公链上开发一个应用。目前在区块链领域，主要是 HyperLedger，multichain，以太坊私有链等多个框架去构建多个实例，去解决已经被证明使用传统互联网技术能很好解决的问题，一般称“强行区块链”。这些应用也有典型特点，希望构建属于自己的区块链系统，其实称之为分布式账本技术更合适些。在这种情况下，更推荐是基于 PBFT 协议的分布式数据库，而不是私有区块链系统。

**扩展学习：**

对于构建一个区块链服务。如在以太坊上使用智能合约开发公链应用，并在以太坊节点上运行对公众提供有效服务。在比特币上，利用比特币有限的功能，提供一些存证服务。区块链开发者应当建立有效的渠道，能够收集当前区块链的不足，并进行针对性的改进，这种过程我们称之为 Improvement Proposals（改进提案）：比特币隔离见证（由一系列 BIP 组成）就是一个很好的例子。应用层的开发者利用 IP 过程可以将使用过程中发现的问题反馈给开发者，形成一个良

性循环。

#### 区块链的应用场景：

1. 金融行业对于经济发展具有极大的意义，被誉为全球经济发展的第一动力。支付就是金融的重要组成部分之一，特别是跨境支付。然而现在的跨境支付效率十分低，一方面是汇率比较高，另一个则是速度比较慢。这是需要跨境汇款的人群最大的困扰，也是这个行业的一大痛点。比特币设计之初的本质是点对点传输，当其底层技术区块链应用于支付领域的时候，将点对点传输的优点带入，就可以很好地解决跨境支付的痛点。同时，区块链具有不可篡改和智能合约等技术特点，在基于区块链的跨境支付过程中，汇款通道透明，具有可追溯性，汇款方和收款方都有了可靠的保证，支付更加安全。
2. 游戏是传统互联网的三驾马车之一，具有用户规模大，收入能力强的特点。作为离钱、离用户最近的领域，游戏总是能第一时间拥抱新技术，并成为新技术最重要的应用场景。经过数十年的发展，传统游戏现在已经遇到一些瓶颈。游戏一直归开发商所控制。在游戏里，玩家的资产不完全由自己控制，一旦开发商倒闭，游戏就结束，玩家此前在游戏中投入的时间、精力和资金等就全部消失。将区块链技术应用于游戏当中，可以将游戏的所有权下放给所有的玩家，游戏不再受中心组织的控制。游戏交由玩家社区自治，运行机制透明，每一笔交易数据都上链不可篡改，避免了中心组织捏造虚假数据。玩家对自己的资产持有真正的把控权，当游戏开发公司倒闭后，游戏仍然在运营，玩家的资产也不会消失。
3. 食品溯源。食品从原材料到到达消费者这个过程，中间经历了很多很复杂的工序，包括生产、批发、销售等，过程中可能会存在各种各样的问题，使得整体生产效率低下。如虚假产品的混入、中间商赚取差价等问题，使得消费者最终获得的产品质量或性价比并非最佳。虽然传统的运转中也有溯源，但是这一过程都是有人的参与，有人的地方就总是会有虚假的存在，这些问题并不能得到有效的解决。透明性作为区块链的特点之一，因此区块链应用于食品溯源链成为了必然。通过使用分布式账本技术，供应链上的所有数据都记录到这一分布式的账本上，这一账本归网络中的所有节点共同维护更改，保证了数据的真实与不可篡改。一个应用了区块链溯源链的产品，从生产到消费者手上的每一个流程步骤都是透明化的。

# 《计算机新技术讲座》第三次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**scala 语言之美

**报告人信息：**

杨凯，南京研发中心技术副总经理。2010 年本科毕业，曾参与公司检察院产品、技术架构模型、音视频、大数据等项目的核心编码和架构设计

**企业信息：**

北京华宇软件股份有限公司（以下简称华宇）以软件与信息服务为主营业务，创立于 2001 年 6 月 18 日，华宇秉承“自强不息，厚德载物”的企业精神，坚持诚信为本的价值观，以“提供专业的技术、优秀的产品和卓越的服务，以信息化创造客户价值”为使命，专注于电子政务领域的软件与信息服务，为客户的信息化事业提供全方位的解决方案与服务，是国内电子政务领域的领导者。

