

## (12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织  
国际局

(43) 国际公布日  
2017 年 1 月 19 日 (19.01.2017)



(10) 国际公布号  
WO 2017/008650 A1

- (51) 国际专利分类号:  
G06F 17/30 (2006.01)
- (21) 国际申请号: PCT/CN2016/088302
- (22) 国际申请日: 2016 年 7 月 4 日 (04.07.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:  
201510408180.1 2015 年 7 月 13 日 (13.07.2015) CN
- (71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 开曼群岛英属开曼群岛大开曼资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。
- (72) 发明人: 丁崔灿 (DING, Cuican); 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。
- (74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY

AGENCY LTD.); 中国北京市金融街 35 号国际企业大厦 A 座 16 层, Beijing 100033 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

[见续页]

(54) Title: DEVICE AND METHOD FOR FILTERING DATA

(54) 发明名称: 一种用于过滤数据的设备及方法

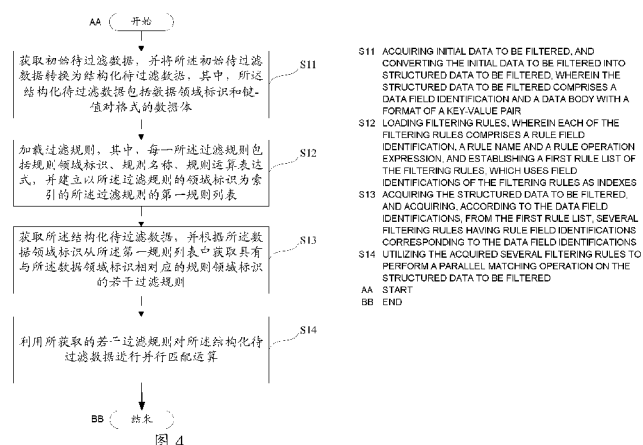


图 4

(57) Abstract: A device and method for filtering data. The method comprises: after initial data to be filtered is acquired each time, converting the initial data to be filtered into structured data to be filtered, and utilizing a corresponding filtering rule to perform a matching calculation in real time, thereby obtaining a filtering result, solving the problem of instantaneity. An arithmetical operation, a character string operation, a relational operation, a logic operation, a regular expression operation and a set operation are supported at the same time, and an expansion interface is reserved. In addition, the filtering rule is in the form of a simple operation expression with a variable, solving the problem of complex description of the filtering rule, difficult expansion and difficult management.

(57) 摘要: 一种用于过滤数据的设备及方法, 每次获取初始待过滤数据后将所述初始待过滤数据转换为结构化待过滤数据, 并利用相应的过滤规则实时进行匹配计算, 随即得到过滤结果, 解决实时性问题, 同时支持算术运算, 字符串运算, 关系运算, 逻辑运算, 正则表达式运算, 集合运算, 且预留了扩展接口, 并且所述过滤规则为简单的带变量的运算表达式形式, 解决了过滤规则描述复杂、扩展不易及管理困难的问题。

WO 2017/008650 A1

**本国际公布:**

— 包括国际检索报告(条约第 21 条(3))。

## 一种用于过滤数据的设备及方法

本申请要求 2015 年 07 月 13 日递交的申请号为 201510408180.1、发明名称为“一种用于过滤数据的设备及方法”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

### 技术领域

本申请涉及计算机领域，尤其涉及一种从海量数据中根据设定的过滤规则实时过滤出满足过滤规则的数据的技术。

### 10 背景技术

随着信息技术爆发式增长，数据量与日俱增，众多领域对海量数据的处理的要求不断增加。

针对如何从海量数据中根据设定的过滤规则过滤出满足过滤规则的数据，现有技术中有以下几种方法：

15 基于内存关系型数据库的 SQL 语句（Structured Query Language）来过滤有效数据，然而，该方法需要将海量数据缓存在内存数据库的逻辑数据表中，占用大量内存资源，且 SQL 语句的周期性执行难以达到实时性要求；

