尊敬的发明人，您好：

我公司为您代理的以下专利申请的撰写工作已完成，为了保证申请人的权利，需要您的确认和签字。

发明名称：一种基于内存拥塞感知的多线程静态映射方法

申请类型：发明

申请人为：西安交通大学

发明人为：张兴军、张钰鑫

第一发明人身份证号：610113196908071633

经发明人代表确认代理人撰写的该项专利申请文件的技术方案描述清楚、完整、准确、已充分公开且能够实现，技术词汇用词准确，其创新点已全部记载在权利要求书中，其中所涉及的软件及程序均为现有技术。发明人同意此申请文件向国家知识产权局递交。

　　　　　　　　　　　申请人代表签字：

　　　　　　　　　　　　　　　年　月　　日

1.一种基于内存拥塞感知的多线程静态映射方法，其特征在于，在NUMA架构的多核处理器节点上，预运行应用程序追踪检测应用程序访存行为，之后根据追踪到的访存特征将线程绑定至计算核上，线程映射完毕。

2.根据权利要求1的方法，

其中通过检测应用程序访存行为并将线程绑定至计算核，实现线程静态映射过程。其特征在于，包括以下步骤：

步骤1，多线程程序在运行环境下，从时间与空间两个维度检测并统计线程间通信情况。在多线程并行程序运行过程中，利用本发明中引入的通信检测机制，统计不同线程ID之间进行基于共享内存模式的通信。对每次通信进行记录，统计为线程间通信时序表以及通信量矩阵，其中通信时序表包含3列分别记录通信的时间戳、进行通信的线程对其线程号i、j。通信量矩阵表示为A，n表示程序运行的线程总数，矩阵元素a(i,j) (i)表示线程i和线程j的通信量数值，本步骤的主要目标是收集并统计线程间的通信信息。

步骤2，计算线程分组。从步骤1中统计得到的通信量矩阵A，基于贪心策略进行线程两两配对，首先从A中随机选择一个线程i（初始可选择0号线程），之后搜索矩阵A找到线程j，满足a(i,j) = max(a(i,k)) (k,k≠0)，重复上述步骤直至将n个线程分为n/2组。将生成的所有线程对结合从步骤1中统计的通信时序表，得到所有线程对的通信时间戳信息，针对所有线程对的通信时间戳信息利用KMeans聚类，然后根据聚类结果选择线程对总数最多那一类提取出所有不同的线程对，将每个线程对的两个线程映射到同一节点相邻计算核心上，不同线程对分散映射在不同节点。最后得到线程与核的映射关系map={thread\_id:core\_id}。

步骤3，迁移线程。调用内核函数或设置系统环境变量为每个线程依次设置其cpu亲和度，将程序中的各线程依据分组计算的结果，迁移到对应的计算核心上。至此，整个映射过程完成，之后可用映射后的线程放置位置运行应用程序测试其加速效果。

3.根据权利要求2所述的映射执行步骤，其中检测并统计线程间通信包括：

从时间维度检测线程间的通信时序，通过基于Intel开发的Pin工具进行。Pin是一种动态二进制插桩框架，可以用于创建基于动态程序分析工具，支持x86-64指令架构。具体使用开源的Numalize检测工具，Numalize基于Pin开发以缓存行粒度追踪应用程序的所有访存行为,当不同线程访问同一缓存行时产生一次通信，这里对Numalize稍作改动以统计通信时序信息。定义一个包含3列的二维数组对每次通信进行记录，每条记录包括<timestamp,i,j>。以及

从空间维度检测线程间的通信量，同样使用Numalize。每次检测到的通信记录提取其通信线程对(i,j)，在要求2所述的矩阵A中对应位置更新，由于A为对称矩阵A[i][j]应该与A[j][i]相等，故A[i][j]++，A[j][i]++。