**报告内容：**

课程讲述了当前大数据处理的热门语言—scala，采用 scala 语言通过 spark 分布式处理框架进行案例演示，以及介绍大数据和人工智能在企业级应用中的案例。

## 二、收获和扩展学习

**报告收获：**

通过这次报告，让我们了解到了 Scala 在大数据和人工智能在企业级应用中的一些使用场景。Scala 是一种有趣的语言。它一方面吸收继承了多种语言中的优秀特性，一方面又没有抛弃 Java 这个强大的平台，它运行在 Java 虚拟机 (Java Virtual Machine) 之上，轻松实现和丰富的 Java 类库互联互通。它既支持面向对象的编程方式，又支持函数式编程。它写出的程序像动态语言一样简洁，但事实上它确是严格意义上的静态语言。Scala 就像一位武林中的集大成者，将过去几十年计算机语言发展历史中的精萃集于一身，化繁为简，为程序员们提供了一种新的选择。

● 相比较与 Java，在下觉得 Scala 最主要的有以下两点优势：

1. FP 泛型支持

如果用多了 Spring 中大量的 template 接口，你就会觉得 FP 其实还是蛮好用的。

而这仅仅是 FP 好处的冰山一角。函数其实就是一个 input -> output (scala 也是这么表示一个函数的)，没有任何副作用，与状态无关，由于这种特性，所以函数式的编程范式在分布式领域有很多好处。对于函数式编程，我的知识实在是皮毛，但可以这么说，FP 相对与 OO 有哪些优势，Scala 对于 Java 差不多就有哪些优势。正因为 FP 有如此多的优势，所以 Java8 才引入了 FP。从某种程度上来说，Java 认可了 Scala 的做法。

2. 类型系统支持

如果说 Java 是一种类型安全的语言，那么毫无疑问，Scala 的类型更加安全，从某种程度上说，Scala 的类型是图灵完备的，而 Java 不是。我的一位好朋友在这方面研究的比较

深。正是以上这两点大优势，造成了 Scala 比 Java 更加安全，同时又具备灵活性，想象力。

### 3. 其他语言层面上的优势

在 Java 中，你是否有时很想继承多个 AbstractClass 呢？但是 Java 只支持单继承，在 Scala 中，你可以进行 mixin；在 Java 中，想要一个 singleton？要么在 static block 中做，要么利用 Enum 的单例特性完成，或者其他更纠结的方法。在 Scala 中，只要声明为 object，即为单例；在 Java 中，想要延迟加载一个单例？double check 吧在 Scala 中，只要在 object 中将变量修饰为 lazy 即可；在 Java 中，想要对集合进行一些操作？使用一层一层的 for 循环，在 Scala 中，使用 collection 的一些集合操作，即可获得如写 SQL 般的享受；在 Java 中，在并发中想对 Future 进行回调？Future 不是 Listenable (无法支持回调)，除非你使用额外的工具(如 guava, spring)，在 Scala 中，本来就主张异步编程，future 和 promise 的配合让人非常愉快；在 Java 中，要透明扩展一个第三方库的类怎么办？包装，再加一层。在 Scala 中，有强大的 implicit 机制让你更优雅的做到这一点，同时还能保证类型安全；

#### ● 尽管罗列了如此多的好处，但 Scala 有如下劣势：

1. 语法复杂，学习曲线非常高
2. 国内 Scala 程序员很难找（目前 Scala 的影响力也在缓慢扩大，比如 Scala 社区中的明星 Spark 的流行也在慢慢拉动 Scala 的流行，如同 rails 之于 ruby）
3. 社区，生态还比较小，Scala 风格的库还非常少

#### ● 与 java 差别

相对于 Java 而言，Scala 的代码更为精简（减低犯错），而且功能更为广泛（Scala 其实是 Scalable Language 的简称，意为可扩展的语言），许多 Scala 的特性和语法都是针对 Java 的不足和弱点来设计的。Scala 的特点是有许多函数式语言的特性（例如 ML, Miranda, Scheme, Haskell），譬如惰性求值，list comprehension, type inference, anonymous function, pattern matching 等等，同时也包含 Object-Oriented 的特性（OO 能与 FP 混合使用是 Scala 的亮点）。此外，许多相似于高级编程语言的语法也渗入其中（例如 Python），不仅提高了 Scala 代码的可读性，维护、修改起来也较为省时省力。

