

akka

讲师：陈博

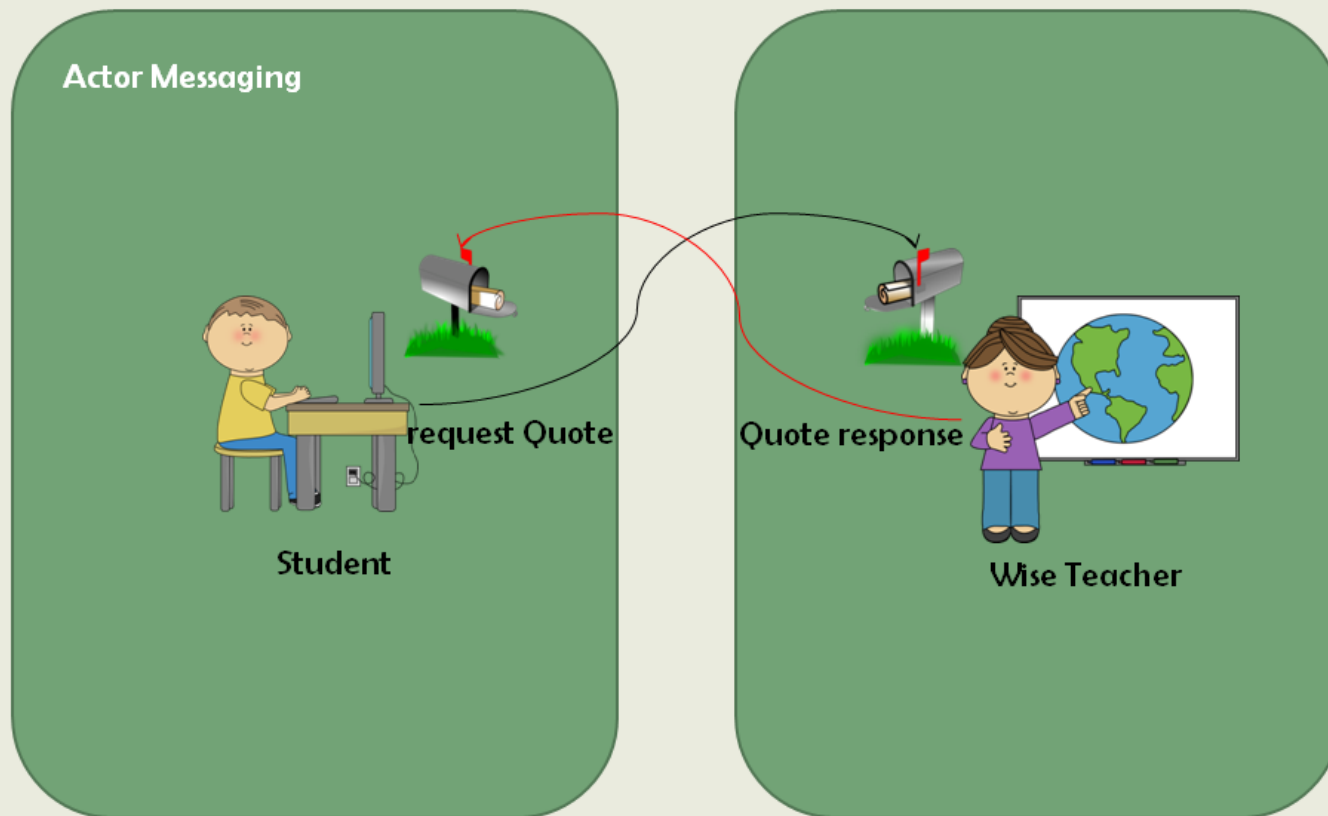


- **Actor Model**是用来编写并行计算或分布式系统的高层次抽象，让程序员不必为多线程模式下共享锁而烦恼，被用在Erlang 语言上，高可用性99.9999999 % 一年只有31 ms当机。

Actors将状态和行为封装在一个轻量的进程/线程中，但是不和其他Actors分享状态，每个Actors有自己的世界观，当需要和其他Actors交互时，通过发送事件和消息(看到Domain Events影子 呵呵)，发送是异步的，非堵塞的(fire-and-forget)，发送消息后不必等另外Actors回复，也不必暂停，每个Actors有自己的消息队列，进来的消息按先来后到排列，这就有很好的并发策略和可伸缩性，可以建立性能很好的事件驱动系统。



