(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101799809 A
(43)申请公布日 2010.08.11

(21)申请号 200910077661.3

(22)申请日 2009.02.10

(71)申请人 中国移动通信集团公司
地址 100032 北京市西城区金融大街 29 号

(72)发明人 徐萌 邓超 高丹 罗治国
周文辉 郑诗豪 沈亚飞 陈磊

(74)专利代理机构 北京同达信恒知识产权代理有限公司 11291
代理人 魏杉

(51)Int.Cl.
G06F 17/30(2006.01)

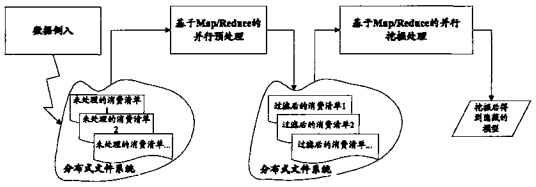
权利要求书 2 页 说明书 7 页 附图 3 页

(54)发明名称

数据挖掘方法和数据挖掘系统

(57)摘要

本发明公开了数据挖掘方法及数据挖掘系统，本发明方法包括：设置数据挖掘的工作流，所述工作流中包括多个并行的数据处理任务；启动所述工作流，并在所述多个并行的数据处理任务被触发时，为其中的每个数据处理任务分配执行节点，以使所述多个并行的数据处理任务在分配的执行节点上并行执行；以及，所述执行节点在执行每个数据处理任务时，通过 Map/Reduce 机制将数据处理任务分配给并行执行的 Map 任务进行处理，将该数据处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据处理任务的处理结果。采用本发明，可提高数据挖掘效率。



CN 101799809 A

1. 一种数据挖掘方法，其特征在于，包括：

设置数据挖掘的工作流，所述工作流中包括多个并行的数据处理任务；

启动所述工作流，并在所述多个并行的数据处理任务被触发时，为其中的每个数据处理任务分配执行节点，以使所述多个并行的数据处理任务在分配的执行节点上并行执行；以及

所述执行节点在执行每个数据处理任务时，通过映射 Map/ 简化 Reduce 机制将数据处理任务分配给并行执行的 Map 任务进行处理，将该数据处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据处理任务的处理结果。

2. 如权利要求 1 所述的数据挖掘方法，其特征在于，设置所述工作流时，还包括：根据用户输入的信息为所述数据处理任务中的部分或全部数据处理任务分别设置执行节点的数量；

所述为数据处理任务分配执行节点，具体为：根据所述工作流模块设置的执行节点数量，为设置有执行节点数量的数据处理任务分配相应数量的执行节点；

或者，所述为数据处理任务分配执行节点，具体为：根据各数据处理任务的数据处理量为各数据处理任务分配执行节点，其中，为数据处理量大的数据处理任务分配的节点数量大于为数据处理量小的数据处理任务分配的节点数量。

3. 如权利要求 2 所述的方法，其特征在于，为设置有执行节点数量的数据处理任务分配的相应数量的执行节点为当前可用执行节点中负载轻的执行节点。

4. 如权利要求 2 所述的方法，其特征在于，根据各数据处理任务的数据处理量为各数据处理任务分配执行节点，具体为：根据各数据处理任务的数据处理量的比例，为各数据处理任务分配相同比例的执行节点。

5. 如权利要求 1 所述的数据挖掘方法，其特征在于，当所述工作流中的数据处理任务被触发时，从所述工作流获取该数据处理任务的输入数据的存储位置，并将获取到的存储位置告知相应的执行节点，以使执行节点根据该存储位置获取相应数据处理任务的输入数据。

6. 如权利要求 1 至 5 任一项所述的数据挖掘方法，其特征在于，所述数据处理任务包括：数据预处理任务或 / 和挖掘算法实现处理任务。

7. 一种数据挖掘系统，其特征在于，包括：

工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的数据预处理任务；

数据预处理模块，用于当所述工作流中的所述多个并行的数据预处理任务被触发时，为其中的每个数据预处理任务分配执行节点，以使所述多个并行的数据预处理任务在分配的执行节点上并行执行，并且在执行每个数据预处理任务时，通过 Map/Reduce 机制将数据预处理任务分配给并行执行的 Map 任务进行处理，将该数据预处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据预处理任务的处理结果。

8. 如权利要求 7 所述的数据挖掘系统，其特征在于，所述工作流模块进一步用于，在设置工作流时，根据用户输入的信息为所述数据预处理任务中的部分或全部处理任务分别设置执行节点的数量；

所述数据预处理模块进一步用于，根据所述工作流模块设置的执行节点数量，为设置

有执行节点数量的数据预处理任务分配相应数量的执行节点。

9. 如权利要求 7 所述的数据挖掘系统，其特征在于，所述数据预处理模块进一步用于，根据所述多个并行的数据预处理任务中各处理任务的数据处理量，为各数据预处理任务分配执行节点，其中，为数据处理量大的处理任务分配的节点数量大于为数据处理量小的处理任务分配的节点数量。

10. 如权利要求 7 所述的数据挖掘系统，其特征在于，当所述工作流中的处理任务被触发时，从所述工作流获取该处理任务的输入数据的存储位置，并将获取到的存储位置告知相应的执行节点，以使执行节点根据该存储位置获取该处理任务的输入数据。

