Scalable Commutativity Rule

Monday, March 21, 2016 10:36 PM

随着芯片产业的发展，由于高频的处理器的功耗过大，因此现有的芯片厂商都倾向于生产多核芯片，现有的服务器支持级联模式，32核乃至72核的服务器已不罕见。但是对于传统的操作系统来说，对多核的支持并不完美。如何提高多核的利用率，是一个需要考虑的问题。

这篇文章考察了interface(如system call)对软件扩展性的影响，分析得出一个基本的规律：一个接口的操作相互关联，他就可以被实现为多核可拓展的。为了帮助实现这个规律，推出了一个新的工具名为Commuter。并且使用这个工具验证了18个Posix系统调用。而后基于xv6实现了原型操作系统sv6，与Linux进行比较。

对于多核拓展性而言，经常被考虑为实现问题，而不是接口问题。但是如果考虑一个共享内存的多处理器系统，如果操作之间不需要读取其他核上的数据，则这个操作就是无冲突的，也即是可拓展的。如果把接口和实现良好的分离，那接口的设计中可能就有一些限制导致无法实现很好的隔离。

作者提出了一个新形式的叫做SIM的交换。该交换是基于状态的和接口的，也是一体的，即（SIM，state-dependent, interface-based and monotonic。当一个操作基于一个特定的系统状态和参数时，存在一个无冲突的实现。基于这个规律，可以指导一个新的方式去设计一个可拓展的软件。

对于这个直观性的规律，要得出一个准确的描述以及证明其正确性。

为了方便证明，先提取一系列的抽象：

1. Actions

   Actions通常是一个请求或者回应，比如一个带着参数的系统调用或者其返回的结果。请求和回应之间一一对应。所以actions包含以下要素：

   a. 操作集合
   b. 操作参数
   c. 相关线程
   d. 识别号

   对一系列的actions的集合称为history

   $H = A\ B\ C\ A\ C\ B\ D\ D\ E\ E\ F\ G\ H\ F\ H\ G$

2. Commutativity

   SI-commutes:

   只有X||Y'||Z是一个良好的历史时，X||Y||Z也是良好记录

   其中Y'意味着对Y中的任意一个线程，其Action顺序一致。

   然而SI-Commutativity不是monontic的。

   SIM：

   在历史H=X||Y中，对Y的任意前缀P，P在X||P中是SI-commutativity的