以此记录，随时更新，待程序运行完毕即可得到通信时序表及通信矩阵。

4.根据权利要求3所述的基于Numalize的通信检测方案，

其中，Numalize定义了一个hash表，对内存物理地址空间进行按照缓存行粒度进行划分，索引为划分后的缓存行地址，hash表项为一个队列，记录了在该缓存行上进行访存的不同的线程ID。 规定队列大小为4，以保证统计的精确度。每次当队列中记录入一个新的线程，将队首元素出队，然后计该新线程与仍在队列中的其他线程之间产生一次通信，同时使用rdtsc系统函数读取发生通信的时间戳，将上述信息在要求3中所述的通信时序表与通信矩阵中更新记录。

5.根据权利要求2所述的映射执行步骤，

其中，利用贪心策略将所有线程两两配对并通过通信时序表得到所有线程对的通信时间戳信息后，需要用KMeans算法对上述时间戳信息进行聚类。这里具体使用带权重的KMeans聚类算法，目标函数为 ，其中k表示类别个数，Ci 表示类别i ,t表示当前时间戳，Acommt表示t时间戳下的通信量，μi表示类别i所有时间戳的平均值。KMeans通过最小化上述目标函数将时间轴上接近且通信量接近的时间戳聚为一类。

6.根据权利要求1的方法和2所述的映射执行步骤，

迁移线程时，使用hwloc库函数hwloc\_set\_cpubind()或numactl 设置环境变量将线程依次绑定至核。具体为，遍历要求2得到的线程与核的映射关系map，对map内的所有键值对{thread\_id:core\_id},执行hwloc\_set\_cpubind 函数：hwloc\_set\_cpubind()，线程绑定完毕。

一种基于内存拥塞感知的多线程静态映射方法

**技术领域**

本发明总体上属于计算机体系结构和高性能计算领域，更具体地，涉及多线程并行环境下基于共享内存通信的静态线程映射方法。

**背景技术**

在以OpenMP为代表的多线程并行的编程模型中，应用程序的线程之间通过共享内存空间交换和共享数据，我们称这一过程为隐式通信模式。在并行计算机的计算节点内部，多线程并行程序正是通过这一方式实现数据的交换和共享。

在众多并行计算机体系结构中，NUMA（Non-Uniform Memory Access）架构是指非均匀访问存储模型，这种模型将CPU的资源分开，以node为单位进行切割，每个node里有着独有的计算核心、内存等资源。NUMA架构提升了CPU性能，却造成了如下两个问题：

首先，当多线程并行应用程序运行在NUMA节点时，若分布在相邻核上的线程之间交换数据，一般将通过邻近几个核心共享的L2 Cache进行；若分布在同一块处理器内物理位置较远的核心上的线程之间交换数据，一般将通过一个处理器共享的L3 Cache进行；若分布在不同处理器上的线程之间交换数据，则只能借助片上互连，从其他处理器的内存中读取。很显然，以上列举的三种方式的代价是不同的，第二种略大于第一种的开销，而由于NUMA架构的特性，第三种开销远大于前两者。因此可以说，在NUMA架构上运行的多线程并行程序通信很不均匀，这就影响了程序运行时的访存效率，最终降低程序整体执行效率。

其次，现代NUMA架构每个node集成的计算核心数量呈增多趋势，在应用程序运行的一段时间内，对于某个node其内部的一些核心发生大量内存访存行为，这样会增加这个node的内存带宽，使其内部的所有内存访问延迟增加。特别地，当这个node的内存带宽超过其硬件上规定的临界带宽，内存访问延迟会急剧升高，大大降低了程序的访存效率，影响了整体性能。这种现象是由于节点间内存带宽不均衡导致的，称之为内存拥塞问题。为了避免内存拥塞问题的发生，需要寻找一种策略来平衡node之间的内存带宽，使每个node内存带宽大致相同以达到提高程序访存效率提升程序性能的目的。

以上两个问题其实存在矛盾的地方，若为了平衡线程之间的通信开销，则需要使线程间的通信尽量发生在node内部，避免过多的跨结点通信。这样会潜在地导致node之间的内存带宽不均衡，从而出现内存拥塞问题。为了有效解决上述两个问题，需要设计一种合理的映射机制，对这两个因素综合考虑，提升程序性能。

