



(12)发明专利

(10)授权公告号 CN 103942235 B

(45)授权公告日 2017. 11. 03

(21)申请号 201310178513.7

US 2003009352 A1,2003.01.09,

(22)申请日 2013.05.15

Intawat Nookaew 等.“A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in *Saccharomyces cerevisiae*”.《Nucleic Acids Research》.2012,1-14.

(65)同一申请的已公布的文献号

申请公布号 CN 103942235 A

(43)申请公布日 2014.07.23

(73)专利权人 张一凡

地址 250200 山东省济南市经十路3028号

审查员 魏留强

(72)发明人 张一凡

(51)Int.Cl.

G06F 17/30(2006.01)

G06F 9/44(2006.01)

(56)对比文件

WO 2003010659 A1,2003.02.06,

CN 101799809 A,2010.08.11,

US 8386400 B2,2013.02.26,

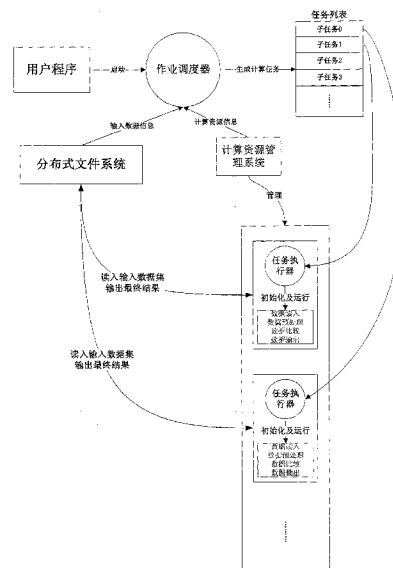
权利要求书2页 说明书8页 附图4页

(54)发明名称

针对大规模数据集交叉比较的分布式计算系统和方法

(57)摘要

本发明涉及一种针对大规模数据集交叉比较的分布式计算系统,包括交叉比较编程模型、主结点、编程接口和基于异构分布式机群的后端分布式处理框架。该分布式计算系统旨在应用分布式计算环境高效处理满足数据集交叉比较模式的计算问题。本发明通过提供用户直观的交叉比较编程模型,帮助用户将待处理计算过程进行抽象简化,实现了对各种不同交叉比较计算问题的统一支持;提供用户简洁的编程接口,帮助用户开发串行交叉比较程序,用户无需掌握并行编程知识;系统隐藏了并行计算的实现细节,用户无需掌握系统内部结构,降低了系统的使用难度。此外,本发明提出的交叉比较编程模型及接口与硬件无关,可以方便的在不同分布式机群环境中实现。



1. 一种针对大规模数据集交叉比较的分布式计算系统,其特征在于,该分布式计算系统包括交叉比较编程模型、主结点、编程接口和基于异构分布式机群的后端分布式处理框架,其中交叉比较编程模型将交叉比较计算抽象为四个独立的计算过程;主结点负责分布式系统的资源管理及任务调度;后端分布式框架负责数据处理计算;编程接口使用户只需要定义相关数据类型及计算逻辑,以上共同使用实现高效处理交叉比较计算问题;所述交叉比较编程模型包括四个独立的计算模块:数据读取模块、数据预处理模块、数据比较模块和数据输出模块;所述数据读取模块为根据计算任务的需要读取不同的输入数据;所述数据预处理模块为预处理输入数据,生成预处理后的中间计算结果,其中预处理是用户自定义的处理方法;所述数据比较模块对两个不同的预处理后中间计算结果进行比较计算,得到最终计算结果,其中比较计算处理是用户自定义的比较计算处理方法;所述数据输出模块收集生成的最终计算结果,以用户定义的方式将最终计算结果生成输出文件;所述编程接口分别为Reader编程接口、Preprocess编程接口、Compare编程接口和Writer编程接口:其中所述Reader编程接口对应数据读取模块,用户自定义读入数据的INDEX(索引)及CRUDEDATA(初始内容)类型,并且自定义读入数据的读入方法及数据类型;所述Preprocess编程接口对应数据预处理模块,用户只需定义读入数据内容的预处理过程计算方法,并定义INTERMEDIATEDATA(预处理后内容)类型;所述Compare编程接口对应数据比较模块,用户只需定义两个不同预处理后数据的比较计算方法,并定义RESULTDATA(比较结果);所述Writer编程接口对应数据输出模块,用户自定义输出数据的格式及数据输出方法。

2. 根据权利要求1所述的一种针对大规模数据集交叉比较的分布式计算系统,其特征在于,所述基于异构分布式机群的后端分布式处理框架,包括作业调度器、任务执行器、分布式机群资源管理系统和分布式文件系统:其中作业调度器运行于分布式机群主结点(Master结点),用户在提交计算作业后,作业调度器根据整个分布式机群中空闲资源情况,作业对资源的需求及待处理数据集的存储情况生成计算任务队列,依据任务分配策略将不同任务发布到各个从结点中;其中任务执行器运行于分布式机群从结点(Slaves结点),任务执行器在接收到相应任务之后启动子任务处理过程,依照用户编写的方法进行数据的读入,预处理,比较和输出操作;其中分布式机群资源管理系统运行于分布式机群所有结点,起到管理分布式机群中所有结点计算机,监控资源利用情况,控制结点之间通信及数据传输,及处理结点故障的作用;其中分布式文件系统运行于分布式机群所有结点,用于存储输入数据集及最终输出结果。

