Atitit 并发技术的选项 attilax总结 艾龙 著

[1. 三大并发模型 1](#_Toc25692)

[2. 从可读性考虑 优先使用 并行工作者 多线程模式，不要使用异步流水线模式 2](#_Toc22891)

[2.1. 多线程模式方便全局异常捕获。 异步不能全局捕获，只能每个毁掉里面处理 2](#_Toc7078)

# 三大并发模型

[1. 并发系统可以使用不同的并发模型去实现。 1](#_Toc6578)

[2. 并行工作者 并行工作者模型。进来的任务分配给不同的工作者 银行模式 2](#_Toc21975)

[2.1. 并行工作者的优势 并行工作者并发模型的优势在于理解起来比较简单。为了增加应用的并行计算，你只是需要添加更多的工作者就可以了。 2](#_Toc314)

[2.2. 并行工作者的劣势 3](#_Toc6095)

[2.3. 委派器将任务分配给Worker，Worker单独完成任务，java 7 中 java.util.concurrent　包中好多工具都是基于此模型实现的。明显存在的存在的问题是工作器之间需要共享资源。 5](#_Toc8516)

[3. 流水线（Assembly Line） 事件驱动系统 医院模式 5](#_Toc5884)

[3.1. 流水线的优势 7](#_Toc21116)

[3.1.1. 1：Worker之间不需要状态共享 7](#_Toc4404)

[3.2. 2：有一定的顺序，有利于模型的回复 7](#_Toc27604)

[3.2.1. 3：符合计算机硬件的特点 7](#_Toc6220)

[3.3. 流水线的劣势 。回调地狱 8](#_Toc7118)

[3.4. Nodejs采用的模型 9](#_Toc31926)

[3.5. Actor 和Channel 9](#_Toc17703)

[4. 功能并行性（Functional Parallelism）三：函数式并发模型 9](#_Toc25095)

# 并发的实现

## 并行工作者模型实现 多进程 多线程 携程 定时器

## 异步流水线模型实现 事件驱动

# 从可读性考虑 优先使用 并行工作者 多线程模式，不要使用异步流水线模式

## 多线程模式方便全局异常捕获。 异步不能全局捕获，只能每个毁掉里面处理