11. 一种数据挖掘系统，其特征在于，包括：

工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的挖掘算法实现处理任务；

挖掘算法实现模块，用于当所述工作流中的所述多个并行的挖掘算法实现处理任务被触发时，为其中的每个挖掘算法实现处理任务分配执行节点，以使所述多个并行的挖掘算法实现处理任务在分配的执行节点上并行执行，并且在执行每个挖掘算法实现处理任务时，通过 Map/Rkduce 机制将挖掘算法实现处理任务分配给并行执行的 Map 任务进行处理，将该挖掘算法实现处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应挖掘算法实现处理任务的处理结果。

12. 如权利要求 11 所述的数据挖掘系统，其特征在于，所述工作流模块进一步用于，在设置工作流时，根据用户输入的信息为所述挖掘算法实现处理任务中的部分或全部处理任务分别设置执行节点的数量；

所述挖掘算法实现模块进一步用于，根据所述工作流模块设置的执行节点数量，为设置有执行节点数量的挖掘算法实现处理任务分配相应数量的执行节点。

13. 如权利要求 11 所述的数据挖掘系统，其特征在于，所述挖掘算法实现模块进一步用于，根据所述多个并行的挖掘算法实现处理任务中各处理任务的数据处理量，为各挖掘算法实现处理任务分配执行节点，其中，为数据处理量大的处理任务分配的节点数量大于为数据处理量小的处理任务分配的节点数量。

14. 如权利要求 11 所述的数据挖掘系统，其特征在于，当所述工作流中的处理任务被触发时，从所述工作流获取该处理任务的输入数据的存储位置，并将获取到的存储位置告知相应的执行节点，以使执行节点根据该存储位置获取该处理任务的输入数据。

15. 一种数据挖掘系统，其特征在于，包括：工作流模块，以及如权利要求 7 至 10 任一项所述的数据预处理模块和如权利要求 11 至 14 任一项所述的挖掘算法实现模块；

所述工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的数据预处理任务，以及多个并行的挖掘算法实现处理任务。

数据挖掘方法和数据挖掘系统

技术领域

[0001] 本发明涉及通信领域中的数据挖掘技术，尤其涉及数据挖掘方法和数据挖掘系统。

背景技术

[0002] 数据挖掘(data mining)是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道但又是潜在有用的信息和知识的过程。

[0003] 数据挖掘应用的领域很广泛，在如银行、电信、保险、交通、零售等商业领域都有着广泛的应用。数据挖掘所能解决的典型商业问题包括：数据库营销(Database Marketing)、客户群体划分(Customer Segmentation &Classification)、背景分析(Profile Analysis)、交叉销售(Cross-selling)等市场分析行为，以及客户流失性分析(Churn Analysis)、客户信用记分(Credit Scoring)、欺诈发现(Fraud Detection)等等。

[0004] 数据挖掘流程通常包括：数据预处理(ETL)、数据挖掘算法实现、结果展示三个主要步骤。通过ETL步骤，可对源数据进行预处理以得到待挖掘数据；通过数据挖掘算法实现步骤，可实现满足业务需要的数据挖掘算法得出分析结果；通过结果展示步骤，可将数据挖掘算法的处理结果展示给用户。

[0005] 现有的数据挖掘流程采用单机节点上的串行方式实现。单机节点的数据挖掘系统，其可挖掘的数据量及算法的负载度，依赖于单个执行节点的性能。数据挖掘通常要对海量数据进行处理，而现有的数据挖掘系统由于采用单机节点上的串行机制因而性能较低，仅能支持少量数据的挖掘，无法进行较大范围的数据挖掘处理。考虑到现有数据挖掘系统的性能较低，目前的大部分ETL操作在进入数据挖掘流程之前在数据库基础上完成，由于没有在数据挖掘系统中进行完整的数据挖掘流程，增加了数据库数据导出、存储，以及存储的数据导入到数据挖掘系统的相关数据输入/输出操作，因而操作比较复杂。对于数据挖掘算法实现步骤，同样会由于单机节点上的串行方式的性能原因，导致其能够处理的数据量受到限制，不能满足海量数据处理的需求。

[0006] 一种改进方式是采用基于小型机及磁盘阵列实现单节点平台进行数据挖掘，该方法可一定程度上提高数据量的数据挖掘性能，增加可处理的数据量，但成本高、软硬件相对封闭、对厂商依赖性强，而且该方法依然采用串行的数据挖掘机制，因而其性能仍然难以较大提高。

[0007] 随着行业用户规模的迅速增大，数据挖掘所面临的数据量越来越大，而现有的数据挖掘方法受限于单节点的处理能力以及串行方式的限制，这就导致现有传统的单节点串行方式的数据挖掘系统的处理效率低，不能满足海量数据处理的需求。

发明内容

[0008] 本发明实施例提供的数据挖掘方法和数据挖掘系统，以解决现有技术中采用单节点串行方式进行数据挖掘所导致的处理效率低的问题。

[0009] 本发明的一个实施例提供的数据挖掘方法，包括：

[0010] 设置数据挖掘的工作流，所述工作流中包括多个并行的数据处理任务；

[0011] 启动所述工作流，并在所述多个并行的数据处理任务被触发时，为其中的每个数据处理任务分配执行节点，以使所述多个并行的数据处理任务在分配的执行节点上并行执行；以及

[0012] 所述执行节点在执行每个数据处理任务时，通过 Map/Reduce 机制将数据处理任务分配给并行执行的 Map 任务进行处理，将该数据处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据处理任务的处理结果。