Scala 与 Java 语法上的明显差异有：

1. 不需要分号结尾
2. 类型定义开头需大写（与 Haskell 相同）
3. 函数定义需 def 开头（与 Python、Ruby 相同）
4. return 可以省略

#### 扩展学习：

适合 Scala 的应用场景很多，大数据处理是之一，今天的 Spark、Kafka 等也是让 Scala 火热的主要技术之一。在基于 Actor 的高并发场景，为 Java 带来了处理大量的并发应用带来简单的方案。对于有小规模应用 Akka 的场景，确实非常受益。不过，是否相比 Go、Erlang 等更合适，这就要针对不同的业务具体分析。应用更多的是事务类处理服务，也就是主要进行数据库操作的

微服务场景。在这方面，在电商等领域，自然就是使用 Scala 来解决这类问题，一开始仅仅是作为 Better Java，就是结合 Scala 的一些优势地方，比如丰富的集合处理能力、类型推演等，可以编写比 Java 更为简洁的代码，在这些场景中，使用一些 FP 的特性，也使用一些 Java 的模式。对于 Scala 中对于 FP 泛型支持和类型系统支持让它比 Java 更加安全，同时又具备灵活性和想象力。包括：可变量、mutable、非正常的执行流程（for, break, return）。通过下面几组测试：不容许使用 var，只容许使用 val 不容许使用 mutable，只容许使用 immutable 不容许使用 Java Bean，没有 getter/setter，只有 Case Class 不容许使用 while/break，只有顺序式代码 no null, but Option。会发现是比较有挑战 and 困难的。但是随着一个项目执行下来，发现做这样的改变其实并不困难。而且从此带来的收益是非常的明显的，代码量相比之前的模式有大幅度的简化和质量上的提升。

对于不同业务的开发，不同语言所表露出来的特性和优势对于开发人员的选择是非常重要的。代码的质量，以及高并发或者需要更加灵活性的处理中 Scala 都表现出了它无与伦比的优势所在。

# 《计算机新技术讲座》第四次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**知识图谱在公安行业的落地

**报告人信息：**

吕晓宝，中科曙光南京研究院研发工程师。吕博士于 2012 年和 2018 年分别获得南京大学理学学士和中国科学技术大学物理学博士学位。目前在中科曙光博士后工作站和东南大学博士后流动站从事知识图谱相关研究，并负责相关产品的研发和市场落地工作。

**企业信息：**

曙光信息产业股份有限公司是在中国科学院大力推动下，以国家“863”计划重大科研成果为基础组建的国家高新技术企业。公司直属中科院领导和管理，主要从事研究、开发、生产制造高性能计算机、通用服务器及存储产品，并围绕高端计算机提供软件开发、系统集成与技术服务。

**报告内容：**

知识图谱是人工智能的一个重要分支。本讲座主要介绍知识图谱的相关概念以及在行业落地过程中面临的挑战和机遇，并介绍在公安行业的具体落地方案。

## 二、收获和扩展学习

**报告收获：**

通过这次报告，让我们了解到了知识图谱的相关概念以及在行业落地过程中面临的挑战和机遇。知识图谱是认知人工智能领域的典型技术，其本质是一种语义网络，用来描述真实世界中存在的各种实体或概念，以及它们之间的关联关系。知识图谱技术是大数据环境下建设智慧警务的最佳选择：

a)知识图谱的数据组织结构符合人的思维模式，是人工智能的基础环境。

b)知识图谱的组织结构能够灵活的应对行业的数据种类变化，使得新的数据种类可以快速融合发挥作用。

c)知识图谱不仅存储数据，也存储领域知识，包含通用领域和行业领域内事物的全面属性、精细化分类、概念名称、特定表达，以及事物之间的基本规则、行业规则、领域规范等。

**扩展学习：**

知识图谱还可以在司法改革中发挥其作用，让机器通过深度学习以认知个案,是人工智能司法应用的前提与薄弱之处。当前,地方法院的实践尚未形成有效的知识积累方法。人工智能司法应用的前提是法律知识图谱的构建以及裁判规则的类型化与要素化。要件事实型民事裁判论与司法人工智能的生成规律具有内生契合性,可作为神经网络深度学习、分词设置、知识图谱设计的基础与前端理论。具体应用路径是依要件事实论不断进行层级解构,将案件认事用法解构为不同层级要素,并由法律专家分层级、分阶段标注,从而形成以要素标注的法律知识图谱大数据,以供机器学习。要件事实论之于人工智能司法应用具有独立性,人工智能难以代替法官。人工智能司法应用与民事诉讼制度具有相互促进关系,两者的深度融合将开拓中国民事诉讼的崭新发展阶段。