现有的线程映射方法包括静态映射与动态映射方法，静态映射方法是指在程序首次运行时利用工具对程序进行插桩，分析程序运行时性能。后续通过一系列决策，根据需求计算出最合适的映射方法，在以后相同的程序执行之前，调用已计算好的映射方法，设置线程的亲和度，将线程绑定到具体的核心上。动态映射方法相较于静态映射是在程序运行过程中根据某一时段线程通信情况迁移线程，这样虽然可以精准捕获程序不同时段的访存行为，但整个过程动态迁移线程造成不小的额外开销，可能会抵消掉一些映射的优化效果，且实现起来相对困难。

目前成熟的一些静态线程映射方法针对线程通信量矩阵通过图论方法、启发式算法和数学规划方法来寻求接近最优的映射方法。但是上述分组方法仅在通信量的空间维度求解问题，可能会导致内存拥塞问题。

**发明内容**

本发明的目的在于提供一种基于多线程共享内存通信模式的静态的线程到核映射优化方法，以解决上述问题。

本发明的技术方案是这样实现的：

本发明设计并实现了多线程环境下静态的线程到核的映射方法，方法中包含一种线程到核的映射机制。本发明将该机制分为三个功能模块，其中通信检测部分借助Intel 提供的Pin动态二进制插桩框架加以实现。映射机制的流程具体包含以下步骤：

步骤1，流程开始，预运行应用程序进行通信检测，分别得到时域通信列表与空域通信量矩阵；

步骤2，将得到的空域通信量矩阵输入到基于贪心策略的分组算法进行处理，输出得到分组后的全部线程对；

步骤3，将步骤2中的分组后的所有线程对结合步骤1中的时域通信列表得到所有线程对的通信时间戳信息；

步骤4，将步骤3中的所有线程对的通信时间戳信息使用Kmeans算法进行聚类，将聚类结果结合机器的硬件拓扑结构得到线程与计算核心的映射关系；

步骤5，将步骤4中的映射关系输入至迁移线程模块，使用hwloc库的cpu绑定函数将线程一一绑定至核，流程结束。

进一步地，整个流程具体实现细节如下：

1）基于Pin tools实现通信检测模块。使用Numalize通信检测工具，Numalize基于Pin tools开发，以缓存行粒度追踪应用程序的所有内存访问,当不同线程访问缓存行时检测到线程间的通信，输出为通信矩阵。

为了同时统计通信的时间戳信息，对Numalize进行修改，这样可统计每个通信事件的时刻信息，输出表头为<timestamp,t1,t2>的表格。

2）基于贪心策略的分组算法及时间戳聚类算法实现计算映射模块。分析全局通信矩阵，借鉴Eagermap的贪心策略进行线程两两配对，使通信最频繁的两个线程组对得到thread\_pairs。thread\_pairs结合时域通信列表得到这些pairs的通信时间戳信息。

针对pairs的通信时间戳信息。利用带权重的KMeans聚类算法，目标函数为 。其中k表示类别个数，Ci 表示类别i ,t表示当前时间戳，Acommt表示t时间戳下的通信量，μi表示类别i所有时间戳的平均值。KMeans通过最小化上述目标函数将时间轴上接近且通信量接近的时间戳聚为一类。最后选择通信量最大那一类提取不同线程对，将这些对放在相邻的核心上(减少remote access)并分散到不同节点(缓解memory congestion)得到线程与计算核心的映射关系map={thread\_id;core\_id}。

3）遍历计算映射得到的map，使用hwloc库的hwloc\_set\_cpubind函数将线程一一绑定至核。

本发明的技术效果如下：

1）优化了多线程并行程序在NUMA架构下运行的通信效率和访存效率。本发明提出的映射方法通过将通信频繁的线程对映射到物理位置相邻的计算核心上，同时将通信时间接近的不同线程对分散映射在不同节点上。不但避免了两个线程之间通过处理器片间互连的高开销方式进行数据交换和共享，而且缓解了由于节点间内存带宽不均衡导致的内存拥塞问题，因此该映射方法优化了各线程访问内存数据的效率，整体上提升了程序的执行性能。