[0013] 本发明的另一实施例提供的数据挖掘系统，包括：

[0014] 工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的数据预处理任务；

[0015] 数据预处理模块，用于当所述工作流中的所述多个并行的数据预处理任务被触发时，为其中的每个数据预处理任务分配执行节点，以使所述多个并行的数据预处理任务在分配的执行节点上并行执行，并且在执行每个数据预处理任务时，通过 Map/Reduce 机制将数据预处理任务分配给并行执行的 Map 任务进行处理，将该数据预处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据预处理任务的处理结果。

[0016] 本发明的另一实施例提供的数据挖掘系统，包括：

[0017] 工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的挖掘算法实现处理任务；

[0018] 挖掘算法实现模块，用于当所述工作流中的所述多个并行的挖掘算法实现处理任务被触发时，为其中的每个挖掘算法实现处理任务分配执行节点，以使所述多个并行的挖掘算法实现处理任务在分配的执行节点上并行执行，并且在执行每个挖掘算法实现处理任务时，通过 Map/Reduce 机制将挖掘算法实现处理任务分配给并行执行的 Map 任务进行处理，将该挖掘算法实现处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应挖掘算法实现处理任务的处理结果。

[0019] 本发明的另一实施例提供的数据挖掘系统，包括：工作流模块，以及前述数据挖掘系统中的数据预处理模块和前述数据挖掘系统中的挖掘算法实现模块；

[0020] 所述工作流模块，用于设置数据挖掘的工作流，所述工作流中包括多个并行的数据预处理任务，以及多个并行的挖掘算法实现处理任务。

[0021] 本发明的上述实施例，通过设置包含多个并行执行的数据处理任务的工作流，使这些数据处理任务在被触发后由分配到的执行节点进行并行处理，并且在处理每个数据处理任务时，采用 Map/Reduce 机制，将数据处理任务分配给并行执行的 Map 任务进行处理，将该数据处理任务对应的各 Map 任务的处理结果通过相应的 Reduce 任务进行合并处理得到相应数据处理任务的处理结果，一方面，使得多个数据处理任务可并行执行，另一方面，每个数据处理任务也可通过多个 Map 任务并行执行的方式实现，从而实现了多点并行的处理方式，与现有技术的单节点串行方式相比，可提高数据挖掘效率。

附图说明

[0022] 图 1 为本发明实施例中的数据挖掘系统的架构示意图；

[0023] 图 2 为本发明实施例中的数据挖掘系统的功能结构示意图；

[0024] 图 3a、图 3b 为本发明实施例中的工作流示意图；

[0025] 图 4 为本发明实施例中的数据挖掘流程示意图；

[0026] 图 5 为本发明实施例中并行 ETL 时的执行节点数与 ETL 加速比的示意图；

[0027] 图 6 为本发明实施例中并行 K-means 时的执行节点数与 K-means 加速比的示意图。

具体实施方式

[0028] 下面结合附图对本发明实施例进行详细描述。

[0029] 参见图 1，为本发明实施例中的数据挖掘系统架构示意图，该数据挖掘系统可划分为 3 层：业务应用层 1、数据挖掘平台层 2 和分布式计算平台层 3。

[0030] 其中：

[0031] 数据挖掘平台层 2 是数据挖掘系统实现业务数据挖掘的关键层，可通过该层提供的工作流设置、数据加载、数据预处理、挖掘算法实现和结果显示等功能实现数据挖掘流程；

[0032] 业务应用层 1 提供 GUI（用户界面接口）和数据挖掘算法库的 API（应用程序接口）。数据挖掘平台层 2 可调用 GUI 实现对数据挖掘流程的图形化设置、控制和任务提交，以及对数据挖掘结果的图形化显示；数据挖掘平台层 2 中的上述各功能可通过调用相应的 API 实现，例如：通过调用预处理函数 API 实现数据预处理功能，通过调用挖掘算法函数 API 实现数据挖掘算法实现功能；

[0033] 分布式计算平台层 3 可包括分布式文件系统，以提供分布式数据文件存储与管理功能，可对数据挖掘平台层 2 输出的数据挖掘处理流程的中间数据和结果数据进行输入 / 输出以及存储管理。该层还可进一步包括并行编程环境，以提供基于 Map/Reduce（映射 / 简化）的编程模型，以及任务调度、任务执行和结果反馈等功能，从而可为数据挖掘平台层 2 中的上述各功能提供实现基础，也可为用户在数据挖掘平台层 2 中增加数据挖掘处理功能提供实现基础，从而提高了系统的灵活性和可扩展性。

[0034] 参见图 2，为本发明实施例中的数据挖掘系统的功能结构示意图，该数据挖掘系统包括：工作流模块 21、数据预处理模块 22、挖掘算法实现模块 23，还可进一步包括结果显示模块 24，其中：

[0035] 工作流模块 21，用于设置数据挖掘的工作流，即，设置数据挖掘过程中各处理环节（如数据预处理环节、挖掘算法实现环节、结果显示环节）的先后顺序和各处理环节的衔接关系，负责向与处理环节相对应的功能模块发送启动信号以及各功能模块运行所需的其他控制信息。

[0036] 工作流模块 21 可为用户提供 GUI 形式的工作流设置界面，该工作流界面可提供数据挖掘流程中各处理环节的各种处理任务的图标，用户可通过拖拽处理任务图标到工作流设置界面上，并通过排列各种任务图标的先后顺序和衔接关系以及串并方式来设置数据挖掘的工作流。设置的工作流可以是一个，也可以是多个独立的工作流。设置工作流时，用户还可以通过提供的设置界面输入设置参数为工作流上的处理任务设置该处理任务执行过