3. 一种利用如权利要求1所述分布式计算系统处理数据的方法,其特征该方法包括步骤如下:

(1) 用户分析具体计算问题;

(2) 用户应用本发明所述分布式计算系统所提供的编程接口,分别实现四个独立的计算模块:数据读取模块、数据预处理模块、数据比较模块和数据输出模块的具体处理方法,包括步骤(a)-(d):

(a) 数据读入阶段:在此阶段,子任务执行所需要的子数据集从分布式文件系统中被读入,数据集集中的每一项输入文件在本发明所述的分布式计算系统中以索引A、初始内容的方式存储;

(b) 数据预处理阶段:在此阶段,步骤(1)中读入的数据按照用户自定义的处理方法进

行预处理,得中间计算结果在本发明所述的分布式计算系统中以索引A、预处理后内容的方式存储;

(c) 数据比较阶段:在此阶段,用户自定义的比较计算处理方法从步骤(2)的中间计算结果中取出两个不同数据进行比较计算,得最终计算结果在本发明所述的分布式计算系统中以索引A、索引B和比较结果的方式存储;其中索引A,B分别对应两个不同的输入文件的索引;

(d) 数据输出阶段:在此阶段,所述数据输出模块收集生成的最终计算结果,以用户定义的方式将最终计算结果生成输出文件;

(3) 用户编写包括以上四个独立的计算模块的类文件、确定输入数据集大小、分布式系统结点数量和各结点空闲资源情况,在后端分布式处理框架中进行设置;

(4) 将应用前端编程接口定义的处理方法及待处理数据上传至后端分布式处理框架,其中数据存储至分布式文件系统中;

(5) 后端分布式处理框架在主结点启动作业调度器,分析用户计算问题,自动生成相互独立的子任务集并将任务分配至不同从结点;在各个从结点启动任务执行器,从分布式文件系统中读取相应数据,执行具体子任务;当所有子任务执行完成后,后端分布式处理框架结束计算任务,输出最终结果并存入分布式文件系统中。

## 针对大规模数据集交叉比较的分布式计算系统和方法

### 技术领域

[0001] 本发明涉及一种针对大规模数据集交叉比较的分布式计算系统和方法,属于分布式计算技术领域。

### 技术背景

[0002] 交叉比较问题又称为数据集交叉连接或笛卡尔乘积问题,计算空间包括两个数据集所有元素的可能组合。针对数据集中所有元素进行交叉比较的计算问题广泛存在于生物计算,数据挖掘和模式识别等领域。比如生物计量学中,为处理人脸识别问题需要对数据库中海量人脸样本进行逐一对比得到相似度;生物信息学中,为深入分析物种进化特征需要对海量DNA, RNA等基因序列进行逐条比对从而得到基因序列关系矩阵;数据挖掘学中,知识挖掘过程中处理噪声的重要步骤就是对数据集逐项与噪声进行分析。因此,高效处理此类计算问题具有很重要的科研及商业价值。

[0003] 随着待处理数据规模的不断增大,数据结构差异化增加,传统解决方案在应对此类问题时显出越来越大的局限性。目前的解决方案主要分为两种,一种设计针对小规模数据集,采用单机处理或在分布式计算平台中的每个计算结点统一部署全部数据的方法,无法解决海量大数据集交叉比较的问题;另一种方案采用传统批处理方式将数据集分割打包成任务集由不同的计算结点并行处理完成,此种方案存在计算资源利用率低,总处理能力及扩展能力受限于批处理任务系统,同样无法高效解决海量大数据集交叉比较的问题。

[0004] 对于大规模数据集交叉比较问题,整个计算问题可以被划分成不同的计算任务被不同的计算结点并行处理,但不同的计算任务可能使用同一子数据集,整个计算问题的描述如图1所示:

[0005] 图1所示的大规模数据集交叉比较计算问题主要包含两个计算过程:

[0006] 1) 数据集中的每一项子数据作为输入进行预处理计算,得到预处理后的中间结果;

[0007] 2) 任意两个不同子数据的预处理后的中间结果作为比较计算的输入,得到最终比较计算结果。

[0008] 在典型的两数据集交叉比较问题中,整个计算空间如图2所示,每个星号代表两个数据集中的元素进行了一次交叉比较运算,对集合 $A = \{A_1, A_2 \cdots A_n\}$ 和集合 $B = \{B_1, B_2 \cdots B_m\}$ ,整个过程需要进行 $N = n * m$ 次交叉比较运算。

