

设计方案

题目

假设我们在全中国 33 个省份各有两台主机(有外网ip), 每台机器上都可以通过 ip 直接下载 pingcap.tar.gz 这个包有 2g (如<http://192.168.1.1/pingcap.tar.gz>)。设计一个系统, 让每个地区的人能使用最快的速度下载到这个包。

要求:

- 下载速度要尽可能的快
- 要考虑到容错

提示:

- 注意代码可读性, 添加必要的注释
- 注意代码风格与规范, 添加必要的单元测试和文档
- 注意异常处理, 尝试优化性能

问题分析

- 33个省市共66台主机, 由于地域不同, 从一个地方访问各个省市的速度不一, **在能够(客户端限制并发度)、需要访问的机器数目有限时(最后的任务数不足66)**, 通过广度遍历省市的相邻关系, 确定优先级, 维护一个空闲主机访问优先队列, 依次请求不同的主机;
- 每台机器都能够通过ip直接下载资源, 并且支持断点续传, 所以66台主机都存在下载某一段资源的请求时最快;
- 应该隐含有条件每台主机有一定的下载速率限制, 或者ip限制, 不能多线程连接下载, 否则多线程连接一台最近、或者响应速度最快的主机分片下载最快。
- 由于机器的实际服务速度不一, 不能按主机个数均分所有的资源范围, 避免其他主机早已经完成响应而需要等待最慢的主机响应完成, 所以将任务按一定大小的range分配给各个工作线程, 工作线程向最近的空闲机器发往该range数据, 完成后请求下一个任务range;
- 需要考虑机器可能繁忙, 宕机、下载失败等情况, 所以维护一个redo任务队列, 分配任务时, 优先从redo任务队列中去取任务。

总体设计

如下图所示, 为设计的系统的工作流程示意图。

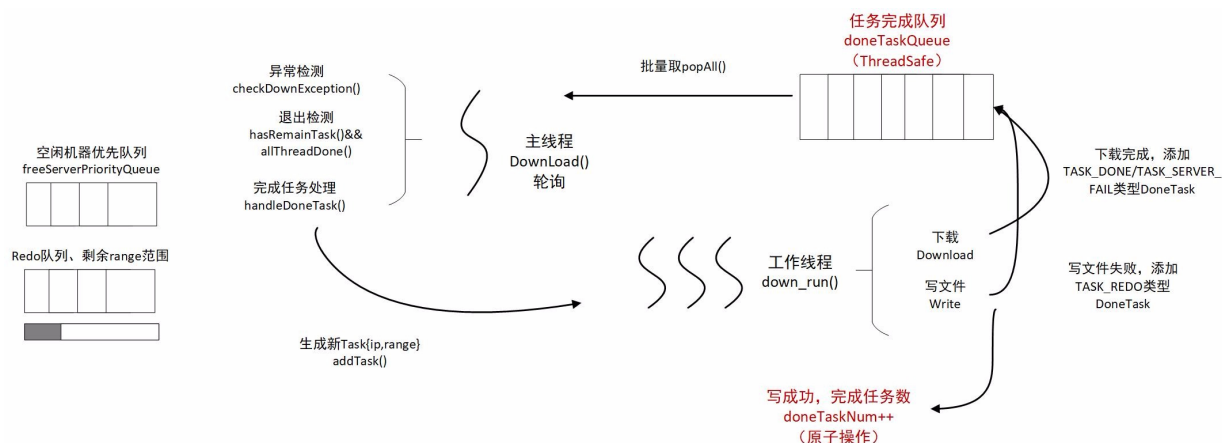


图1.系统工作流程示意图

设计的系统本质上是一个**单生产者，多消费者模型**，主线程轮询监听完成任务队列 `doneTaskQueue` 更新机器空闲状态和任务状态，当存在空闲机器和剩余任务时，生成新的任务交给工作线程。工作线程负责根据ip请求某一段范围range内的资源，工作线程会尝试重用已有未关闭的httpClient连接，减少建立tcp连接的时间，然后保存到本地做为临时文件，等待所有任务完成后合并，工作线程做了一个简单的优化是当下载完成后即写done类型事件任务完成队列，表示下载完成，当写文件失败后再写一个redo类型事件，能够加快下载速度，避免等待写磁盘时间，写文件成功后原子更新doneTaskNum，该值为实际成功的任务数。

主线程循环获取完成队列的完成时间，根据完成的状态，更新服务器状态，是否下线，是否空闲，以及是否需要加到redo队列，再做。因为**认为主要的时间在消费者下载时间**，而生成任务时间，异常处理时间较少，认为所以不需要多个生产者，这样也可以避免对各个状态对象频繁的加锁，如果消费者的下载速度过快，任务队列很快满，可调整每一次的请求range大小，也应能工作良好。

异常检查是，检查是否所有的服务器发现都不可到达，所以需要退出，以及是否下载超时。退出检测是正常的完成了所有的任务后可退出监听，然后合并临时文件。

优先队列的作用，各省市机器根据与客户端的地理位置决定请求优先顺序，只在开始和结束阶段对性能有一定的优化，在最后任务数不足机器数时，优先发给空闲的优先级更高（访问更近的机器）。

整个系统中需要加锁的地方在任务完成队列，本来想使用无锁队列，但是查找资料发现，并不适合多生产者、单消费者模型，对于任务完成队列而言是多个生产者push，而单个消费者pop。

可能的改进方法

可能存在一些可改进的地方：

- 优先级调整，由于省市大小不一，网络通信情况不一，实际的优先级可能并不是遍历的顺序，可以根据完成一定量后，计算平均下载时间，重新调整各个机器的优先级
- 任务分发方法，当前是通过切分成很小片，然后再获取下一次的任务大小，对服务器

缓存不够友好，而且与服务器的交互存在少量间隔，需要等待主线程分配新的任务。固定一个工作线程池，可以尝试先按机器数均等切分任务范围，但是当工作线程处理完初始分配的任务结束后，可以**主动从剩余未完成的工作线程的任务队列中获取新的任务（比如1/2剩余range）**，继续工作。

- 完成队列，使用数组实现的无锁队列对多生产者、单消费者不友好，但是Disruptor这种设计框架可以有较好的性能。

程序说明

- src

源代码目录

- main.cpp：程序入口
- common.h/cpp：预定义的通用的内容，错误码，函数等
- utility.h/cpp：初始化优先队列的处理函数，线程处理函数等
- http_client.h：定义的发起http请求并接收数据的客户端HttpClient
- province_server_node.h：主机节点，省市服务器节点（包含2台主机节点），省市服务器状态的结构定义
- task.h：任务请求范围TaskRequetRange，任务工作者TaskWorker结构的定义
- thread_safe_queue.h：定义的一个线程安全队列
- down_loader.h/cpp：**核心类**，完成下载请求的调度，与各种资源状态的管理，入口函数是DownLoad()。
- Makefile:makefile文件

- test

单元测试

`XX_test.cpp` 对应各个 `XX.h` 和 `XX.cpp` 文件的单元测试,本项目采用google gtest,libgtest.a为gtest的静态链接库。

- include

gtest的头文件