# 《计算机新技术讲座》第五次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**互联网下探索大数据的奥秘

**报告人信息：**

成钢，思特奇行业总监，高级项目管理师。20+年的 IT 行业管理经验，曾在多家知名 IT 企业担任管理职位，团队成员超百人，精通大型系统架构、云计算、项目管理等。

**企业信息：**

北京思特奇信息技术股份有限公司成立于 1995 年，是一家提供电信级软件和运营服务的上市公司。全国大数据排名 50 强、北京软件和信息服务业综合实力百强企业、CMMI5 级认证、ITSS 资质，双软认证、信息系统集成与服务一级资质、国内最顶级 IT 认证资质和国家级高新技术企业。是四大电信运营商战略合作伙伴，目前提供的产品及服务承载的客户量超过 10 亿，为中国民族软件龙头企业。

**报告内容：**

大互联网背景下，讲述大数据的应用场景，帮忙同学们更好的理解大数据理论如何应用到实际场景，讲座将从三个维度介绍互联网背景下大数据在应用，第一：大数据与通信领域的整合；第二：大数据运营助力智慧城市；第三：行业民生中的大数据。

## 二、收获和扩展学习

**报告收获：**

通过这次报告，让我们了解到了在互联网的背景下，大数据技术的支持下的一些行业的发展。大数据技术的体系庞大且复杂，基础的技术包含数据的采集、数据预处理、分布式存储、NoSQL 数据库、数据仓库、机器学习、并行计算、可视化等各种技术范畴和不同的技术层面。首先给出一个通用化的大数据处理框架，主要分为下面几个方面：数据采集与预处理、数据存储、数据清洗、数据查询分析和数据可视化。大数据的基本处理流程与传统数据处理流程并无太大差异，主要区别在于：由于大数据要处理大量、非结构化的数据，所以在各处理环节中都可以采用并行处理。目前，Hadoop、MapReduce 和 Spark 等分布式处理方式已经成为大数据处理各环节的通用处理方法。

**扩展学习：**

在大数据技术支撑下智慧城市的建立也得以实行。智慧城市基础设施层实现与城市运行相关的海量数据采集，并在社会系统中进行汇聚、整合和存储，社会系统对各行业、领域子系统的运行数据进行深度挖掘分析。随着城市信息化的发展，城市中每天都会产生海量的运行数据。而在大数据的时代，传统的数据分析方式已经不能满足获取数据的更大价值。社会系统分析处理城市各子系统运行产生的所有海量数据，而不再依赖数据样本分析；通过对全部海量数据挖掘分析，获取事物之间的关联关系；通过分析各行业领域的关联性获取新的洞察和价值，使城市管理更加科学化、智慧化。例如，智慧旅游子系统对旅游行业存储的全部海量数据进行分析，预测旅游高峰期旅客的不同需求和需要，并据此做出对公共交通的需求，将公共交通需求通过 SOS 传递至智慧的交通子系统。交通子系统根据公共交通需求实现对公共交通的实时调度。

城市各行业子系统通过智慧城市社会系统相互关联，海量的城市运行数据在此汇聚、整合、

共享，为利用大数据技术对海量数据挖掘分析奠定基础。目前，大数据的初步应用不仅显现在企业，也显现在政府的各个领域，例如医疗、教育、经济、人文等，它在时刻改变着城市的方方面面。