[0009] 数据集在进行比较计算之前需要对其中的元素进行预处理计算。相对于比较计算,预处理计算同样需要大量的计算资源与计算时间。因此数据集元素预处理与元素比较运算构成了交叉比较运算中的主要计算过程。同样对于集合 $A = \{A_1, A_2 \cdots A_n\}$ 和集合 $B = \{B_1, B_2 \cdots B_m\}$ ,整个计算过程可以用以下的伪代码描述:

[0010] For each item  $i$  in data set A

[0011] For each item  $j$  in data set B

[0012]  $M[i, j] = F(P(A[i]), P(B[j]))$

[0013] 其中M表示计算结果矩阵,F表示交叉比较运算,P表示预处理运算。

[0014] 此外,目前解决方案在使用过程中存在着并行程序开发困难,用户需要掌握计算平台内部实现方式,系统只针对解决某一特殊计算问题等缺点。因此,迫切需要研究一种针对海量数据集进行交叉比较的分布式计算系统,既满足高效处理大规模数据集的需要,又可以为用户提供简洁易用的使用接口,同时面向一般性的数据集交叉比较计算问题。

## 发明内容

[0015] 针对现有技术的局限与不足,本发明提供一种针对大规模数据集交叉比较的分布式计算系统。

[0016] 本发明还提供一种利用上述分布式计算系统处理数据的方法。

[0017] 本发明的技术方案在于:

[0018] 一种针对大规模数据集交叉比较的分布式计算系统,包括交叉比较编程模型、编程接口和基于异构分布式机群的后端分布式处理框架。该分布式计算系统旨在应用分布式计算环境高效处理满足数据集交叉比较模式的计算问题。本发明通过提供用户直观的交叉比较编程模型,帮助用户将待处理计算过程进行抽象简化,实现了对各种不同交叉比较计算问题的统一支持;提供用户简洁的编程接口,帮助用户开发串行交叉比较程序,用户无需掌握并行编程知识;系统隐藏了并行计算的实现细节,用户无需掌握系统内部结构,降低了系统的使用难度。

[0019] 根据本发明优选的,所述的交叉比较编程模型包括四个独立的计算模块:数据读取模块、数据预处理模块、数据比较模块和数据输出模块;所述数据读取模块为根据计算任务的需要读取不同的输入数据;所述数据预处理模块为预处理输入数据,生成预处理后的中间计算结果,其中预处理是用户自定义的处理方法;所述数据比较模块对两个不同的预处理后中间计算结果进行比较计算,得到最终计算结果,其中比较计算处理是用户自定义的比较计算处理方法;所述数据输出模块收集生成的最终计算结果,以用户定义的方式将最终计算结果生成输出文件。在本发明所述分布式计算系统中的主结点调度下,四个独立计算模块在不同子结点协同运行,构成了一个具体的交叉比较计算任务。

[0020] 为指导用户编程,本发明所述分布式系统还提供了4个编程接口,所述的4个编程接口的使用方法包括步骤(i)-(iv):

[0021] (i) 用户创建自定义类并实现此编程接口;

[0022] (ii) 用户根据具体计算问题定义INDEX(索引),CRUDEDATA(初始内容),INTERMEDIATEDATA(预处理后内容)及RESULTDATA(比较结果)的数据类型;

[0023] 其中索引值一般为原始输入数据文件名;初始内容为原始输入数据;预处理后内容为输入数据在预处理过程后得到的计算结果;比较结果为输入数据在进行比较运算后得到的最终计算结果。

[0024] (iii) 用户根据具体计算问题编写算法;

[0025] (iv) 用户将编写完成的类文件提交后端分布式计算处理框架,执行计算任务;

[0026] 编程接口如下所示(以JAVA语言示例):

```

public interface ComputingSystem<INDEX,CRUDEDATA,INTERMEDIATEDATA,RESULTDATA>
{
    Map<INDEX,CRUDEDATA> Reader();

    INTERMEDIATEDATA Preprocess(CRUDEDATA crudedata);

[0027]    RESULTDATA Compare(INTERMEDIATEDATA intermediatedataa,INTERMEDIATEDATA
intermediatedatab);

    void Writer();
}

```

[0028] 根据本发明优选的,所述的编程接口分别为Reader编程接口、Preprocess编程接口、Compare编程接口和Writer编程接口:

[0029] 其中所述Reader编程接口对应数据读取模块,用户自定义读入数据的INDEX(索引)及CRUDEDATA(初始内容)类型,并且自定义读入数据的读入方式及数据类型;提供了很好的兼容性;

[0030] 所述Preprocess编程接口对应数据预处理模块,用户只需关注读入数据内容的预处理过程,定义INTERMEDIATEDATA(预处理后内容)类型;此方法与INDEX(索引)分离,使用户只需关注核心预处理过程;

[0031] 所述Compare编程接口对应数据比较模块,用户只需关注两个不同预处理后数据的比较过程,定义RESULTDATA(比较结果);此方法同样与INDEX(索引)分离,使用户只需关注核心比较过程;

[0032] 所述Writer编程接口对应数据输出模块,用户自定义输出数据的格式及形式;方便用户将计算数据导入其它计算平台。

[0033] 根据本发明优选的,所述基于异构分布式机群的后端分布式处理框架,包括作业调度器、任务执行器、分布式机群资源管理系统和分布式文件系统:

[0034] 本发明针对大规模异构分布式机群计算节点可用资源变化,硬件配置不同的特点,提出了大规模数据集交叉比较分布式计算系统后端处理框架,所述分布式处理框架主要负责计算任务的执行,分布式系统的管理及数据集的存储工作。大规模数据集交叉比较分布式计算系统后端处理框架组成如图4所示:

[0035] 其中作业调度器运行于分布式机群主结点(Master结点),用户在提交计算作业后,作业调度器根据整个分布式机群中空闲资源情况,作业对资源的需求及待处理数据集的存储情况生成计算任务队列,依据任务分配策略将不同任务发布到各个从结点中;

[0036] 其中任务执行器运行于分布式机群从结点(Slaves结点),任务执行器在接收到相应任务之后启动子任务处理过程,依照用户编写的方法进行数据的读入,预处理,比较和输出操作;

[0037] 其中分布式机群资源管理系统运行于分布式机群所有结点,起到管理分布式机群中所有结点计算机,监控资源利用情况,控制结点之间通信及数据传输,及处理结点故障的作用;

[0038] 其中分布式文件系统运行于分布式机群所有结点,用于存储输入数据集及最终输出结果;本发明支持读写多种分布式文件系统,在这里选取Hadoop分布式文件系统。

[0039] 一种利用上述分布式计算系统处理数据的方法,包括步骤如下:

[0040] (1) 用户分析具体计算问题;

[0041] (2) 用户应用本发明所述分布式计算系统所提供的编程接口, 分别实现四个独立的计算模块: 数据读取模块、数据预处理模块、数据比较模块和数据输出模块的具体处理方法, 包括步骤(a) - (d):

[0042] (a) 数据读入阶段: 在此阶段, 子任务执行所需要的子数据集从分布式文件系统中被读入, 数据集中的每一项输入文件在本发明所述的分布式计算系统中以索引A、初始内容的方式存储;

[0043] (b) 数据预处理阶段: 在此阶段, 步骤(1)中读入的数据按照用户自定义的处理方法进行预处理, 得中间计算结果在本发明所述的分布式计算系统中以索引A、预处理后内容的方式存储;

[0044] (c) 数据比较阶段: 在此阶段, 用户自定义的比较计算处理方法从步骤(2)的中间计算结果中取出两个不同数据进行比较计算, 得最终计算结果在本发明所述的分布式计算系统中以索引A、索引B和比较结果的方式存储; 其中索引A, B分别对应两个不同的输入文件的索引;

[0045] (d) 数据输出阶段: 在此阶段, 所述数据输出模块收集生成的最终计算结果, 以用户定义的方式将最终计算结果生成输出文件;

[0046] 如图3所示的交叉比较编程模型自动实现了计算过程并行化, 用户在使用过程中只需将所需计算问题划分为以上四个阶段, 分别定义各阶段在单个计算结点的实现方法, 从而避免考虑并行编程的问题。

[0047] (3) 用户编写包括以上四个独立的计算模块的类文件、确定输入数据集大小、分布式系统结点数量和各结点空闲资源情况, 在后端分布式处理框架中进行设置;

[0048] (4) 将应用前端编程接口定义的处理方法及待处理数据上传至后端分布式处理框架, 其中数据存储至分布式文件系统中;

[0049] (5) 后端分布式处理框架在主结点启动作业调度器, 分析用户计算问题, 自动生成相互独立的子任务集并将任务分配至不同从结点; 在各个从结点启动任务执行器, 从分布式文件系统中读取相应数据, 执行具体子任务; 当所有子任务执行完成后, 后端分布式处理框架结束计算任务, 输出最终结果并存入分布式文件系统中。

[0050] 本发明的优势在于:

[0051] 本发明针对大规模数据集交叉比较的分布式计算系统针对但不限于基于Linux操作系统的异构分布式机群。在典型的异构分布式计算环境中, 每个计算结点都拥有不同的处理及存储能力, 计算结点之间由高速网络连接, 形成统一管理的计算环境。考虑到海量数据集的存储问题, 本分布式计算系统整合了高性能分布式文件系统(Hadoop分布式文件系统), 提供了充足的扩展能力。

[0052] 本发明四个主要模块构成了本分布式计算平台的基本结构, 用户可以通过部署此计算平台到分布式计算机群实现高效处理大规模数据集交叉比较运算的目的。

[0053] 本发明针对大规模数据集交叉比较运算的特点, 充分利用了分布式计算环境的优势, 通过提出统一的交叉比较问题编程模型简化了用户开发不同数据集交叉比较程序的步骤, 提供了简洁的编程接口使用户无需考虑并行编程问题, 并且计算平台对用户隐藏了并行计算过程中数据分配, 任务调度, 资源管理等多方面的实现细节, 降低了系统的使用难度。此外, 本发明提出的交叉比较编程模型及接口与硬件无关, 可以方便的在不同分布式机

