

| 大数据技术-第五章：YARN资源管理器
YARN基本架构及原理



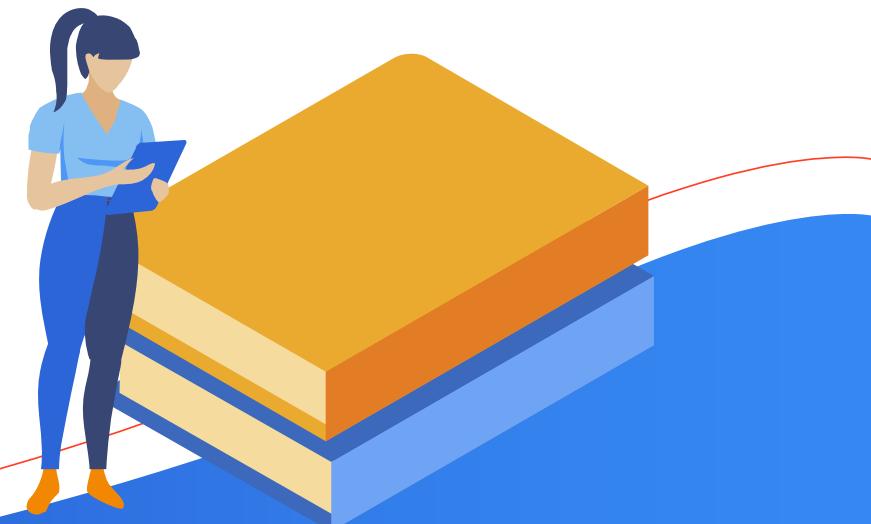
CONTENTS

01. YARN的架构

02 YARN的核心组件

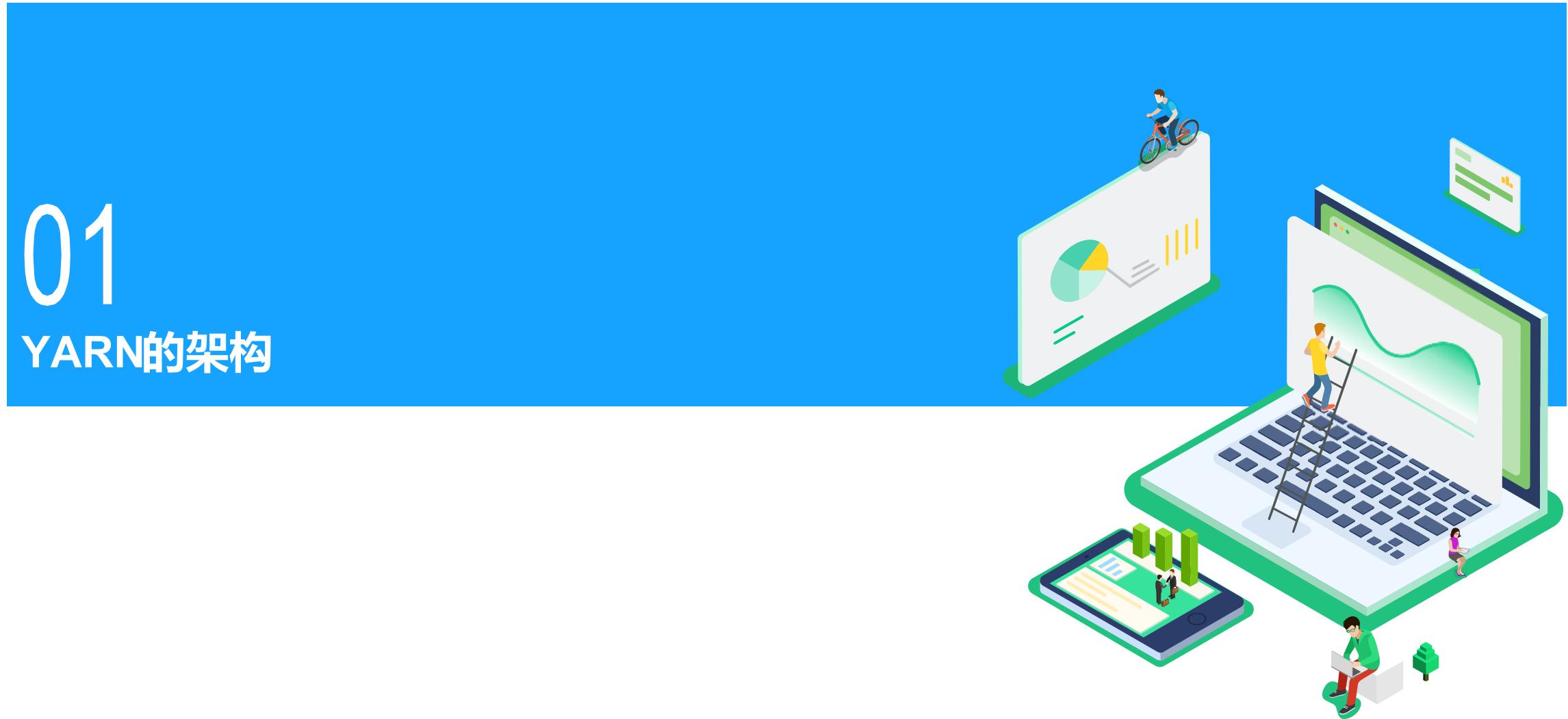
03 YARN的容错性

04 YARN的高可用

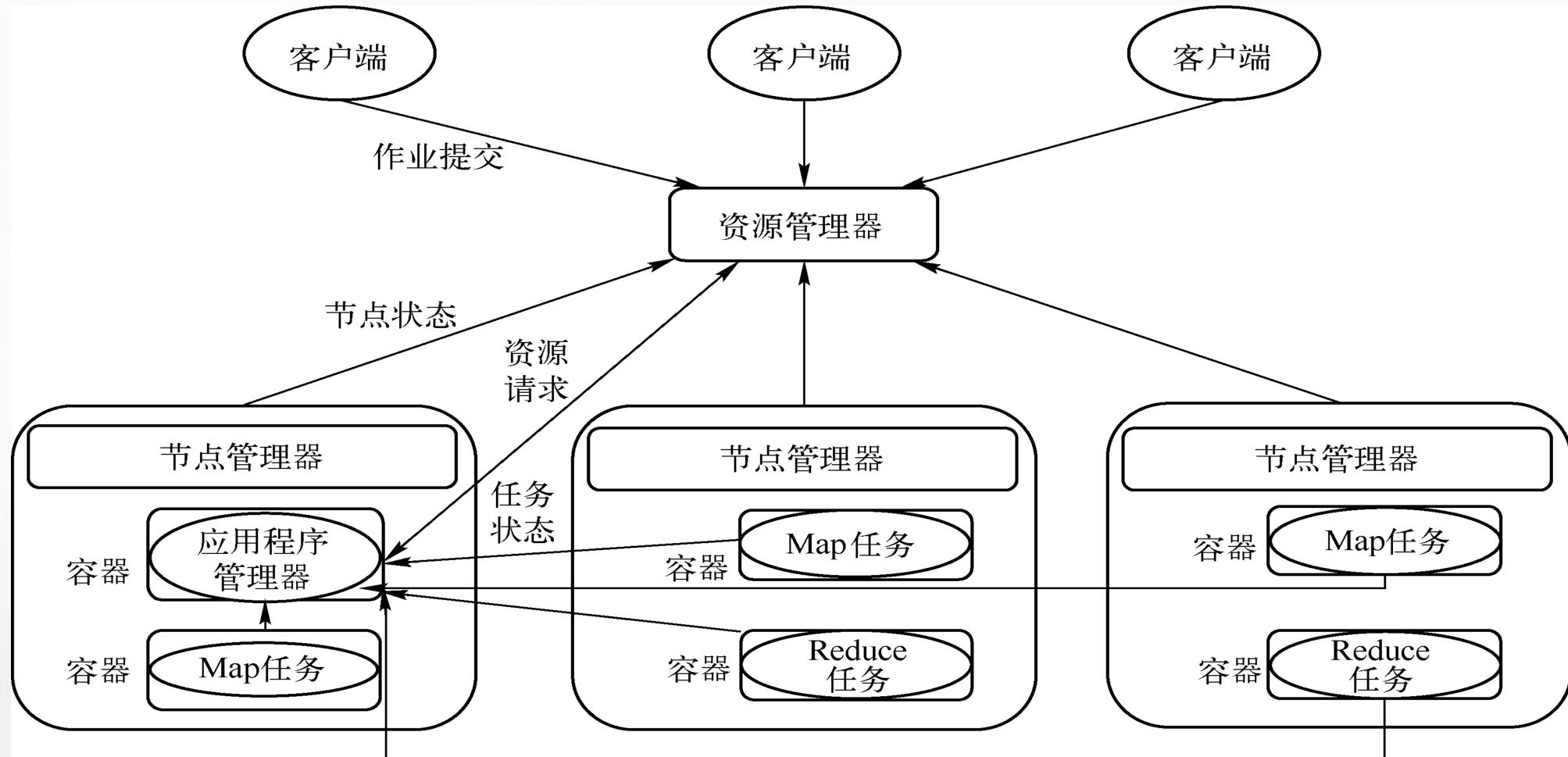


01

YARN的架构



YARN的架构

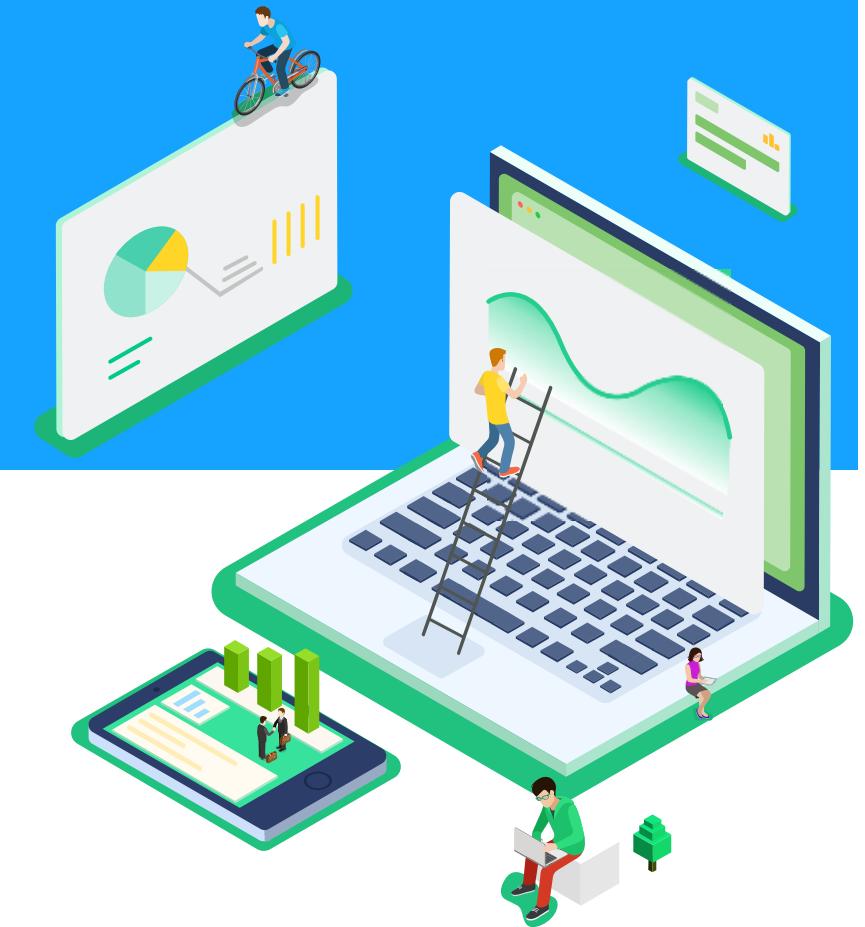


Yarn 采用传统的 master-slave 架构模式，其主要由 4 种组件组成：

- ResourceManager (RM) : 全局资源管理器，负责整个系统的资源管理和分配；
- ApplicationMaster (AM) :负责应用程序 (Application) 的管理；
- NodeManager (NM) :负责 slave 节点的资源管理和使用；
- Container (容器) :对任务运行环境的一个抽象。

02

YARN的核心组件