大数据的应用场景：

1. **医疗大数据，看病更高效。**借助于大数据平台我们可以收集不同病例和治疗方案，以及病人的基本特征，可以建立针对疾病特点的数据库。如果未来基因技术发展成熟，可以根据病人的基因序列特点进行分类，建立医疗行业的病人分类数据库。在医生诊断病人时可以参考病人的疾病特征、化验报告和检测报告，参考疾病数据库来快速帮助病人确诊，明确定位疾病。在制定治疗方案时，医生可以依据病人的基因特点，调取相似基因、年龄、人种、身体情况相同的有效治疗方案，制定出适合病人的治疗方案，帮助更多人及时进行治疗。同时这些数据也有利于医药行业开发出更加有效的药物和医疗器械。
2. **生物大数据，改良基因。**们所说的生物大数据技术主要是指大数据技术在基因分析上的应用，通过大数据平台人类可以将自身和生物体基因分析的结果进行记录和存储，利用建立基于大数据技术的基因数据库。大数据技术将会加速基因技术的研究，快速帮助科学家进行模型的建立和基因组合模拟计算。基因技术是人类未来战胜疾病的重要武器，借助于大数据技术的应用，人们将会加快自身基因和其它他生物的基因的研究进程。未来利用生物基因技术来改良农作物，利用基因技术来培养人类器官，利用基因技术来消灭害虫都即将实现。
3. **金融大数据，理财利器。**金融行业的大数据面临的往往是同样的问题，但是情况可能要好点，类似企业和个人的一些信用记录现在有全国性质的统一数据库能够拿到部分数据。但是对于单个银行来说，同样是无法拿到用户在其他银行的行为记录数据的，其二银行本身在做很多信贷风险分析的时候，确实需要大量数据做相关性分析，但是很多数据来源于政府各个职能部门，包括工商税务，质量监督，检察院法院等，这些数据短期仍然是无法拿到。还有就是企业或个人本事日常产生的各种行为数据更难拿到，那么对客户的风险性评估还是得借用原来的老方法而已。

# 《计算机新技术讲座》第六次报告

71116233 白丰硕

## 一、报告内容

**讲座主题：**容器技术的原理与应用

### 报告人信息：

鲍松，SAP 南京创新中心资深研究员。2003 年本科毕业于南京信息工程大学计算机科学与技术系专业，2006 年研究生毕业于南京航空航天大学计算机应用专业。2018 年加入 SAP，专注于新兴技术在智能企业领域中的应用，重点研究方向为微服务的架构与治理、区块链以及运维自动化技术。加入 SAP 之前，曾就担任摩托罗拉南京研发中心高级工程师、毕马威智能创新空间高级经理。

### 企业信息：

SAP 成立于 1972 年，总部位于德国沃尔多夫市，是全球最大的企业管理和协同化商务解决方案供应商，世界第三大的独立软件供应商，全球第二大云公司，在全球有 120 多个国家的超过 172,000 家用户正在运行 SAP 软件。财富 500 强 80% 以上的企业都正在从。SAP 的管理方案中获益。

### 报告内容：

容器技术以其高效、灵活和不可变更的特点成为当前各大云计算平台软件部署与自动化的最佳实践与事实上的业界标准，并将为企业的数字化平台提供敏捷、弹性和自动化的全新特性。容器技术推动着 DevOps 概念的变迁，已经成为了当今各大互联网公司基础架构部门的必然选择。本次讲座将结合操作系统的基本概念，深入浅出的介绍容器技术的基本原理及其容器技术在 SAP 创新中心的实践案例。

## 二、收获和扩展学习

### 报告收获：

通过这次报告，让我们了解到了容器技术基本原理和特点以及其凭借应用优势成为了当今各大互联网公司基础架构部门的必然选择。容器技术虚拟化技术已经成为一种被大家广泛认可的容器技术服务器资源共享方式，容器技术可以在按需构建容器技术操作系统实例的过程当中为系统管理员提供极大的灵活性。由于 hypervisor 虚拟化技术仍然存在一些性能和资源使用效率方面的问题，因此出现了一种称为容器技术（Container）的新型虚拟化技术来帮助解决这些问题。

Docker 就是虚拟化的一种轻量级替代技术。Docker 的容器技术不依赖任何语言、框架或系统，可以将 APP 编程一种标准化、可移植的、自管理的软件，并脱离服务器硬件在任何主流系统中开发、调试和运行。简单的说就是，在 Linux 系统上迅速创建一个容器(类似虚拟机)并在容器上部署和运行应用程序，并通过配置文件可以轻松实现应用程序的自动化安装、部署和升级，蜂场方便。