群环境中实现。

## 附图说明

[0054] 图1为现有的对大规模数据集交叉比较问题的图解描述；

[0055] 图2为图1大规模数据集交叉比较计算空间；

[0056] 图3为本发明所述的大规模数据集交叉比较编程模型；

[0057] 图4为本发明针对大规模数据集交叉比较的分布式计算系统后端处理框架；

[0058] 图5为利用上述分布式计算系统处理数据的方法流程框图。

[0059] 具体实施方法：

[0060] 下面结合实施例和说明书附图对本发明做详细的说明，但不限于此。

[0061] 实施例1、

[0062] 如图3-5所示。

[0063] 一种针对大规模数据集交叉比较的分布式计算系统，包括交叉比较编程模型、主结点、编程接口和基于异构分布式机群的后端分布式处理框架。该分布式计算系统旨在应用分布式计算环境高效处理满足数据集交叉比较模式的计算问题。本发明通过提供用户直观的交叉比较编程模型，帮助用户将待处理计算过程进行抽象简化，实现了对各种不同交叉比较计算问题的统一支持；提供用户简洁的编程接口，帮助用户开发串行交叉比较程序，用户无需掌握并行编程知识；系统隐藏了并行计算的实现细节，用户无需掌握系统内部结构，降低了系统的使用难度。

[0064] 所述的交叉比较编程模型包括四个独立的计算模块：数据读取模块、数据预处理模块、数据比较模块和数据输出模块；所述数据读取模块为根据计算任务的需要读取不同的输入数据；所述数据预处理模块为预处理输入数据，生成预处理后的中间计算结果，其中预处理是用户自定义的处理方法；所述数据比较模块对两个不同的预处理后中间计算结果进行比较计算，得到最终计算结果，其中比较计算处理是用户自定义的比较计算处理方法；所述数据输出模块收集生成的最终计算结果，以用户定义的方式将最终计算结果生成输出文件。

[0065] 所述的编程接口分别为Reader编程接口、Preprocess编程接口、Compare编程接口和Writer编程接口：

[0066] 其中所述Reader编程接口对应数据读取模块，用户自定义读入数据的INDEX（索引）及CRUDEDATA（初始内容）类型，并且自定义读入数据的读入方式及数据类型；提供了很好的兼容性；

[0067] 所述Preprocess编程接口对应数据预处理模块，用户只需关注读入数据内容的预处理过程，定义INTERMEDIATEDATA（预处理后内容）类型；此方法与INDEX（索引）分离，使用户只需关注核心预处理过程；

[0068] 所述Compare编程接口对应数据比较模块，用户只需关注两个不同预处理后数据的比较过程，定义RESULTDATA（比较结果）；此方法同样与INDEX（索引）分离，使用户只需关注核心比较过程；

[0069] 所述Writer编程接口对应数据输出模块，用户自定义输出数据的格式及形式；方便用户将计算数据导入其它计算平台。



[0070] 根据本发明优选的,所述基于异构分布式机群的后端分布式处理框架,包括作业调度器、任务执行器、分布式机群资源管理系统和分布式文件系统:

[0071] 其中作业调度器运行于分布式机群主结点(Master结点),用户在提交计算作业后,作业调度器根据整个分布式机群中空闲资源情况,作业对资源的需求及待处理数据集的存储情况生成计算任务队列,依据任务分配策略将不同任务发布到各个从结点中;

[0072] 其中任务执行器运行于分布式机群从结点(Slaves结点),任务执行器在接收到相应任务之后启动子任务处理过程,依照用户编写的方法进行数据的读入,预处理,比较和输出操作;

[0073] 其中分布式机群资源管理系统运行于分布式机群所有结点,起到管理分布式机群中所有结点计算机,监控资源利用情况,控制结点之间通信及数据传输,及处理结点故障的作用;

[0074] 其中分布式文件系统运行于分布式机群所有结点,用于存储输入数据集及最终输出结果;本发明支持读写多种分布式文件系统,在这里选取Hadoop分布式文件系统。

[0075] 实施例2、

[0076] 一种利用如实施例1所述分布式计算系统处理数据的方法,包括步骤如下:

[0077] (1) 用户分析具体计算问题;

[0078] (2) 用户应用本发明所述分布式计算系统所提供的编程接口,分别实现四个独立的计算模块:数据读取模块、数据预处理模块、数据比较模块和数据输出模块的具体处理方法,包括步骤(a)-(d):

[0079] (a) 数据读入阶段:在此阶段,子任务执行所需要的子数据集从分布式文件系统中被读入,数据集中的每一项输入文件在本发明所述的分布式计算系统中以索引A、初始内容的方式存储;

[0080] (b) 数据预处理阶段:在此阶段,步骤(1)中读入的数据按照用户自定义的处理方法进行预处理,得中间计算结果在本发明所述的分布式计算系统中以索引A、预处理后内容的方式存储;

[0081] (c) 数据比较阶段:在此阶段,用户自定义的比较计算处理方法从步骤(2)的中间计算结果中取出两个不同数据进行比较计算,得最终计算结果在本发明所述的分布式计算系统中以索引A、索引B和比较结果的方式存储;其中索引A,B分别对应两个不同的输入文件的索引;