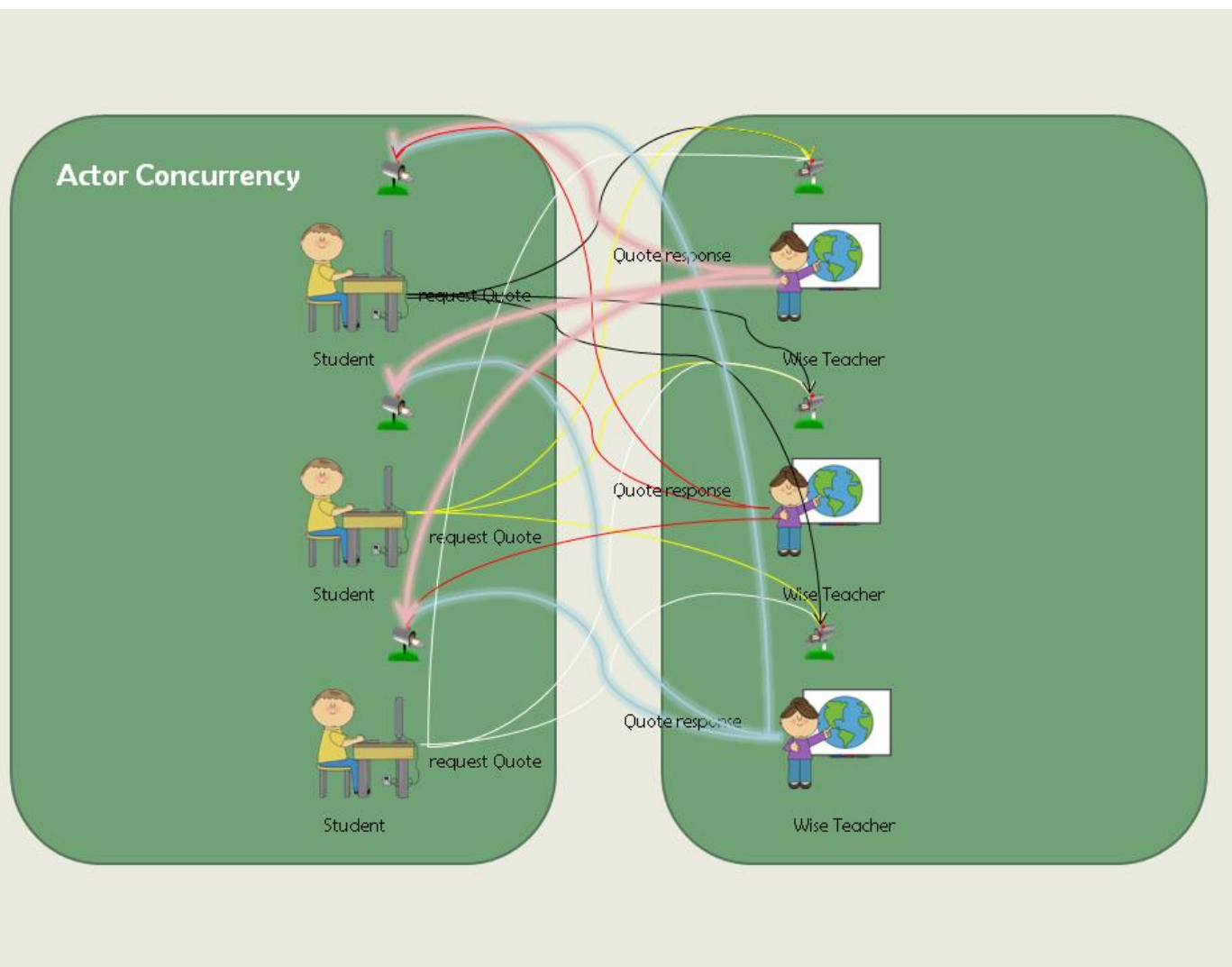
一、消息传递



- 假设有两个人：学生和聪明的老师。学生每天早上都会给老师发送邮件，而聪明的老师都会回复一句名言。这里需要解释：
 - 1、学生发送邮件。一旦发送成功，邮件不能再修改。这天然就具备了不可变性；
 - 2、老师会自己决定何时检查邮箱；
 - 3、老师还会回复一封邮件（也是不可变的）；
 - 4、学生会自己决定何时检查邮箱；
 - 5、学生不会一直等待回信（非阻塞的）这就可以总结出Actor模型的一个基本特征——消息传递