对于 Docker 的优势。1.Docker 在开发与运维的世界中具有极大的吸引力，因为它能保持跨环境的一致性。2.在过去的几年里，所有主流的云计算提供商，包括亚马逊 AWS 和谷歌的 GCP，都将 Docker 融入到他们的平台并增加了各自的支持。3.Docker 容器可以在不同的开发与产品发布生命周期中确保一致性，进而标准化生产环境。4.Docker 可以确保应用程序与资源是分隔开的。5.从安全角度来看，Docker 确保运行在容器中的应用程序和其他容器中的应用程序是完全分隔与隔离的，在通信流量和管理上赋予完全的控制权。

Docker 目前已经成为了非常主流的技术，已经在很多成熟公司的生产环境中使用，但是

Docker 的核心技术其实已经有很多年的历史了，Linux 命名空间、控制组和 UnionFS 三大技术支撑了目前 Docker 的实现，也是 Docker 能够出现的最重要原因。

## 扩展学习：

### Kubernetes

Kubernetes 是 Google 开源的容器集群管理系统。它构建在 Docker 之上，为容器化的应用提供资源调度、部署运行、服务发现、扩容缩容等一整套功能，本质上可以看做是基于容器技术的 Micro-PaaS 平台，是第三代 PaaS 技术的代表之作。Google 从 2004 年开始就已经开始在使用容器技术了，于 2006 年发布了 Cgroup，而且内部开发的强大的集群资源管理平台 Borg 和 Omega，这些都已经广泛用在 Google 的各个基础设施中，而 Kubernetes 得灵感来源于 Google 内部的 Borg 系统，更是吸收了包括 Omega 在内的容器管理器的经验和教训。

### Kubernetes 有着以下优秀的特性：

#### 1. 强大的容器编排能力

Kubernetes 可以说是同 Docker 一起发展起来的，深度集成了 Docker，天然适应容器的特点，设计出强大的容器编排能力，比如容器组合、标签选择和服务发现等，可以满足企业级的需求

#### 2. 轻量级

Kubernetes 遵循微服务架构理论，整个系统划分出各个功能独立的组件，组件之间边界清晰，部署简单，可以轻易地运行在各种系统和环境中。同时 Kubernetes 中的许多功能都实现了插件化，可以非常方便地进行扩展和替换。

#### 3. 开放开源

Kubernetes 顺应了开放开源的趋势，吸引了大批开发者和公司参与其中，协同工作，共同构建生态圈。同时，Kubernetes 同 Openstack、Docker 等开源社区积极合作、共同发展。

### Kubernetes 的核心概念

#### 1. Pod

Pod 是若干相关容器的组合，Pod 包含的容器运行在同一台宿主机上，这些容器使用相同的网络命名空间、IP 地址和端口，相互之间能通过 localhost 来发现和通信。这些容器还可以共享一块存储卷空间。在 Kubernetes 中创建、调度和管理的最小单元就是 Pod，而不是容器，Pod 通过更高层次的抽象，提供了更加灵活的部署和管理模式。

#### 2. Replication Controller

Replication Controller 用来控制管理 Pod 副本，它确保任何时候 Kubernetes 集群中有指定数量的 Pod 副本在运行。如果少于指定数量的 Pod 副本，它会启动新的 Pod 副本，反之它会杀死多余的副本以保证数量不变。而且，Replication Controller 是弹性伸缩、滚动升级的实现核心

#### 3. Service

Service 是真实应用服务的抽象，定义了 Pod 的逻辑集合和访问这个 Pod 集合的策略。Service 将代理 Pod 对外表现为一个单一的访问接口，外部不需要了解后端 Pod 是如何运行的，这给扩展和维护带来很多好处，提供了一套简化的服务代理和发现机制

#### 4. Label

Label 是用于区分 Pod、Service、Replication Controller 和 Key/Value 对，实际上，Kubernetes 中的任意 API 对象都可以通过 Label 进行标识。每个 API 对象可以有多个 Label，但是每个 Label 的 Key 只能对应一个 Value。Label 是 Service 和 Replication Controller 运行的基础，他们都通过 Label 来关联 Pod，相比于强绑定模型，这是一种非常好的松耦合关系。

#### 5. Node

Kubernetes 属于主从分布式集群架构，Kubernetes Node 运行并管理容器。Node 作为 Kubernetes 的操作单元，用来分配给 Pod 进行绑定，Pod 最终运行在 Node 上，Node 可以认为是 Pod 的宿主机