**专利申请明细表**

**兹全权委托北京博思佳知识产权代理有限公司办理下列申请专利有关事项**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 发 明 名 称 | | **一种拓扑驱动的软硬协同图模式匹配方法** | | |
| 第一  申请人 | 姓名或名 称 | 之江实验室 | | |
| 地址、邮编 | 默认：浙江省杭州市余杭区之江实验室南湖总部，311121  日本专利：之江实验室 中国310023浙江省杭州市余杭区文一西路1818号 | | |
| 企业机构代码或身份证号 | 12330000MB1478604D | | |
| 第二  申请人  （如果为多个申请单位共同申请，可填） | 姓名或名 称 | **华中科技大学** | | |
| 地址、邮编 | **地址：湖北省武汉市洪山区珞喻路1037号，430074** | | |
| 企业机构代码或身份证号 | **12100000441626842D** | | |
| 发明人  或设计人 | 姓 名 | 张宇、齐豪、罗康、赵进、张湛 | 第一发明人身份证号 | **431122198707170515** |
| 专利负责人 | 姓名 | 吴梅英，18662590857，wumeiying@zhejianglab.com  地址：浙江省杭州市余杭区之江实验室南湖总部主楼1402办公室，311100 | | |
| 地址、邮编 |
| 电话/E-mail |
| 技术（撰写）  联系人 | 姓名 | **齐豪、**浙江省杭州市余杭区之江实验室南湖总部11号楼B座B102室，**310023、16619714096，theqihao@hust.edu.cn** | | |
| 地址、邮编 |
| 电话/E-mail |
| 申请专利类型 | | R发明　　　　　实用新型　　　　外观设计 | | |
| 催缴费联系人电话、Email、地址 | | 吴梅英，18662590857，wumeiying@zhejianglab.com，地址：浙江省杭州市余杭区之江实验室南湖总部主楼1402办公室，311100 | | |

注：

1、红色字体部分需要填写或选择

说 明 书 摘 要

本发明涉及一种拓扑驱动的软硬协同图模式匹配方法，方法包括：动态图划分模块首先识别出中心顶点，之后从中心顶点开始搜索，将图结构划分为多个子图，以提高图模式匹配的访存局部性；冗余集合计算检测模块动态检测并且统计集合计算的次数，并保存频繁集合计算的结果，从而减少了后续的频繁集合计算的冗余计算；混合计算模块支持多种类型的集合计算，并且为每种集合计算类型提供了代价函数，以选择最优的集合计算类型，从而减少大数据环境下图模式匹配的运行时间。

权 利 要 求 书

1、一种拓扑驱动的软硬协同图模式匹配方法，包括以下步骤：

1. 图结构的动态划分。首先为每个处理核心分配一个中心顶点，然后从这个中心顶点出发进行深度优先遍历或者广度优先遍历，直到深度达到阈值或者已经没有可以访问的顶点。
2. 冗余集合计算检测。在处理一个子图时，每个处理核心统计集合计算的次数，如果次数达到一定的阈值，当前处理核心将该集合计算的结果保存。当下次再进行已保存的集合计算时，直接使用之前保存的结果，从而减少了冗余的集合计算。
3. 混合集合计算。根据二个集合的长度和硬件平台特征，设计一个集合计算性能代价函数。基于这个代价函数，选择合适的集合计算算法。

根据一种优选的实施方式，所述图结构的动态划分步骤至少包括如下图结构的动态划分子步骤：

1. 选择中心顶点：在进行动态图划分之前，统计度数大一定阈值的顶点，后续将这些顶点作为遍历的起始顶点。
2. 执行遍历划分子图：每个处理核心从中心顶点出发进行搜索，在搜索的过程中维护一个访问顶点数组。当搜索的深度达到阈值或者顶点都已经被访问过时停止。访问过的顶点被放入一个队列中。在图结构动态划分的同时，当前处理核心并行地处理队列中的顶点，并且使用窃取工作技术来减少负载不均衡的情况。

