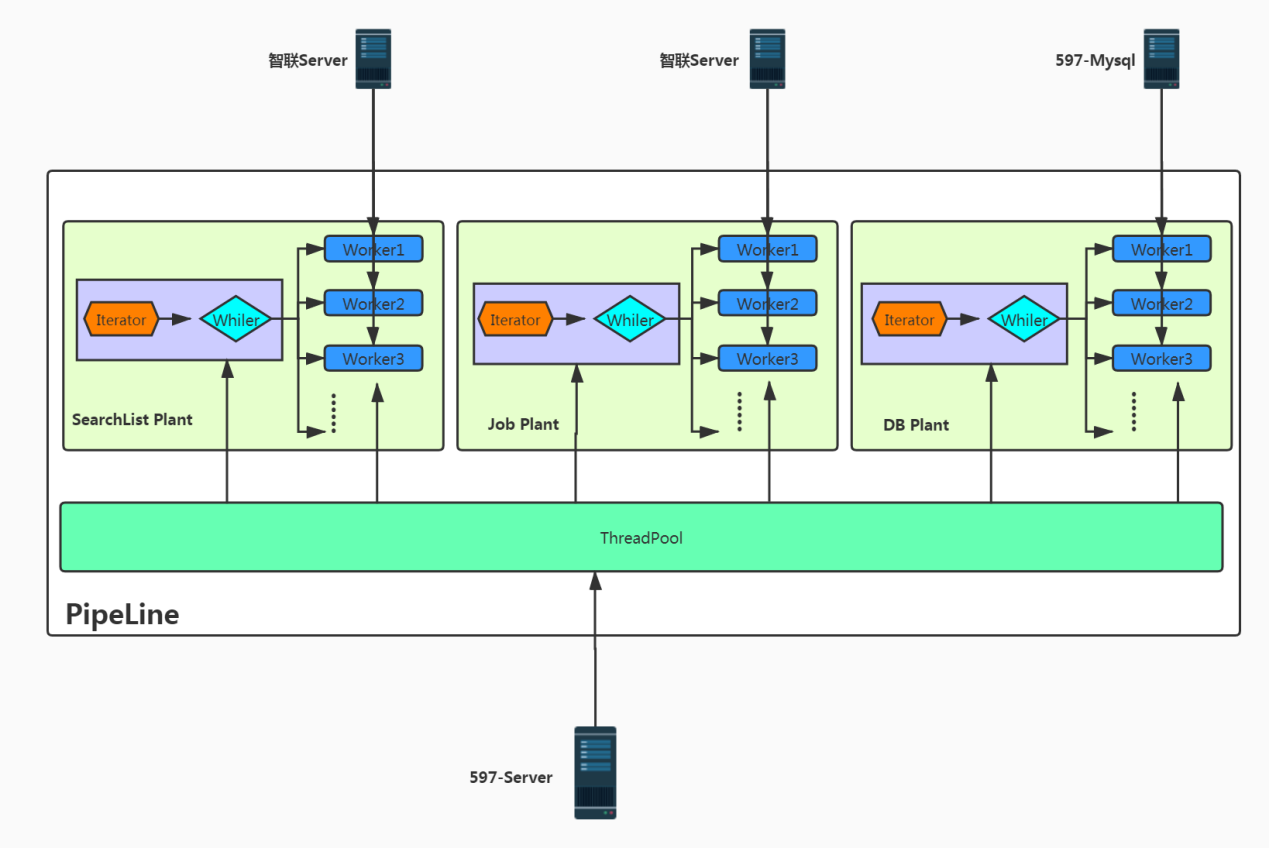
**一、整体架构**



**解释**

1. Pipeline即生产线，负责协调多个车间之间具体的关系。

2. Plant即车间，执行每项任务的具体单元。

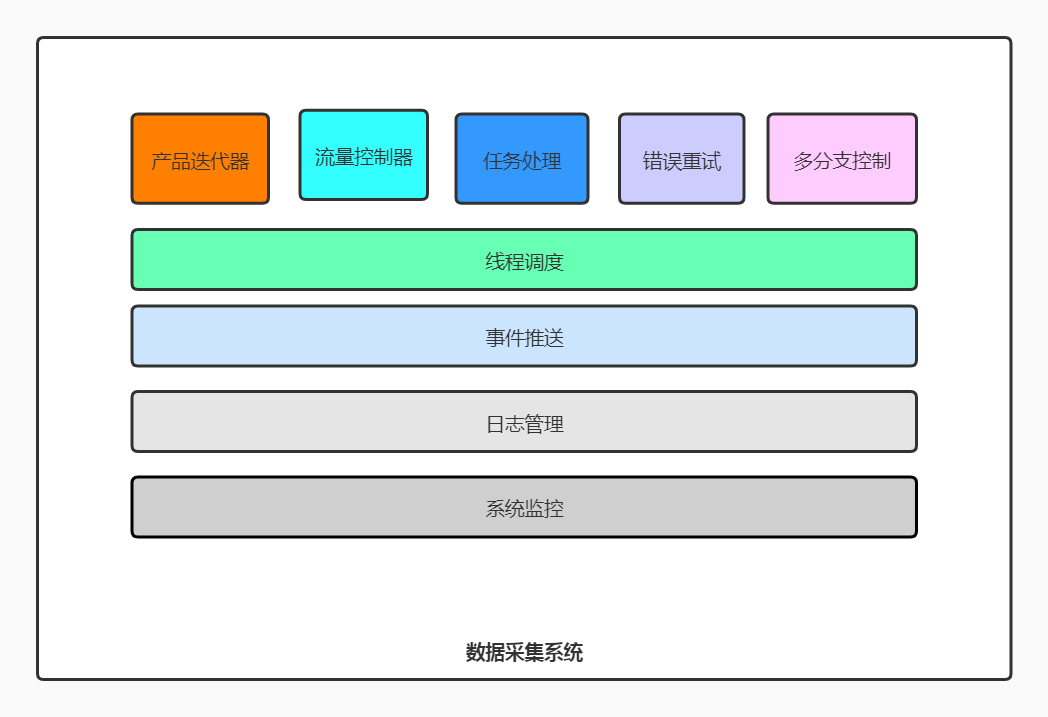
3. Iterator即车间产物，实际上是一个迭代器。

4. Whiler即循环器，主要负责循环分派任务。

5. worker即工作任务，是一些具体任务，包含其所需上下文封装。

6. ThreadPool即线程池，主要负责给整个系统提供工作线程，实现CPU资源可调度和共享。

**二、功能规划**



**解释**

1. 产品迭代器，由生产车间生产又由消费车间消费。
2. 流量控制器，负责控制消费车间在消费产品时的速度，即qps控制。
3. 任务处理，即具体任务处理，可以是Http请求、数据保存、数据清洗等.
4. 错误重试，控制某些情况下，具体任务重试的时间。
5. 多分支控制，当生产车间与消费车间是多对多关系，实现由具体的Pipeline进行实现。
6. 线程调度，在系统各个环节中的线程都交由统一的线程池管理，实现CPU资源利用率提升。
7. 事件推送，通过事件总线实现，实现相关功能分离与通信。
8. 日志管理，统一日志管理，不同功能日志不同输出方式。
9. 系统监控，监控系统内存、线程等资源实时使用情况。

**二、技术要点**

1.流量控制器：

流控器是一个公共的抽象，具体实现的流速可固定也可实时调整。常用的流控算法有：令牌桶、漏桶、滑动窗口。 我使用令牌桶，虽然理论上会有瞬间大并发问题，我通过以下2点进行规避处理，

（1）、流控器计时器的精度提高到ns级别，由于实际中系统的处理速度根本达不到纳秒，所以不会产生上述问题。

（2）、每类型的多任务调度采用Reactor模型，有统一一个mainWorker线程进行任务分配，所以流控器相对是单线程调用，不会产生并发问题，更不会产生瞬间大并发问题。

2.可停止迭代器：

作为产品生产和消费的载体而存在，使得各模块的关注点分离。是我自认为最为出彩的核心设计之一，与常规的java迭代器不同，它的可停止包括两个含义：停止退出和阻塞暂停：

1. 、系统中并不使用hasNext()进行元素个数判断，而直接调用next()获取下一个元素，当迭代器需要停止时会抛出StopException，使得mainWorker线程退出工作。
2. 、而在迭代器具体实现里next()可通过自己的方式使得调用next()的mainWorker线程阻塞。由此可以暂停当前类型任务的调度。