基于 Hbase（一个分布式的、面向列的开源数据库）的海量数据存储方案，使用 Map-Reduce 算法（一种编程模型算法，用于大规模数据集的并行运算）来过滤有效数据，  
20 然而，Map-Reduce 模型任务是类似于批处理的后计算模式，对已经存储在 Hbase 中的海量数据，只能周期性的执行运算匹配结果，实时性难以得到保障，并且复杂的 Map-Reduce 模型任务需要通过扩展编写来实现，难以满足对大量过滤规则的实时可变和多种计算的需求；

基于 CEP 引擎（复杂事件处理，Complex Event Processing），使用模式匹配算法来  
25 过滤有效数据更适合做企业应用系统的监测和决策控制，然而成熟的 CEP 引擎大多是商业软件，用户成本高，并且 CEP 引擎都有各自的模式规则描述方法，比如 Drools 使用 XML 格式，Esper 使用 EPL 格式，针对不同的系统的需求需要编写大量的适配代码来使用，且对于非标准化的匹配算法需要扩展编写来实现，增加实现难度，此外，CEP 引擎的实现各不相同，故不易对 CEP 引擎的性能监测和调优。

## 发明内容

本申请要解决的技术问题是如何能够在不占用大量内存资源下，根据设定的过滤规则，从海量数据中实时地过滤出满足过滤规则，并且能够满足大量过滤规则的实时可变  
5 和多种计算的需求。

为实现上述目的，本申请提供了一种用于过滤数据的方法，其中，所述方法包括：

获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，  
所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；

加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称及规则运算  
10 表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表；

获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获得  
具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；

利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

进一步地，所述获取初始待过滤数据包括：

15 从分布式消息中间件中获取所述初始待过滤数据。

进一步地，将所述初始待过滤数据转换为结构化待过滤数据还包括：

将所述结构化待过滤数据发送至阻塞队列；

获取所述结构化待过滤数据包括：

从所述阻塞队列中获取所述结构化待过滤数据。

20 进一步地，利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算  
包括：

对所获取的过滤规则进行规则编译，以建立可运行抽象语法树；

将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，  
并利用若干所述可运行抽象语法树进行并行匹配计算。

25 进一步地，对所获取的过滤规则进行规则编译，以建立可运行抽象语法树包括：

对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树；

对所述抽象语法树进行预计算，以获得所述可运行抽象语法树；

其中，对所述抽象语法树进行一次预计算包括：

根据所述抽象语法树创建运行堆栈，将所述抽象语法树中的元素传入所述运行堆栈  
30 中；

当所述元素为操作符时，将所述操作符对应的两个操作数传出所述运行堆栈，计算以获得计算结果；

当所述元素为特殊元素，则将所述特殊元素转换为程序语言数据结构元素后传入运行堆栈中。

5 进一步地，利用若干所述可运行抽象语法树进行并行匹配计算包括：

将所述可运行抽象语法树的变量替换为所述数据体中的参数；

利用所述运行堆栈对所述可运行抽象语法树进行匹配计算。

进一步地，所述方法还包括：

新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

10 进一步地，建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表还包括：

建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则列表；

所述新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译包括至少以下任一项：

15 将新增的过滤规则增加至所述第二规则列表中；

从所述第二规则列表中删除相应的过滤规则；

从所述第二规则列表中查找过滤规则，并对所查找的过滤规则进行修改编译。

进一步地，每一所述过滤规则还包括：所述过滤规则所绑定的通知器的信息；

所述方法还包括：

20 将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的通知器，以备传输。

根据本申请另一方面还提供了一种用于过滤数据的设备，其中，所述设备包括：

第一装置，用于获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；

25 第二装置，用于加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称及规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表；