程所需的控制信息，输入的设置参数或控制信息可包括：输入数据（或称源数据）的存储路径或 / 和输出数据的存储路径，还可以包括执行该任务的执行节点（如 Map 任务数、Reduce 任务数）个数或 / 和执行节点的标识。如果用户在设置工作流时没有为处理任务设置控制信息，则工作流模块 21 可为其设置默认的控制信息。在启动工作流后，由工作流来控制各处理任务的执行。在数据挖掘过程中，每执行完成工作流上的一个处理任务，都会向工作流模块 21 返回处理结果数据（即该处理任务的输出数据）的位置信息，工作流模块 21 在根据设置的工作流触发下一个处理任务时，会将该数据的位置信息传递给被触发的处理任务，使被触发的处理任务根据该位置信息获取输入数据。

[0037] 通过工作流模块 21 可设置包含多个处理任务并行的工作流。对于 ETL 处理环节，可设置多个并行的 ETL 处理任务，这些并行的 ETL 处理任务中，可以是针对不同源数据的处理任务，这些处理任务可以是执行相同类型操作的处理任务，也可以是执行不同类型操作的处理任务。例如，如图 3a 所示的工作流上包括 3 个 ETL 处理任务，并分别以 31、32 和 33 标识，其中，任务 31 是对数据表 1 的数据进行属性删除操作，任务 32 是对数据表 2 的数据进行属性增加操作，任务 33 是对任务 31 和任务 32 的处理结果进行整合操作，任务 31 和 32 并行执行。同理，对于挖掘算法实现环节，也可以按照类似的方式设置并行执行的任务。例如，如图 3b 所示，可对通过任务 31 和 32 处理后的数据表 1 和数据表 2，分别通过并行的挖掘算法实现处理任务 34 和 35 进行处理，并将处理结果通过结果显示处理任务 35 进行显示。

[0038] 数据预处理模块 22，用于根据工作流模块 21 设置的工作流执行 ETL 处理操作。当启动数据挖掘工作流后以及在数据挖掘过程中，数据预处理模块 22 可对被触发的处理任务进行资源调度，根据工作流中是否设置有处理任务的控制信息以及设置的控制信息类型，调度过程可以有以下几种方式：

[0039] 方式一：设置工作流时为处理任务指定了执行节点个数和标识，则数据预处理模块 22 按照指定的执行节点个数和标识为处理任务分配相应的执行节点。

[0040] 方式二：设置工作流时为处理任务指定了执行节点个数，则数据预处理模块 22 按照指定的执行节点个数为处理任务分配相应数量的执行节点，较佳地，数据预处理模块 22 可根据当前各执行节点的负载情况，为处理任务分配负载较轻的执行节点。

[0041] 方式三：设置工作流时没有为处理任务分配执行节点数量和标识，则数据预处理模块 22 根据为处理任务默认分配的执行节点数量或 / 和标识为处理任务分配执行节点，如：为并行的每个处理任务分别分配一个执行节点。在默认分配了执行节点数量而未指定执行节点标识的情况下，数据预处理模块 22 可根据当前各执行节点的负载情况，为处理任务分配负载较轻的执行节点。

[0042] 方式四：设置工作流时没有为处理任务分配执行节点数量和标识，则数据预处理模块 22 根据资源分配策略为处理任务分配执行节点。资源分配策略可以是：根据各处理任务需要处理的数据量，为数据量大的处理任务分配较多的执行节点，为数据量小的处理任务分配较少的执行节点，较佳地，可按照各处理任务处理的数据量的比例，分配相应比例的执行节点。例如：对于 2 个并行且分别处理数据表 1 和数据表 2 的 ETL 处理任务，数据表 1 的大小是数据表 2 的 3 倍，则分配给处理数据表 1 的任务的执行节点数量是处理数据表 2 的任务的 3 倍，如，可以分配当前可用的总执行节点个数的 3/4 的节点给数据表 1，另外 1/4

给数据表 2。

[0043] 挖掘算法实现模块 23，用于根据工作流模块 21 设置的工作流执行挖掘算法处理操作。当启动数据挖掘工作流后以及在数据挖掘过程中，挖掘算法实现模块 23 可对被触发的处理任务进行资源调度，根据工作流中是否设置有处理任务的控制信息以及设置的控制信息类型，调度过程可以与数据预处理模块 22 的资源调度过程类似，在此不再赘述。

[0044] 结果显示模块 24，用于根据工作流模块 21 设置的工作流，对显示任务所指示的处理结果进行展示，可调用 GUI 界面进行结果展示。

[0045] 本发明的上述实施例中，并行执行的数据预处理操作和并行执行的挖掘算法实现操作可采用 Map/Reduce 机制实现。Map/Reduce 是一种分布式处理海量数据的实现方式，该机制可让程序分步到一个由普通节点组成的超大集群上并发执行。根据 Map/Reduce 机制，对于并行执行的各处理任务中的每个处理任务，可通过调用 Map 函数，将每个处理任务由多个 Map 任务并行处理，这些 Map 任务被分配到为所属处理任务分配的执行节点上执行，再通过调用 Reduce 函数，分别对每个处理任务的各 Map 任务的处理结果进行合并等操作。这样，多个数据预处理任务之间或多个挖掘算法实现处理任务之间可以并行执行，并且每个数据预处理任务或挖掘算法实现处理任务内部也可并行执行，从而提高了数据挖掘系统的处理效率。