根据一种优选的实施方式，所述冗余集合计算检测步骤至少包括如下子步骤：

1. 检测冗余集合计算：集合操作需要二个集合和一个操作类型，将这三个信息映射成一个独一无二的索引，然后在每次做集合计算之前，检测集合操作映射成的索引，并统计索引值出现的次数。
2. 保存冗余集合计算：当索引值重复的次数达到阈值，将当前集合计算的结果保存。当下一次进行同样的集合计算时，直接使用之前保存结果，从而避免了冗余的集合计算。
3. 避免多处理核心同步：当同一个集合计算被多个处理核心计算时，对这个集合计算的统计值需要同步，导致大量的额外开销。由于在动态划分步骤我们将图划分为多个子图，这使得同一个集合计算很可能被同一个处理核心处理。为了避免同步开销，每个处理核心申请独立的保存结果的数据空间，它们互不干扰。

根据一种优选的实施方式，所述混合集合计算检测步骤至少包括如下子步骤：

1. 混合集合计算的类型：集合计算当前有四种主流的计算方式
   1. 基于合并的算法: 使用有序数组存储数据，不断迭代二个数组，判断是否相等。
   2. 基于二分查找的算法：使用有序数组存储数据，不断迭代长度小的数组，在长度大的数组中进行二分查找。
   3. 基于哈希的算法：将一个集合做哈希并保存，遍历另外一个集合，并计算哈希值，判断是否存在第一个集合哈希的结果中。
   4. 基于比特数组的算法：使用比特数组存储一个顶点的邻居，并通过按位与的操作判断二个集合的相同元素
2. 集合计算的硬件单元：针对计算-访存比大的计算类型，在计算单元中设计专用硬件，基于合并的算法；针对计算-访存比小的计算类型，使用近内存架构减少数据的移动，比如基于比特数组的算法。
3. 混合集合计算的代价函数：目标是运行时间最短，每种计算类型的时间可以分为计算时间和访存时间。由于二者可并行，因此代价函数是MIN(计算时间，访存时间)。

说 明 书

**一种拓扑驱动的软硬件协同图模式匹配方法**

**技术领域**

本发明属于大数据处理的图模式匹配技术领域，具体涉及一种拓扑驱动的软硬件协同图模式匹配方法，利用图数据结构特征优化大规模数据下的图模式匹配。

**背景技术**

随着大数据时代的到来，最近的研究更加关注新兴的图模式匹配问题，即探索图的特定模式。图模式匹配可以解决许多图分析问题。比如，k团（k-clique listing，k-CL)列出了图G (k ≥ 3) 中的所有 k-cliques; 子图列表（SL）列出了与模式 P 同构的 G 的所有子图; k-motif 计数（k-MC）计算所有可能的k-顶点模式出现的次数; k-频繁子图挖局（k-frequent subgraph mining，k-FSM)：给定k 和阈值σmin，k-FSM考虑所有边少于k 的模式，如果图中匹配模式P的个数σ大于σmin，则列出模式P，这称为频繁的模式。图模式匹配已广泛应用于许多领域，如社交网络分析、机器学习、生物和化学信息学等。

最先进的图模式匹配算法是模式感知的。它将用户定义的模式编译成优化的执行计划以指导图匹配过程。具体来说，模式感知算法从图中的每个单独顶点开始探索搜索树，并使用集合交集和差集操作来确定下一个候选顶点以在部分匹配的子图中扩展。此外，它还定义模式中二个顶点的偏序进行剪枝。这种范式极大地减少了内存占用和提高了性能。因此，最近的软件系统和硬件加速器都采用这种模式感知的方式。

然而，加速图模式匹配仍然面临着很多挑战：（1）大量的数据传输：图模式匹配需要重复访问图数据。图模式匹配从图中的每一个顶点出发，然后进行DFS搜索（称为Task）。多个Task之间存在对相同图数据的大量重复访问（比如一个顶点邻居）。图模式匹配访问的图数据是原始图数据的几十倍甚至是几百倍，导致 CPU 和内存之间的大量数据传输。（2）冗余集合计算：图模式匹配过程中存在大量相同的集合计算。现实世界中的图多数是幂律图，存在大量的高度数顶点，而大量集合计算都与这些顶点相关联。这些相同集合计算的次数高达上百甚至上千，导致大量的冗余计算。（3）集合计算的多样性：在图模式匹配中，做集合操作的二个集合的长度之比（长集合/短集合）是多样性。一种集合计算的算法难以满足所有长度比值的集合计算的性能要求。比如，当二个集合的长度是类似时，适合使用基于合并的算法来计算（同时遍历二个集合，找出相同的元素）；当二个集合的长度差别很大时，适合使用基于二分查找的算法来计算（遍历小集合，在长集合中进行二分查找）。