第三装置，用于获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；

30 第四装置，用于利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配

运算。

进一步地，所述第一装置包括：

从分布式消息中间件中获取所述初始待过滤数据的单元。

进一步地，所述第一装置包括：

5 用于将所述结构化待过滤数据发送至阻塞队列的单元；

所述第三装置包括：

从所述阻塞队列中获取所述结构化待过滤数据的单元。

进一步地，所述第四装置包括：

用于对所获取的过滤规则进行规则编译，以建立可运行抽象语法树的单元；

10 用于将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的单元。

进一步地，所述用于对所获取的过滤规则进行规则编译，以建立可运行抽象语法树的单元包括：

用于对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树的模块；

15 用于对所述抽象语法树进行预计算，以获得所述可运行抽象语法树的模块，其中，该模块用于：

根据所述抽象语法树创建运行堆栈，将所述抽象语法树中的元素传入所述运行堆栈中，

20 当所述元素为操作符时，将操作符对应的两个操作数传出所述运行堆栈，计算以获得计算结果，

用于当所述元素为特殊元素，则将所述特殊元素转换为程序语言数据结构元素后传入所述运行堆栈中。

进一步地，所述用于将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的单元包括：

25 用于将所述可运行抽象语法树的变量替换为所述数据体中的参数的模块；

用于利用所述运行堆栈对所述可运行抽象语法树进行匹配计算的模块。

进一步地，所述设备还包括：

第五装置，用于新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

进一步地，所述第二装置还包括：

30 建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则列表的单元；

所述第五装置包括：

用于将新增的过滤规则增加至所述第二规则列表中的单元；

用于从所述第二规则列表中删除相应的过滤规则的单元；

用于从所述第二规则列表中查找过滤规则，并对所查找的过滤规则进行修改编译的

5 单元。

进一步地，每一所述过滤规则还包括：所述过滤规则所绑定的通知器的信息；

所述设备还包括：

第六装置，用于将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的通知器，以备传输。

10 与现有技术相比，根据本申请一实施例所提供的用于数据过滤的设备及方法采用流式运算方式，不会在内存中缓存也不会固化数据，即每次获取初始待过滤数据后将所述初始待过滤数据转换为结构化待过滤数据，并利用相应的过滤规则实时进行匹配计算，随即得到过滤结果，解决海量流式数据的过滤的实时性问题；

进一步地，根据本申请一实施例所提供的用于数据过滤的设备及方法所述过滤数据  
15 的方法和设备支持算术运算，字符串运算，关系运算，逻辑运算，正则表达式运算，集合运算，且预留了扩展接口，并且所述过滤规则为简单的带变量的运算表达式形式，解决了过滤规则描述复杂、扩展不易及管理困难的问题；

此外，本申请根据本申请一实施例所提供的用于数据过滤的设备及方法为自主设计开发，成本相对较低，且在任意代码路径上都可以监控和调优。

20

#### 附图说明

通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本申请的其它特征、目的和优点将会变得更明显：

图 1 示出根据本申请一方面提供的一种用于过滤数据的设备的设备示意图；

25 图 2 示出根据本申请一优选的实施例提供的一种用于过滤数据的设备的设备示意图；

图 3 示出根据本申请另一优选的实施例提供的一种用于过滤数据的设备的设备示意图；

图 4 示出根据本申请一方面提供的一种用于过滤数据的方法流程图；

30 图 5 示出根据本申请一优选的实施例提供的一种用于过滤数据的方法流程图；

图 6 示出根据本申请另一优选的实施例提供的一种用于过滤数据的方法流程图；

图 7 示出根据本申请一优选的实施例提供的一种包括所述用于过滤数据设备的系统的设备示意图；

图 8 至图 10 示出根据本申请一具体场景中利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算的示意图。

附图中相同或相似的附图标记代表相同或相似的部件。

### 具体实施方式

下面结合附图对本申请作进一步详细描述。

图 1 示出根据本申请一方面提供的一种用于过滤数据的设备的设备示意图，其中，所述设备 1 包括：第一装置 11、第二装置 12、第三装置 13 和第四装置 14。

具体地，所述第一装置 11 用于获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；所述第二装置 12 用于加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称、规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表；所述第三装置 13 用于获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；所述第四装置 14 用于利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

进一步地，所述第一装置 11 用于获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体在此，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体。

其中，所述数据领域标识用于表明所述结构化待过滤数据的类别，其中，所述类别例如且不限于：主机的 CPU 占用率、某网站的访问延迟时间等，所述数据领域标识可以采用数据或文字等进行标识，此外，任何能够被计算机识别的标识的方式都可以作为所述数据领域标识的实施方式，并以引用的方式包含于此。其中，所述键-值对格式的数据体记录所述结构化待过滤数据的键-值对格式（Key-Value 格式）的详细信息，所述数据体例如（仅为举例，并不限于此）：instanceId = AY123456, clusterId = Hangzhou , value = 92 , bizTime = 1427041923825 , unit = Percent，其中，每一等号左侧表示键（Key），每一等号右侧表示值（Value），等号左右两侧的信息组成键-值对格式的数据体，在此，