[0046] 上述实施例中，由工作流设置的并行处理任务还可以包括不同处理环节的处理任务，如，2 个并行的处理任务分别是对数据表 A 进行 ETL 处理的任务和对数据表 B 进行挖掘算法实现的处理任务；又如，4 个并行的处理任务分别是对数据表 A 进行 ETL 处理的任务、对数据表 B 进行 ETL 处理的任务、对数据表 C 进行挖掘算法实现的处理任务和对数据表 D 进行挖掘算法实现的处理任务。这种情况下，对各处理任务的处理过程与前述的处理过程类似。通过上述描述可以看出，通过设置包含有多个并行的处理任务，并且每个处理任务都采用 Map/Reduce 机制进行并行处理，使多个处理任务可并行执行，并且每个处理任务也是并行实现，从而提高了数据挖掘的效率。

[0047] 根据本发明实施例提供的上述数据挖掘系统，一种数据挖掘流程可包括：

[0048] 设置数据挖掘的工作流，该工作流中包括多个并行的 ETL 处理任务和多个并行的挖掘算法实现处理任务；

[0049] 启动工作流；

[0050] 当该工作流上的多个并行的 ETL 处理任务被触发时，为这些被触发的处理任务分配执行节点，执行节点分配过程可参照前述方式，各执行节点对于分配到的处理任务，根据为该任务指定的输入数据位置读取数据，并对读取的数据进行相应的 ETL 处理操作。该过程中，各执行节点的处理是并行进行的；

[0051] 同理，当该工作流上的多个并行的挖掘算法实现处理任务被触发时，为这些被触发的处理任务分配执行节点，执行节点分配过程可参照前述方式，各执行节点对于分配到的处理任务，根据为该任务指定的输入数据位置读取数据，并对读取的数据进行相应的挖掘算法实现处理操作。该过程中，各执行节点的处理是并行进行的。

[0052] 图 4 给出了一种数据挖掘的流程示意图，在这个流程中数据挖掘工作包括三个步骤：数据加载、并行预处理和并行数据挖掘。并行预处理过程负责清洗、过滤原始 CDR（呼叫详细记录）数据（如图 4 中的消费清单），并生成高质量的数据供数据挖掘算法使用，例如

属性选择、统计、归一化等。并行数据挖掘过程负责接收经过预处理的数据作为训练集，挖掘潜在的模型，例如聚类（如使用K-means算法实现）、分类、关联规则、社会关系网分析等。

[0053] 在本发明的另一实施例所提供的数据挖掘系统中，包括上述的工作流模块21和数据预处理模块22，还可进一步包括结果显示模块24，其数据挖掘过程与上述数据挖掘过程类似，只是没有上述流程中的挖掘算法实现操作。

[0054] 在本发明的另一实施例所提供的数据挖掘系统中，包括上述的工作流模块21和挖掘算法实现模块23，还可进一步包括结果显示模块24，其数据挖掘过程与上述数据挖掘过程类似，只是没有上述流程中的数据预处理操作。

[0055] 本发明实施例中所述的数据预处理操作可包括：属性增加、属性删除、属性位置交换、添加ID属性、多表合并（如2个数据表的join操作）、属性规约、数据冗余处理、数据抽样、数据噪声处理等。

[0056] 本发明实施例中所述的挖掘算法实现操作可包括：聚类（如使用K-means算法实现）处理、数据分类、关联规则挖掘等。

[0057] 需要指出的是，本发明上述实施例提供的数据挖掘系统中采用的方法和传统数据挖掘方法不一样。传统数据挖掘系统通常会先将本地待挖掘数据从文件中全部读入到内存进行处理，而本发明实施例提供的数据挖掘系统中，并行预处理和并行挖掘算法实现过程所需要的海量数据文件被存储在集群系统中并使用DFS（分布式文件系统）进行管理，而不用先全部读入到内存。按照Map/Reduce机制，并行预处理和并行数据挖掘算法实现过程使用按行顺序读写的方式从DFS中接收数据和输出数据。

[0058] 以下是本发明实施例所提供的数据挖掘方法与现有数据挖掘方法的性能比较，实现环境包括：58个PC节点，每个PC节点的硬件环境为4核CPU、8GB内存、1T硬盘、1GB网络适配器。其中2个PC节点各作为NameNode（负责和程序通信，管理文件系统的元数据或修饰属性）和JobTracker（分配工作以及负责和用户程序通信，是MapReduce的总调度），其他节点作为DataNode（存储实际数据）和TaskTracker（负责执行任务）。实验使用的Hadoop版本是hadoop-0.17.1，blocksize设为64MB且副本数设为2。

[0059] 对于ETL，表1给出了在数据量为300G、数据类型为电信领域数据（包括通话费用和通话信息）、设定Map任务数为50、Reduce任务数为1的情况下，来比较不同DataNode数量情况下加速比的性能，结果如表1所示，对比效果图如图5所示。

[0060] 表1

[0061]

节点数	MapReduce 时间				实际缩放比例	理论缩放比例
	小时	分钟	秒	总计(秒)		
8	3	1	10	10870	1	1
16	1	31	49	5509	1.97	2
24	1	1	51	3711	2.93	3
32	0	48	4	2884	3.77	4
64	0	23	21	1401	7.78	8