2）解决了传统映射无法适应通用的NUMA处理器平台和通用的多线程并行应用程序的问题。首先，本发明中使用的贪心策略的分组算法及时间戳聚类算法，获取计算平台的硬件拓扑结构进行线程分组计算，能够适应不同类型和不同规模NUMA多核处理器平台。其次，本发明采用静态的线程映射机制，在不同程序运行时均可采用本发明中的方法，因此独立于应用程序，具有通用性。

**附图说明**

图1是本发明实施所述的方法流程图。

**具体实施方式**

下面结合附图和实施例，对本发明的具体实施方法做进一步的详细解释。以下实施例用于说明本发明的具体实施方法，但并不会限制本发明的范围。

参照图1所示，本发明所述静态的线程到核映射方法基本流程，包括：

步骤1：流程开始，预运行应用程序进行通信检测，具体使用Numalize通信检测工具。定义一个hash表、一个通信量矩阵以及一个通信时序表；对内存物理地址空间进行按照缓存行粒度进行划分，索引为划分后的缓存行地址，hash表项为一个队列，记录了在该缓存行上进行访存的不同的线程ID. 规定队列大小为4，以保证统计的精确度。每次当队列中记录入一个新的线程，将队首元素出队，然后计该新线程与仍在队列中的其他线程之间产生一次通信，将通信量矩阵的对应位置的元素加1。同时使用rdtsc系统函数读取发生通信的时间戳timestamp，在通信时序表中增加一条记录<timestamp,t1,t2>,t1,t2为发生通信的线程对id。通信检测完毕后，得到一个通信量矩阵以及一个通信时序表。

步骤2：将通信量矩阵使用贪心分组算法处理。从通信量矩阵A中随机选择一个线程i（初始可选择0号线程），之后搜索矩阵A找到线程j，满足a(i,j) = max(a(i,k)) (k,k≠0)，重复上述步骤直至将n个线程分为n/2组。得到全部线程对。

步骤3：所有线程对分别遍历步骤1的通信时序表，每对线程分别得到自己的时间序列{t1,t2…..tn} ,之后综合所有对的通信时间序列，得到所有线程对的通信时间戳序列ts。

步骤4：将步骤3得到的时间戳序列ts，使用Kmeans算法聚类。使用带权重的KMeans聚类算法，目标函数为 。k表示类别个数，Ci 表示类别i ,t表示当前时间戳，Acommt表示t时间戳下的通信量，μi表示类别i所有时间戳的平均值。KMeans通过最小化上述目标函数将时间轴上接近且通信量接近的时间戳聚为一类。最后选择通信量最大那一类提取不同线程对，将这些对放在相邻的核心上(减少remote access)并分散到不同节点(缓解memory congestion)。得到线程与计算核心的映射关系map={thread\_id:core\_id}。

步骤5：将映射关系输入至线程绑定函数执行映射。遍历步骤4得到的线程与核映射关系map中的每个键值对{thread\_id:core\_id}，使用hwloc库的hwloc\_set\_cpubind函数将线程一一绑定至核,映射流程结束。

本发明适用于NUMA多核处理器平台下的任意多线程并行应用程序。

以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

图示

描述已自动生成

图1

本发明公开一种基于内存拥塞感知的多线程静态映射方法，该方法首先预运行应用程序以追踪检测其访存行为，之后根据追踪到的访存特征将线程绑定至计算核上，解决NUMA架构下线程间通信开销不均衡以及节点间内存带宽不均衡发生内存拥塞导致的程序性能下降问题。该机制包括以下步骤：利用基于Intel提供的pin工具检测线程间的通信情况；定义通信量矩阵以及通信时序表统计通信信息；利用基于贪心策略的分组算法以及KMeans聚类计算线程的合理映射；根据映射计算结果迁移线程。

图示

描述已自动生成