ResourceManager (RM)

ResourceManager 是一个全局的资源管理器，负责整个系统的资源管理和分配。它主要由两个组件组成：

(1) Scheduler: 资源调度器，主要功能和特点如下：

- 负责将资源分配给各种正在运行的应用程序，这些应用程序受到容量、队列等限制；
- Scheduler 是纯调度程序，不会监视或跟踪应用程序的状态；
- 由于应用程序故障或硬件故障，它不提供有关重新启动失败任务的保证；
- Scheduler 根据应用程序的资源需求来执行其调度功能，它是基于资源容器的抽象概念来实现的，容器 (Container) 内包含内存、CPU、磁盘、网络等因素；
- Scheduler 是一个可插拔的插件 (即可配置)，负责在各种队列、应用程序等之间对集群资源进行区分。当前支持的Scheduler类包括：FairScheduler、FifoScheduler、CapacityScheduler；

(2) Application Manager: 负责接受 job 提交请求，为应用程序分配第一个 Container 以运行 ApplicationMaster，并提供失败时重新启动运行着 ApplicationMaster 的 Container 的服务。

ApplicationMaster (AM)

当用户提交一个应用程序时，将启动一个被称为 ApplicationMaster 的轻量级进程的实例，用以协调应用程序内所有任务的执行。它的主要工作包括：

- (1) 向 ResourceManager 申请并以容器 (Container) 的形式提供计算资源；
- (2) 管理在容器内运行的任务：
 - 跟踪任务的状态并监视它们的执行；
 - 遇到失败时，重新启动失败的任务；
 - 推测性的运行缓慢的任务以及计算应用计数器的总值。

NodeManager (NM)

NodeManager 进程运行在集群中的节点上，是每个节点上的资源和任务管理器。它的主要功能包括：

- 接收 ResourceManager 的资源分配请求，并为应用程序分配具体的 Container；
- 定时地向 ResourceManager 汇报本节点上的资源使用情况和各个 Container 的运行状态，以确保整个集群平稳运行；
- 管理每个 Container 的生命周期；
- 管理每个节点上的日志；
- 接收并处理来自 ApplicationMaster 的 Container 启动/停止等请求。