[0062] 从表1可以看出，加速比随着DataNode节点数的增加接近线性增长。通过图5中描述的随着节点数增加的加速比性能情况，可以看出当节点数增加时加速比接近线性。试

验结果表明基于 Map/Reduce 的并行 ETL 是支持电信领域数据挖掘应用的有效方法，而且基于 Map/Reduce 的方法可以被应用于解决其他领域面临的海量数据量的 ETL 处理问题。

[0063] 对于数据挖掘处理过程，表 2 对比了基于 Map/Reduce 的并行 k-means 算法在客户职业细分聚类上的时间性能，给出了对于经过并行 ETL 过滤后的 3G 数据量的数据，不同节点数进行 k-means 迭代的运行时间。图 6 给出了不同 DataNode 节点数上 K-means 加速比。实验设定的 Reduce 任务数和聚类分成的组数 k 相同，理论上 k 值越大应该有更多平行的 Reduce 任务，速度会更快。本实现选择了设定 k 为最小值 2、Reduce 任务数为 2、Map 任务数为 50 的情况，且设定 k 个聚类中心的初始值为固定值。

[0064] 表 2

[0065]

节点数	MapReduce 时间			实际缩放比例	理论缩放比例
	分钟	秒	总计(秒)		
8	53	5	3185	1	1
16	27	59	1679	1.90	2
24	18	47	1127	2.83	3
32	14	24	864	3.69	4
64	9	15	555	5.74	8

[0066] 通过图 6 描述的随着节点数增加的加速比性能情况，可以看出当节点数增加时加速比接近线性，但是到达 64 个节点时候，其加速比仅相当于 48 节点情况，这说明数据挖掘算法的并行化效果与处理的数据量、处理问题的复杂性等多种因素相关。实验结果显示基于 Map/Reduce 的并行 k-means 算法可以应用于分析海量数据的聚类问题，而目前商用数据挖掘算法仅支持百 MB 的数据挖掘，基于本发明实施例提供的数据挖掘系统支持的数据为商用工具的 1000 倍。

[0067] 通过以上实验可以看出，在数据处理量方面，本发明实施例提供的数据挖掘系统及其方法仅采用 64 个节点就可以以较高性能支持 300G 数据的 ETL 处理和挖掘算法实现处理，而现有的数据挖掘系统仅能支持 300M 数据的挖掘。在数据响应时间方面，实验表明处理 300G 数据的 ETL 操作在 20 分钟级，Kmeans 算法的响应时间为 10 分钟级，因而可以在有效的时间内处理海量数据。在数据挖掘的成本方面，由于本发明实施例可基于由 PC 机组成的集群实现，并且对于操作系统也无特殊要求，例如可采用开源 Linux，相比于现有的基于小型机环境的数据挖掘系统，其实现成本较低。

[0068] 显然，本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样，倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内，则本发明也意图包含这些改动和变型在内。