所述数据体所包括的键-对可包括一个或多个，其键-对的数量并不受限制。

优选的，所述初始待过滤数据从海量数据中获取，所述第一装置 11 还包括：从分布式消息中间件中获取所述初始待过滤数据的单元。所述第一装置 11 通过分布式消息中间件，优选的，所述分布式消息中间件为 MetaQ（一种分布式消息中间件），MetaQ 是一款分布式、队列模型的消息中间件，MetaQ 具有以下特点：能够保证严格的消息顺序；提供丰富的消息拉取模式，高效的订阅者水平扩展能力，实时的消息订阅机制，亿级消息堆积能力，利用了 MetaQ 的集群数据 Sharding（分片）的特性，可以使多个设备 1 形成多个功能完全相同的对等节点进行集群，并使使集群具备了负载均衡能力，满足在海量数据背景下的可扩展性，高可用性和性能要求。

10 优选的，所述第一装置 11 还可以包括：用于将所述结构化待过滤数据发送至阻塞队列的单元；相应地，所述第三装置 13 包括从所述阻塞队列中获取所述结构化待过滤数据的单元。

在此，所述阻塞队列能够在队列满的时候阻塞进一步的入队操作直至所述阻塞队列的队列不满。具体地，所述第一装置 11 将所述结构化待过滤数据发送至阻塞队列，则所述结构化待过滤数据进入阻塞队列等待，所述第三装置 13 根据所述结构化待过滤数据的等待次序，依此从所述阻塞队列中获取所述结构化待过滤数据，当所述结构化待过滤数据被获取后即从所述阻塞队列删除。在此，所述阻塞队列中等待的结构化待过滤数据占满所述阻塞队列时，所述阻塞队列阻塞所述第一装置 11 发送过滤数据进入阻塞队列的操作，从而能够避免处理能力不足时，内存占用过大，从而在海量数据过滤过程中起到削峰填谷的作用，避免处理故障。

进一步地，所述第二装置 12 用于加载过滤规则，其中，每一所述过滤规则包括：规则领域标识、规则名称及规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表。

在此，所述规则领域标识用于表明所述过滤规则的类别，其中，所述的类别例如且  
25 不限于：主机的 CPU 占用率、某网站的访问延迟时间等，所述规则领域标识可以采用数据或文字等进行标识，此外，任何能够被计算机识别的标识的方式都可以作为所述数据领域标识的实施方式，并以引用的方式包含于此。优选地，所述规则领域标识的内容与所述数据领域标识的内容相同或基本相同，以便所述第三装置 13 根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤  
30 规则。其中，所述规则名称可以是全局唯一识别的规则名称，以便于过滤规则的管理维

护。其中，规则运算表达式可以是数字、字符串形式组成的规则表达式，例如（仅为举例，并不限于此）：`instanceId = 'AY123456' || clusterId = hangzhou )&& value > 80`，规则运算表达式还可以包括非数字、字符串等原生类型组成的数据集合类型，例如（仅为举例，并不限于此）：数组、哈希集合等。

5       进一步地，所述第二装置 12 建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表，所述第一规则列表用于为第三装置 13 获取过滤规则提供支持。

      进一步地，所述第三装置 13 获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则。具体地，所述第三装置 13 根据所述数据领域标识从所述第一规则列表中获取具有  
10   与数据领域标识相对应的规则领域标识的若干过滤规则。

      进一步地，所述第四装置 14 利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

      优选的，对于每一所述结构化待过滤数据，所述第三装置 13 根据其数据领域标识获取若干具有相应相同的规则领域标识的若干过滤规则，则第四装置 14 利用每一所获取的  
15   过滤规则对所述结构化待过滤数据进行一次匹配运算，所述第四装置 14 对若干所获取的过滤规则进行并行匹配计算，以充分利用多核中央处理器的性能，提高过滤效率。

      具体地，所述第四装置 14 包括：用于对所获取的过滤规则进行规则编译，以建立可运行抽象语法树的单元；和用于将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的单  
20   元。