为了解决上述挑战，本发明提出了一种拓扑驱动的软硬件协同图模式匹配方法，根据图结构的特性自适应的提供图划分和图模式匹配的调度方案，减少数据传输和提高数据的访存时间局部性。基于调度方案，本发明还提出一个冗余检测方案和一个混合集合计算的方案分别减少和加速集合计算。

**发明内容**

针对现有技术之不足，本发明提出一种拓扑驱动的软硬件协同图模式匹配方法，解决了数据传输高、冗余集合计算、集合计算多样性的问题，提升了系统的性能。

为实现上述目的，本发明包括以下步骤：

1. 图结构的动态划分。首先为每个处理核心分配一个中心顶点，然后从这个中心顶点出发进行广度优先或者深度优先遍历，直到遍历的深度达到阈值或者已经没有可以访问的顶点。
2. 冗余集合计算检测。在处理一个子图时，每个处理核心统计集合计算的次数，如果次数达到一定的阈值，当前处理核心将该集合计算的结果保存。当下次再进行已保存的集合计算时，直接使用之前保存的结果，从而减少了冗余的集合计算。
3. 混合集合计算。根据二个集合的长度和硬件平台特征，设计一个集合计算性能代价函数。基于这个代价函数，选择合适的集合计算算法。

根据一种优选的实施方式，所述图结构的动态划分步骤至少包括如下图结构的动态划分子步骤：

1. 选择中心顶点：在进行动态图划分之前，统计度数大一定阈值的顶点，后续将这些顶点作为遍历的起始顶点。
2. 执行遍历划分子图：每个处理核心从中心顶点出发进行搜索，在搜索的过程中维护一个访问顶点数组。当搜索深度达到阈值或者顶点都已经被访问过时停止。被访问过的顶点放入一个队列中。在图结构动态划分的同时，当前处理核心并行地处理队列中的顶点，并且使用工作窃取技术来减少负载不均衡的情况。

根据一种优选的实施方式，所述冗余集合计算检测步骤至少包括如下子步骤：

1. 检测冗余集合计算：集合操作需要二个集合和一个操作类型，将这三个信息映射成一个独一无二的索引，然后在每次做集合计算之前，检测集合操作映射成的索引，并统计索引值出现的次数。
2. 保存冗余集合计算：当索引值重复的次数达到阈值，将当前集合计算的结果保存。当下一次进行同样的集合计算时，直接使用之前保存结果，从而避免了冗余的集合计算。
3. 避免多处理核心同步：当同一个集合计算被多个处理核心计算时，对这个集合计算的统计值需要同步，导致大量的额外开销。由于在动态划分步骤我们将图划分为多个子图，这使得同一个集合计算很可能被同一个处理核心处理。为了避免同步开销，每个处理核心申请独立的保存结果的数据空间，它们互不干扰。

根据一种优选的实施方式，所述混合集合计算检测步骤至少包括如下子步骤：

1. 混合集合计算的类型：集合计算当前有四种主流的计算方式
   1. 基于合并的算法: 使用有序数组存储数据，不断迭代二个数组，判断是否相等。
   2. 基于二分查找的算法：使用有序数组存储数据，不断迭代长度小的数组，在长度大的数组中进行二分查找。
   3. 基于哈希的算法：将一个集合做哈希并保存，遍历另外一个集合，并做哈希，判断是否存在第一个集合哈希的结果中。
   4. 基于比特数组的算法：使用比特数组存储一个顶点的邻居（通常的是高度数顶点的邻居），并通过按位与的操作判断二个集合的相同元素
2. 集合计算的硬件单元：针对计算-访存比大的计算类型，在计算单元中设计专用硬件，比如基于合并的算法；针对计算-访存比小的计算类型，使用进内存架构减少数据的移动，比如基于比特数组的算法。
3. 混合集合计算的代价函数：目标是运行时间最短，每种计算类型的时间可以分为计算时间和访存时间。由于二者可并行，因此代价函数是MIN(计算时间，访存时间)。其中计算时间使用算法的时间复杂度估计；顺序访存时间使用 访存量/带宽 估计；随机访存使用 访存量\*访存延迟 估计。