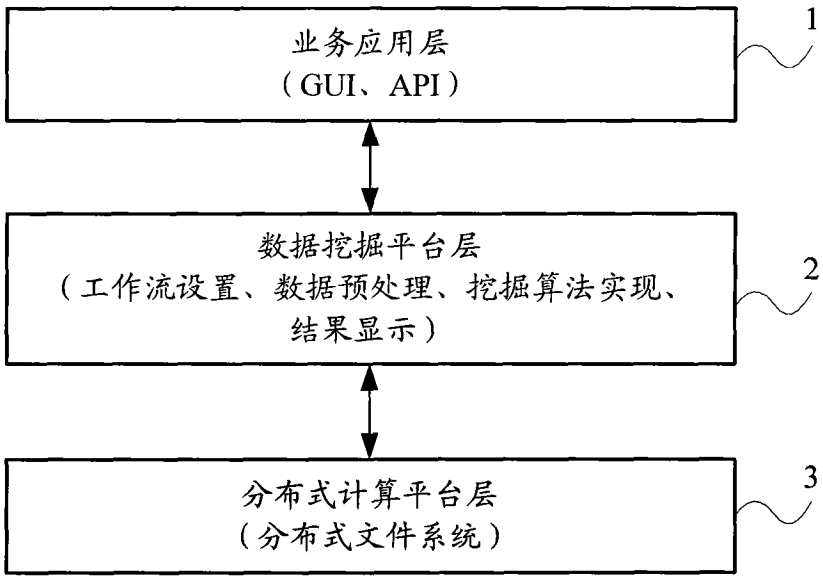

业务应用层
（GUI、API）
1
数据挖掘平台层
（工作流设置、数据预处理、挖掘算法实现、结果显示）
2
分布式计算平台层
（分布式文件系统）
3

图 1

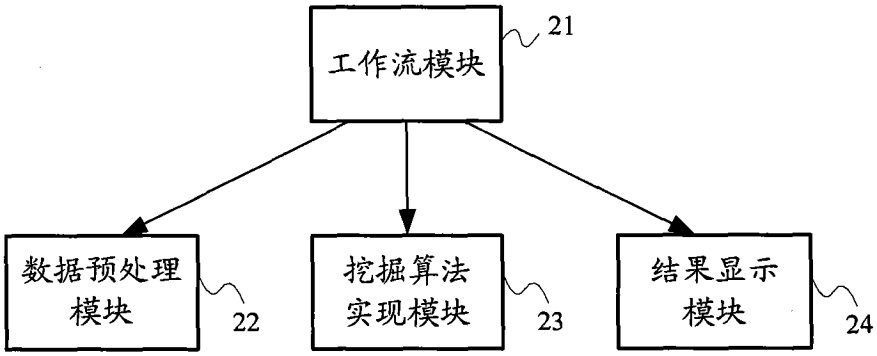

工作流模块
21
数据预处理模块
22
挖掘算法实现模块
23
结果显示模块
24

图 2

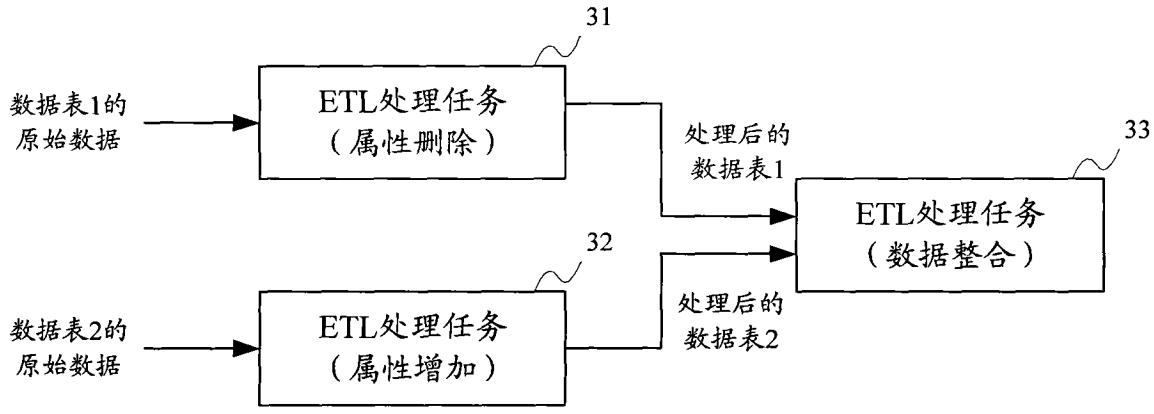

31
数据表1的原始数据
ETL处理任务
（属性删除）
处理后的数据表1
33
ETL处理任务
（数据整合）
32
数据表2的原始数据
ETL处理任务
（属性增加）
处理后的数据表2

图 3a

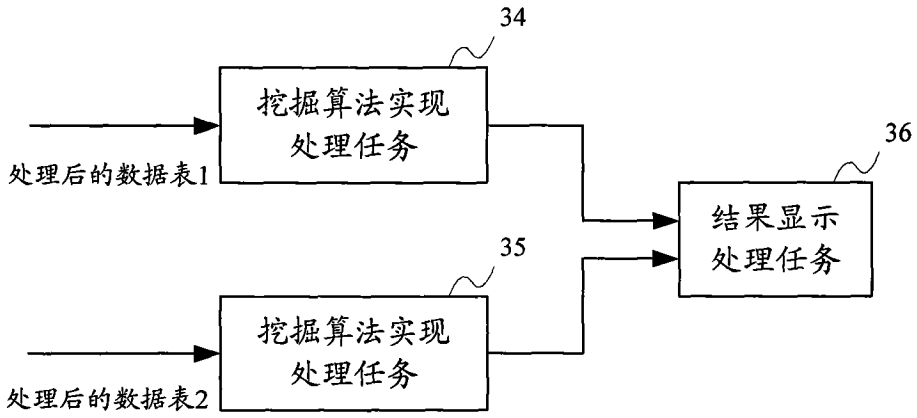

34
挖掘算法实现
处理任务
处理后的数据表1
36
结果显示
处理任务
35
挖掘算法实现
处理任务
处理后的数据表2

图 3b

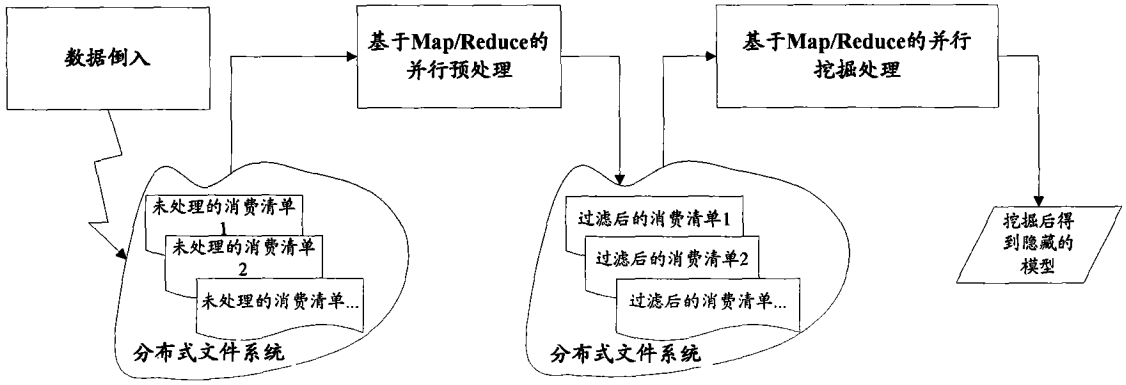

数据倒入
基于Map/Reduce的
并行预处理
基于Map/Reduce的并行
挖掘处理
未处理的消费清单
1
未处理的消费清单
2
未处理的消费清单...
分布式文件系统
过滤后的消费清单1
过滤后的消费清单2
过滤后的消费清单...
分布式文件系统
挖掘后得
到隐藏的
模型