[0082] (d) 数据输出阶段:在此阶段,所述数据输出模块收集生成的最终计算结果,以用户定义的方式将最终计算结果生成输出文件;

[0083] 如图3所示的交叉比较编程模型自动实现了计算过程并行化,用户在使用过程中只需将所需计算问题划分为以上四个阶段,分别定义各阶段在单个计算结点的实现方法,从而避免考虑并行编程的问题。

[0084] (3) 用户编写包括以上四个独立的计算模块的类文件、确定输入数据集大小、分布式系统结点数量和各结点空闲资源情况,在后端分布式处理框架中进行设置;

[0085] (4) 将应用前端编程接口定义的处理方法及待处理数据上传至后端分布式处理框架,其中数据存储至分布式文件系统中;

[0086] (5) 后端分布式处理框架在主结点启动作业调度器,分析用户计算问题,自动生成

相互独立的子任务集并将任务分配至不同从结点;在各个从结点启动任务执行器,从分布式文件系统中读取相应数据,执行具体子任务;当所有子任务执行完成后,后端分布式处理框架结束计算任务,输出最终结果并存入分布式文件系统中。

[0087] 下面结合生物信息学中一个具体的大规模RNA基因序列集交叉比较计算问题对本发明做进一步说明,但不限于此。此具体实例依照图5所示计算过程,包含此大规模数据集交叉比较计算系统的执行及针对大规模RNA基因序列集交叉比较计算问题的处理方法两部分

[0088] 针对大规模RNA基因序列集交叉比较计算问题的处理方法,包括步骤1)-2):

[0089] 1) 分析并处理RNA基因序列集交叉比较计算问题,依据本发明提出的大规模数据集交叉比较编程模型明确四个计算模块所需对应的具体计算任务如下所示:

[0090] a. 数据读入模块计算任务为从分布式文件系统读入RNA基因序列,文件形式为\*.ffa,内容格式如下所示:

[0091] >基因序列ID|生物体名称

[0092] 基因序列编码.....

[0093] b. 数据预处理模块,计算任务为对基因序列进行处理,构造频率向量及成份向量;

[0094] c. 数据比较模块,计算任务为对得到的成分向量进行比较处理,计算成份向量之间的距离;

[0095] d. 数据输出模块,计算任务为将得到的输出数据以(RNA序列A,RNA序列B,距离)的方式保存在文件中。

[0096] 2) 应用本系统提供的大规模数据集交叉比较编程模型前端编程接口,实现以上四个阶段的具体计算方法,生成用户程序类文件,如下为reader接口代码示例:

```
public class Cvtreesystem extends ComputingSystem<String,String,List<Object>,Double>
```

```
{
```

```
    public HashMap<String,String> Reader()
```

```
    {
```

```
        Configuration conf = new Configuration();
```

```
        HashMap<String,String> pairsforpreprocess = new HashMap<String,String>{};
```

```
        try {
```

[0097]

```
            FileSystem fs = FileSystem.get(conf);
```

```
            FSDataInputStream hdfsInStream;
```

```
            FileStatus[] stats = fs.listStatus(new Path("hdfs://192.168.11.189:9000/root/cvtree/input"));
```

```
            String[] filename = new String[stats.length];
```

```
            for(int i = 0; i < stats.length; ++i)
```

```
            {
```

```
                filename[i]=stats[i].getPath().getName();
```

```
        hdfsInStream = fs.open(stats[i].getPath());

        byte[] buffer = new byte[Integer.parseInt(String.valueOf(stats[i].getLen()))];

        hdfsInStream.readFully(0, buffer);

        String wholeContent = new String(buffer);

        pairsforpreprocess.put(filename[i],wholeContent);

        fs.close();

    }

    } catch (IOException e) {

        e.printStackTrace();

    }

    return pairsforpreprocess;

}

}
```

[0099] 大规模数据集交叉比较计算系统执行包括步骤3)-8)：

[0100] 3) 在分布式计算环境的每个计算结点部署此大规模数据集交叉比较计算后端处理框架,包括在各计算结点配置分布式文件系统及分布式机群资源管理系统

[0101] 4) 启动后端大规模数据集交叉比较计算后端分布式处理框架,并配置系统参数

[0102] 5) 将用户编写的类文件上传至后端分布式处理框架

[0103] 6) 将待处理RNA基因序列上传至分布式文件系统

[0104] 7) 分别在主结点及各从结点启动作业调试器及任务执行器,开始计算过程

[0105] 8) 计算完成,得到最终计算结果

[0106] 最终得到计算结果如下所示,输出结果格式如下所示:

