会议纪要

时间：2009年11月26日星期四

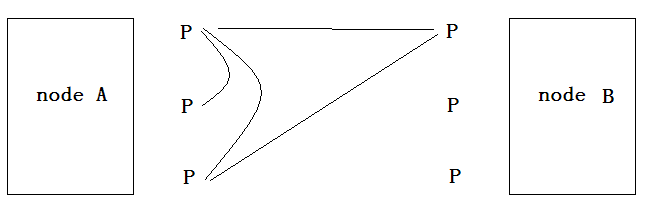
地点：

与会：许式伟、于羲鹤、莫华枫、段雪涛

主题：Erlang三种模型的讨论

**会议内容：**

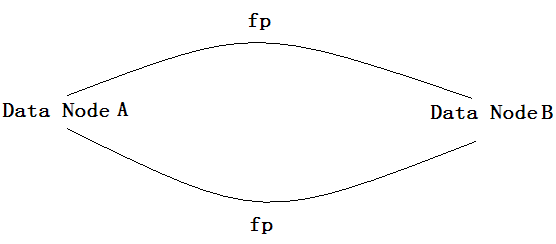
第一模型



这种模型的优点：node中的每一个process都可以作为work和server，由于process具有Erlang的优点，因此server是轻量级的。模型简洁。

实现方法：使用cerl1.0

缺点：（1）会产生死锁问题。发生死锁的场景如下：



A 向 B 发送一个同步消息，等待 B 回复；而与此同时 B 向 A 也发送了一个同步消息，也在等待状态。这时 A、B 两个进程都不能完成自己当前的任务，形成“死锁”。当然由于网络通讯的特殊性，最终 A 和 B 这种死锁是以超时表现出来。这是导致更换模型的直接原因。

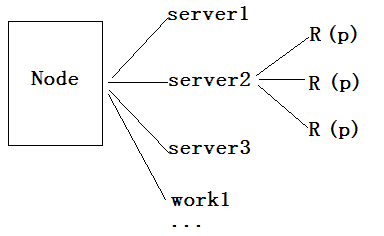
（2）performance，可以通过修改底层库来提高性能。

由于这个模型的缺点的存在导致了第二模型的出现。

第二模型

针对第一模型遇到的问题，设计出了第二模型。与第一种模型相比，这种模型由于没有recv原语，server的实现通过callback和channel来实现。此外，server在没有收到request的时候可以认为只是一个数据结构，在处理request的时候才产生一个process。每一个worker process不需要channel。

Node中可以有多个server，但是只有一个socket。Node向不同的server发送消息是通过channel id来实现。可以认为server = callback + channel id。

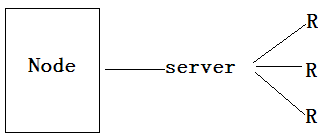


优点：实现了一个异步模型的方案，解决了第一个模型出现的死锁问题。

缺点：第二个模型带来了新的问题，设计复杂。同时需要解决多个Request的重入问题。

实现：使用cerl2.0

第三模型



第三模型是一个传统的服务器模型。每一个node对应一个server来处理request，不提供worker。这种模型的缺点是不能建立轻量级的server。和第二个模型一样，需要解决多个Request的重入问题。

实现：async + sdl简单封装即可。

**会议决议：**

我们将尽快实现第三种模型，推进项目进度。

对boost,asio,libevent做性能评估。