图 4

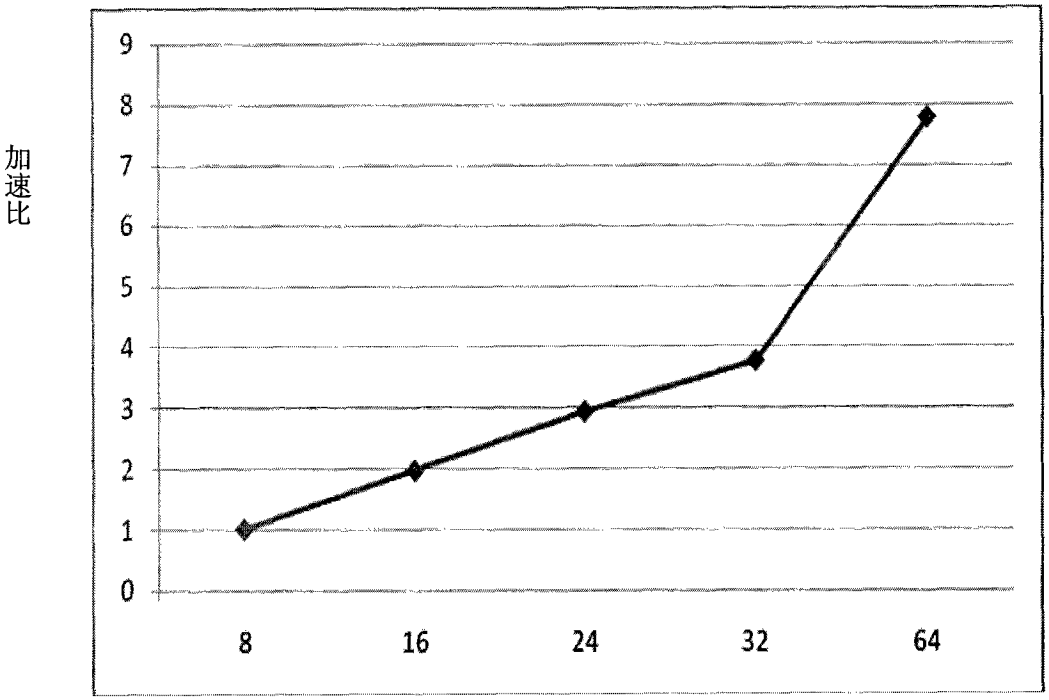

加速比
9
8
7
6
5
4
3
2
1
0
8
16
24
32
64
DataNode 数量

图 5

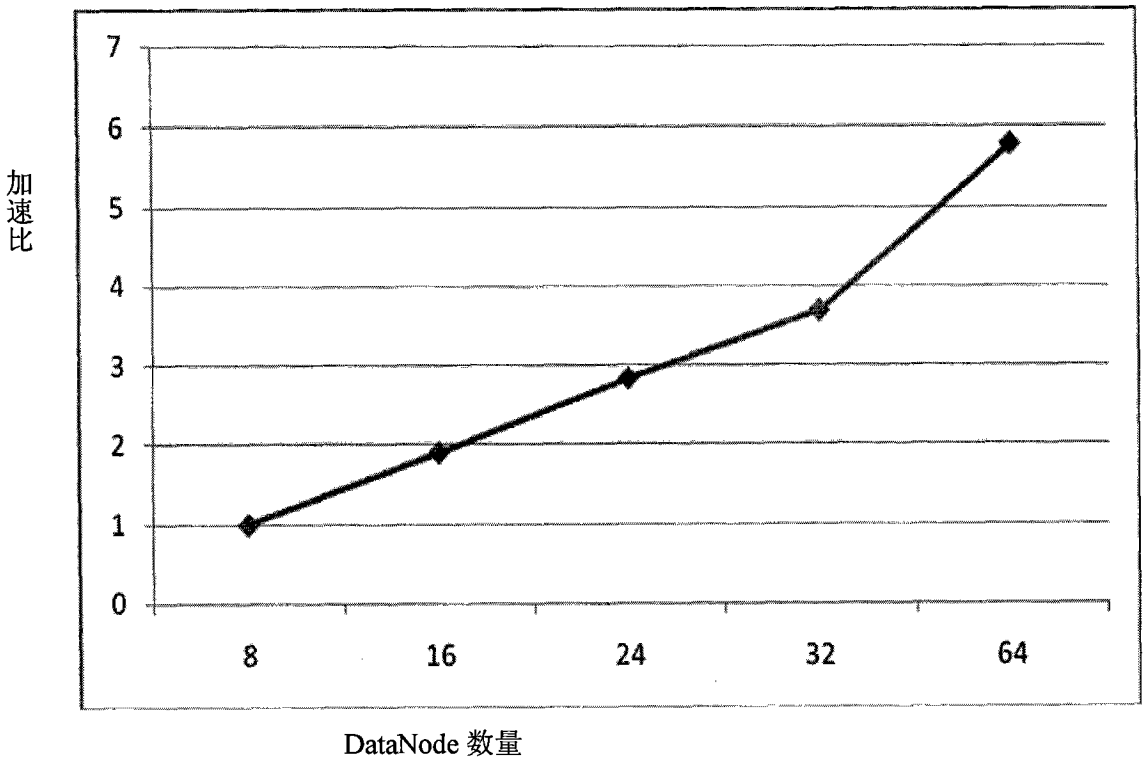

加速比
7
6
5
4
3
2
1
0
8
16
24
32
64
DataNode 数量

图 6