二、并发



- Akka有几个特性：
 - 1.更加简单的并发策略Simpler Concurrency，通过Actors STM & Transactors能够简化编写可靠的并行计算。
 - 2.EDA架构Event-driven Architecture，完美的异步事件驱动架构，不会堵塞。
 - 3.真正的可伸缩性，使用异步消息在多核以及多个节点之间扩展。
 - 4.容错性，重视失败。
 - 5.远程透明性



- **1.什么是Akka**
- Akka 是一个用 Scala 编写的库，用于简化编写容错的、高可伸缩性的 Java 和 Scala 的 Actor 模型应用。
- 官方网站 (<http://akka.io/>) 的介绍是：
- Akka is a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications on the JVM.
- Build powerful concurrent & distributed applications more easily.
- 翻译成中文就是：Akka是一个开发库和运行环境，可以用于构建高并发、分布式、可容错、事件驱动的基于JVM的应用。使构建高并发的分布式应用更加容易。



- 1) 易于构建并行和分布式应用 (Simple Concurrency & Distribution)
Akka在设计时采用了异步通讯和分布式架构，并对上层进行抽象，如Actors、Futures，STM等。
- 2) 可靠性 (Resilient by Design)
 - 系统具备自愈能力，在本地/远程都有监护。
- 3) 高性能 (High Performance)
 - 在单机中每秒可发送500000000个消息。内存占用小，1GB内存中可保存2500000个actors。
- 4) 弹性，无中心 (Elastic — Decentralized)
 - 自适应的负责均衡，路由，分区，配置
- 5) 可扩展 (Extensible)
 - 可以使用Akka 扩展包进行扩展。



什么场景下特别适合使用Akka?

- 我们看到Akka被成功运用在众多行业的众多大企业，从投资业到商业银行、从零售业到社交媒体、仿真、游戏和赌博、汽车和交通系统、数据分析等等等等。任何需要高吞吐率和低延迟的系统都是使用Akka的候选。
- Actor使你能够进行服务失败管理（监管者），负载管理（缓和策略、超时和隔离），水平和垂直方向上的可扩展性（增加cpu核数和/或增加更多的机器）管理。
- 下面的链接中有一些Akka用户关于他们如何使用Akka的描述：<http://stackoverflow.com/questions/4493001/good-use-case-for-akka>
- 所有以上这些都在这个Apache2许可的开源软件中。

- 以下是Akka被部署到生产环境中的领域
事务处理 (在线游戏, 金融/银行业, 贸易, 统计, 赌博, 社交媒体, 电信):垂直扩展, 水平扩展, 容错/高可用性

服务后端 (任何行业, 任何应用):提供REST, SOAP, Cometd, WebSockets 等服务 作为消息总线/集成层 垂直扩展, 水平扩展, 容错/高可用性

并发/并行 (任何应用):运行正确, 方便使用, 只需要将jar包添加到现有的JVM项目中 (使用Scala, java, Groovy或jruby)

仿真:主/从, 计算网格, MaReduce等等.

批处理 (任何行业):Camel集成来连接批处理数据源 Actor来分治地批处理工作负载

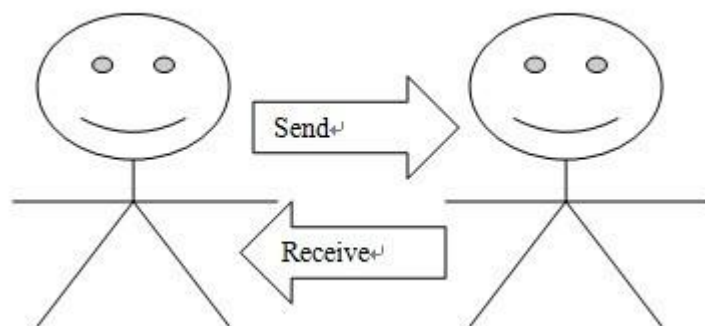
通信Hub (电信, Web媒体, 手机媒体):垂直扩展, 水平扩展, 容错/高可用性