Container (容器)

Container 是 Yarn 中的资源抽象，是执行具体应用的基本单位，它包含了某个 NodeManager 节点上的多维度资源，如内存、CPU、磁盘和网络 IO，当然目前仅支持内存和 CPU。任何一个 Job 或应用程序必须运行在一个或多个 Container 中，在 Yarn 中，ResourceManager 只负责告诉 ApplicationMaster 哪些 Containers 可以用，ApplicationMaster 需要自己去找 NodeManager 请求分配具体的 Container。

Container 和集群节点的关系是：一个节点会运行多个 Container，但一个 Container 不会跨节点。

03

YARN的容错性



- ResourceManager的容错性保障

ResourceManager存在单点故障，但是可以通过配置ResourceManager的HA（高可用），当主结点出现故障，将切换到备用结点继续对外提供服务。

- NodeManager的容错性保障

NodeManager失败之后，ResourceManager会将失败的任务告诉对应的ApplicationMaster，ApplicationMaster决定如何去处理失败任务。

- ApplicationMaster的容错性保障

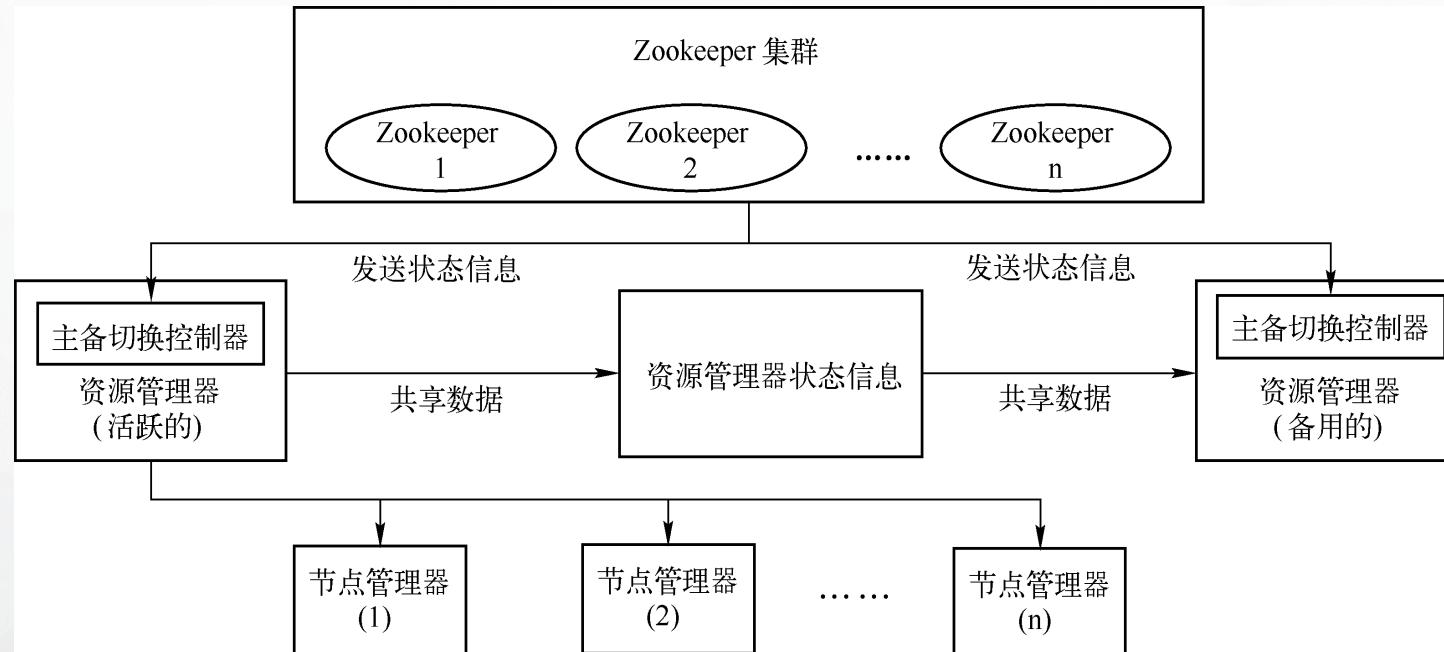
ApplicationMaster失败后，由ResourceManager负责重启。

04

YARN的高可用



HA (High Availability) 指高可用，YARN的HA主要指Resourcemanager的HA，因为Resourcemanager作为主结点存在单点故障，所以要通过HA的方式解决Resourcemanager单点故障的问题。



Turing AI 万维
图灵 | 大数据系列课程

大数据

BIG
DATA
智 / 能 / 科 / 技

放 / 眼 / 未 / 来