3.线程调度

线程调度主要通过一个全局线程池进行管理，可根据服务器资源情况，调整线程池的核心线程数、单个线程存活时长、最多线程数。系统各个模块的线程使用统一通过这个线程池 。

1. 事件推送

系统各个模块通过一个全局事件总线实现事件注册和触发，目前实现是异步推送，而异步的线程的使用又是通过全局线程池。主要为了实现系统各个模块的依赖分离、关注点分离，增加可扩展性。

1. 日志管理

系统各个模块通过全局日志进行管理，并且划分不同领域进行管理和输出。并且通过SpringAOP对共性的部分拦截记录。

1. 系统监控(暂时未实现)
2. Sping，通过Spring来实现IOC、AOP及相关参数化配置
3. Carteins，是一个算法，由于条件多维，时间和空间不能兼容。一次性生成所有条件的笛卡尔积并缓存会耗费内存，而实时迭代遍历又耗费cpu，中和两者最后决定生产所有条件数组的索引的笛卡尔积。 此处必须注意索引的存储使用String以减少内存的开销。

**三、设计思想及模式**

1.算法：限流算法、笛卡尔积

2.思想：领域驱动、IOC、AOP、Reactor

3.模式：单例模式、延迟加载、装饰器模式、观察者模式、迭代器模式、模板模式、代理模式、解释器模式、责任链模式

**四、设计中难点和问题：**

1.基于笛卡尔积算法的多条件迭代器如何兼顾时间和空间

2.mainWorker中如何实现流量和迭代的双重控制、重试

3.如何将组合Pipeline以充分利用采集源的qps限制

4.整个流程的模型设计和抽象，如何保证抽象性以达到较高的可扩展性。

5.如何突破采集源单IP下qps限制，可使用IP代理或肉鸡。有考察西刺等免费代理Ip的可用性很差，如果有需要就只能自己建立http代理服务器或付费购买代理ip。

**五、性能：**

1.智联职位Pipeline采用串行连续采集方式，qps上限为4r/s,实际测试为3.68r/s。如采用串行间断采集方式，qps上限可为5r/s,具体还未进行实际编码和实现。

2.单页列表解析器解析时长为1~2ms

3.条件4\*4的笛卡尔积单次迭代时长在1ms内，即微秒级别。

4.单页面Http请求时长在100~120ms。

5.智联职位Pipeline串行连续采集方式，jstat -jutil 查看YGC频率基本稳定，FullGC频率较少，GC工作较为健康。

6.本系统为IO密集型应用，所以根据目标采集源qps限制和运行服务器环境限制，线程池核心线程数尽可能设置大。

附注：线程池运行过程