游戏与赌博 (MOM, 在线游戏, 赌博):垂直扩展, 水平扩展, 容错/高可用性

商业智能/数据挖掘/通用数据处理:垂直扩展, 水平扩展, 容错/高可用性

复杂事件流处理:垂直扩展, 水平扩展, 容错/高可用性





- Actor模型并非什么新鲜事物，它由Carl Hewitt于上世纪70年代早期提出，目的是为了了解决分布式编程中一系列的编程问题。其特点如下：
- 系统中的所有事物都可以扮演一个Actor
- Actor之间完全独立
- 在收到消息时Actor所采取的所有动作都是并行的，在一个方法中的动作没有明确的顺序
- Actor由标识和当前行为描述
- Actor可能被分成原始（ primitive ）和非原始（ non primitive ）类别
- 很多开发语言都提供了原生的Actor模型。例如erlang, scala等



- Actor，可以看作是一个个独立的实体，他们之间是毫无关联的。但是，他们可以通过消息来通信。
- 一个Actor收到其他Actor的信息后，它可以根据需要作出各种相应。消息的类型可以是任意的，消息的内容也可以是任意的。这点有点像webservice了。只提供接口服务，你不必了解我是如何实现的。
-
- 一个Actor如何处理多个Actor的请求呢？它先建立一个消息队列，每次收到消息后，就放入队列，而它每次也从队列中取出消息体来处理。通常我们都使得这个过程是循环的。让Actor可以时刻处理发送来的消息。



- 并发的英文是 Concurrency，并行的英文是 Parallel，并发与并行是两个不同的概念。
-
- 并行是指在同一时刻，有多条指令在多个处理器上同时执行。
- 并发是指在同一时刻，只能有一条指令执行，但多个进程指令被快速轮换执行，使得在宏观上具有多个进程同时执行的效果。
- 有人做过比喻，要完成吃完三个馒头的任务，一个人可以这个馒头咬一口，那个馒头咬一口，这样交替进行，最后吃完三个馒头，这就是并发，因为在三个馒头上同时发生了吃的行为。
-
- 三个馒头如果分给 三个人吃，这样的任务完成形式叫并行。



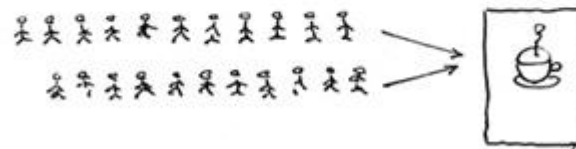
Concurrent and Parallel Programming

05 Apr 2013

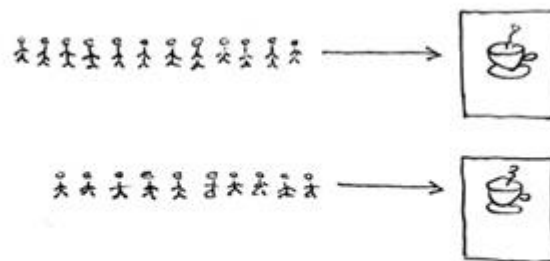
What's the difference between concurrency and parallelism?

Explain it to a five year old.

Concurrent = Two Queues One Coffee Machine



Parallel = Two Queues Two Coffee Machines



© Joe Armstrong 2013

Concurrent = Two queues and one coffee machine.

Parallel = Two queues and two coffee machines.



AKKA是调度模块化的，它由许多拥有不同特性的JAR组成。

- akka-actor – 经典角色、类型角色、IO角色等。
- akka-agent – 代理、整合了Scala的STM特性
- akka-camel – 整合Apache的Camel
- akka-cluster – 集群成员管理、弹性路由
- akka-kernel – AKKA微内核，运行着一个极简应用服务器
- akka-osgi – 在OSG容器里使用AKKA的基本bundle，包括akka-actor的类
- akka-osgi-aries – Aries——服务提供角色系统的蓝图
- akka-remote – 远程角色
- akka-slf4j – SLF4J Logger (事件总线监听器)
- akka-testkit – 测试角色系统的工具包Toolkit for testing Actor systems
- akka-zeromq – 整合ZeroMQ