Key:[AcMNPV.faa, ASFV.faa] Value:0.0016810058966743133

Key:[AsGV.faa, AdhoNPV.faa] Value:0.011683080339163805

[0107] Key:[AsGV.faa, AcMNPV.faa] Value:0.010803612571303491

Key:[ASFV.faa, AdhoNPV.faa] Value:0.0017391175577363247。

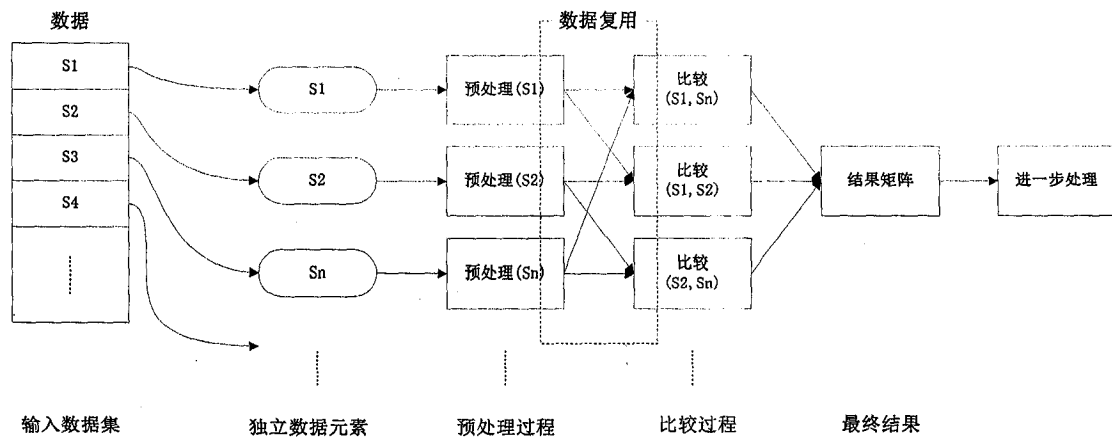


图1

	A1	A2	A3	A4	A5	...	...	An
B1	*	*	*	*	*	*	*	*
B2	*	*	*	*	*	*	*	*
B3	*	*	*	*	*	*	*	*
B4	*	*	*	*	*	*	*	*
B5	*	*	*	*	*	*	*	*
...	*	*	*	*	*	*	*	*
...	*	*	*	*	*	*	*	*
Bm	*	*	*	*	*	*	*	*