本发明提供的一种拓扑驱动的软硬件协同图模式匹配系统，与现有技术相比，本发明至少具有益效果为：

1. 减少冗余访存：本发明将图分解为多个子图，然后每个子图分配给一个处理单元处理；按照这种方式，当处理一个子图中的多个任务时，访问的图数据具有很高的时间局部性，从而降低了冗余的访存。
2. 减少冗余集合计算：本发明通过首先分析重复次数较高的集合计算，然后将这些集合计算的结果保存，从而避免了后续进行相同的集合计算，因此起到了减少冗余计算的作用。
3. 加速集合计算：本发明首先提出了一个集合计算的代价模型，该模型通过分析二个集合长度和硬件特征估计出不同集合计算算法的性能，然后基于这个性能进行选择使用合适的集合计算算法。

**具体实施方式**

本实施例公开中一种基于多核CPU的图模式匹配处理系统（也可以基于GPU、ASIC、FPGA、PIM等硬件实施）。如图1所示，该系统至少包括划分模块（Partition Uint，PU）、冗余检测模块（Redundancy Detection）、集合派遣模块（Sets Dispatcher,SD）。PU模块主要负责图的动态图划分；RD模块主要负责冗余集合计算的检测和统计，以减少冗余集合计算；SD模块负责多种集合计算算法性能的估计和决定集合计算的算法类型。

优选地，图动态划分模块PU配置为如下子步骤对图进行划分：

1. 初始化：每个处理单元将中心顶点、图数据结构的格式以及地址、访问数组（用于判断一个顶点是否被访问过）地址等配置信息传递给PU。
2. 预取顶点：如图3所示，我们设计了三级流水线，首先取顶点，然后根据顶点的访问状态过滤顶点，之后取顶点的邻居。流水线之间添加队列，以提高硬件利用率。
3. 顶点队列：PU将划分的子图通过顶点队列传递给处理单元去处理。注意动态图划分的过程和处理单元进行图模式匹配的过程是并行的。

优选地，集合冗余检测模块RD配置为如下子步骤：

1. 识别冗余的集合计算：一个集合计算由二个集合以及集合操作决定。由原始图数据（比如一个顶点的邻居）组成的集合，我们使用源顶点的ID来标识；由中间数据（比如二个集合交集的中间结果）组成的集合，我们不统计它们，因为该集合组成的集合计算冗余计算频率低并且集合长度较短，计算时间短。
2. 冗余集合检测过程：如图3所示，执行从集合计算开始，包括两个输入集合（set0 和 set1）和一个集合操作。如果集合计算的结果已经保存，则冗余检测器将返回结果的地址。 否则，当前集合计算将被计算并记录它已执行的次数。

优选地，集合分配模块SD配置为如下子步骤：

1. 集合计算的硬件单元：针对计算-访存比大的计算类型，在计算单元中设计专用硬件，比如基于合并的算法。针对计算-访存比小的计算类型，使用近内存架构减少数据的移动，比如基于比特数组的算法。具体来说，在存储器端口附近添加位操作（bitwise opertion），集合计算在存储器的内部完成，减少了存储器与处理单元之间数据移动。
2. 估计集合计算时间：给定两个集合 L 和 S (|L| > |S|),BW为带宽，LM为访存延迟。
   1. 基于合并的算法: MAX(L+S, (L+S)/BW)。
   2. 基于二分查找的算法：LM\*|S|\*log(|L|)。
   3. 基于哈希的算法：(|L|+|S|)\*H（H是hash函数执行的代价）。
   4. 基于比特数组的算法：MAX(|V|/BB, 2\*|V|/BW)（BB是cpu的位宽，V是图中顶点的总个数）。
3. 选择集合计算算法：根据（2）中公式计算出每类算法的执行时间，选择时间最低的算法执行集合计算。估计时间与集合计算是并行的，不会降低性能。

说 明 书 附 图



图1 整体架构图



图2 PU示意图



图3 冗余检测流程图