图2

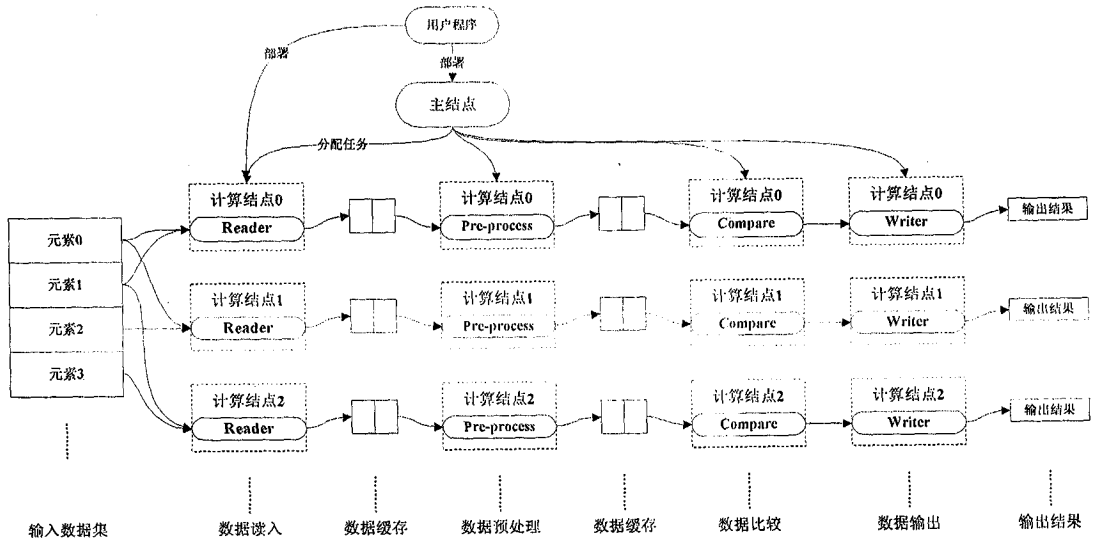


图3

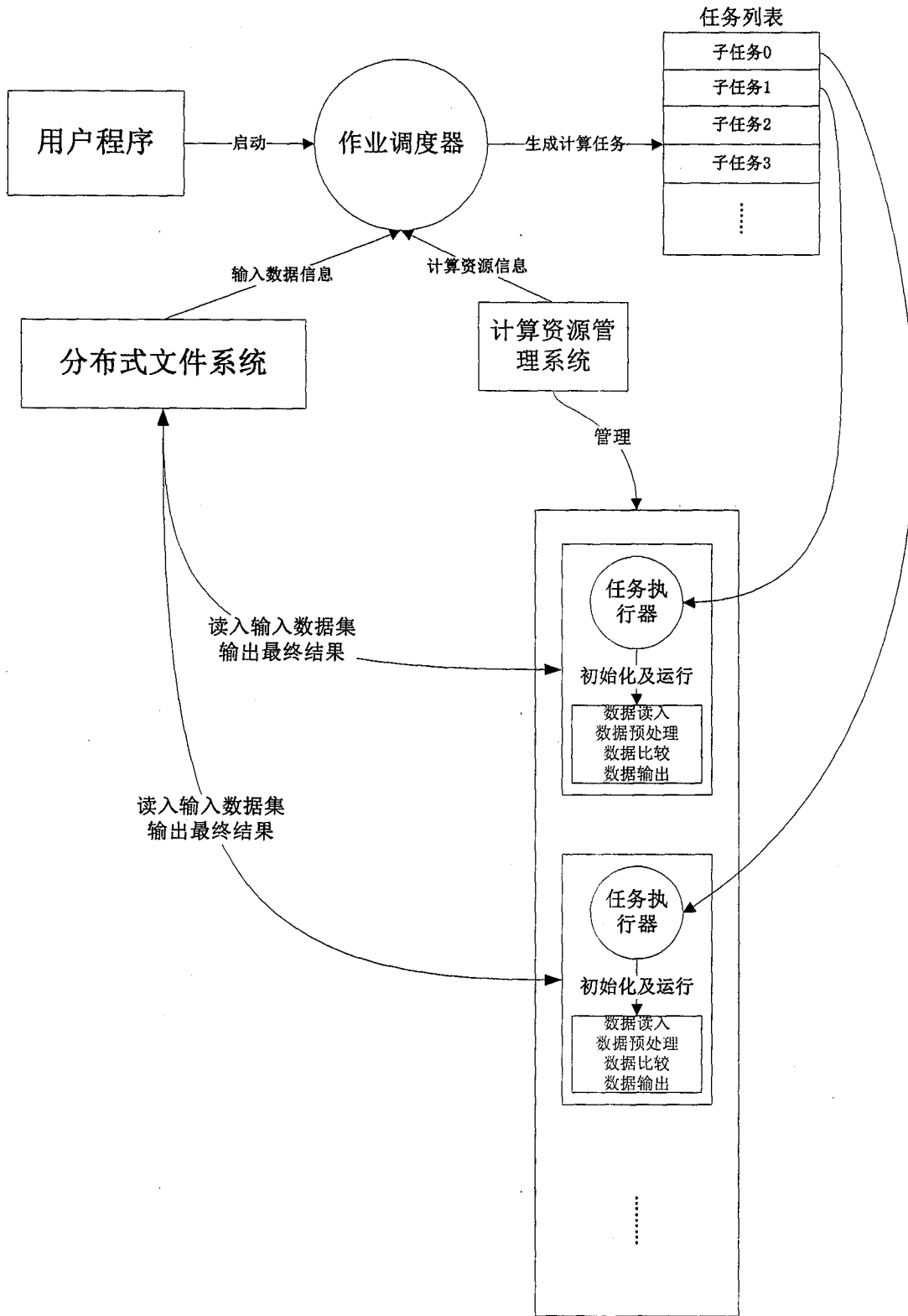


图4

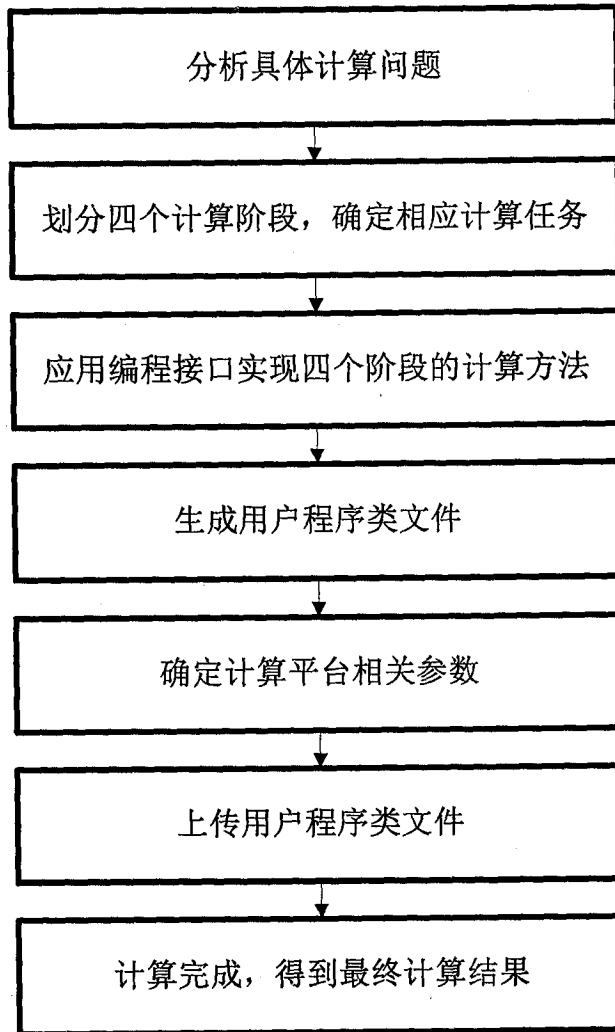


图5