      所述第四装置 14 实现了抽象语法树的功能，能够支持算术运算，字符串运算，关系运算，逻辑运算，正则表达式运算和集合运算等，并预留了扩展接口，可支持用户自定义的运算等。

      进一步地，所述第四装置 14 对所获取的过滤规则进行规则编译，以建立可运行抽象  
25   语法树（AST，Abstract Syntax Tree），在此，所述抽象语法树是是规则表达式的抽象语法结构的树状表现形式。

      具体地，所述用于对所获取的过滤规则进行规则编译、以建立可运行抽象语法树的单元包括：用于对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树的模块；和用于对所述抽象语法树进行预计算，以获得所述可运行抽象语法树的模块。

30       具体地，对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树，可以

采用 Antlr (Another Tool for Language Recognition) 来实现, 能够将用户自定义的过滤规则表达式转化为抽象语法树; 通过对规则表达式进行词法分析得到 AST 的 Token 流, Token 流 (令牌) 包括分析识别出的字符串规则各种运算操作符, 运算操作符包括但不限于例如: 操作符、数字、字符串、变量、正则表达式等。

5 其中, 运算操作符例如包括以下示例代码:

```

private Object computeExpression(CommonTree operator, ArrayDeque<Object>
oprands){
    switch (operator.getType()) {
        case MINUS:
10         case MUL:
        case DIV:
        case MOD:
            return mathsEval(operator, oprands);
        case PLUS:
15         return plusEval(oprands);
        case EQEQ:
        case NE:
        case GT:
        case GE:
20         case LT:
        case LE:
            return RelationEval(operator, oprands);
        case BANG:
        case ANDAND:
25         case OROR:
            return logicalEval(operator, oprands);
        case IN:
        case NOTIN :
            return inEval(operator, oprands);
30         case LIKE:
        case NOTLIKE:
            return likeEval(operator, oprands);
    }
}

```

```

        return null;
    }

```

在具体的应用场景中，例如过滤规则的规则表达式为以下字符串形式的内容：

CPU > 90/100 and clusterId in ['hz','qd'] and instanceId like 'AK47\w+'

5 图 8 至图 10 示出根据本申请一具体场景中利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算的示意图。通过编写 Antlr 词法分析规则，得到如图 9 所示的 AST token 流，并在系统中的保存形式采用运算表达式后缀表示法来解决优先级问题，如图 8 所示，保存形式为：OP:操作符，Num:数字，Var:变量，Regex:正则表达式，StrArray:字符串数组。

10 接着，对所述抽象语法树进行预计算，以获得所述可运行抽象语法树。其中，所述预计算用于将 AST token 流中的常量表达式进行预先计算，以判断子表达式是否可计算，并通过预计算检查所述抽象语法树中每个元素是否为特殊类型，将其中的特殊类型的元素转换为程序语言数据结构元素，例如但不限于，将 Like 操作参数元素解释转化为正则表达式，将 In 操作参数元素解释转化为集合。通过预计算能够将 AST 中的常量表达式

15 进行预先预算，从而加快运行时处理速度，并特殊类型的元素转化为程序语言数据结构元素，其中，所述特殊类型的元素是非数字、字符串组成的原生类型的元素，例如但不限于数据集合类型，例如但不限于数组、哈希 map、哈希集合等。

接上例，在具体的场景中，所述第四装置 14 对图 9 所示的所述抽象语法树进行一次预计算，计算结果为可运行抽象语法树（AST），其中，AST 的 token 流如图 10 所示，

20 其中，“0.9”、“Java.util.HashSet['hz','qd']”及“Java.util.regex.pattern'AK47\W+’”为经过预计算的计算结果。

在一可选的实施例中，进行预计算的示例代码如下：

```

protected Object processNode(Object o) {
    if(!(o instanceof CommonTree))
25         return o;
    CommonTree node = (CommonTree) o;
    switch(node.getType()){
        case Int : return Double.parseDouble(node.getText());
        case RealNumber : return Double.parseDouble(node.getText());
30         case STRING :
            String text = node.getText();

```

```

        text = text.substring(1, text.length()-1);
        if(node.getParent().getType() == LIKE && node.getChildIndex() ==
node.getParent().getChildCount()-1){ //如果是 like 表达式，则编译为正则
            Pattern p = Pattern.compile(text);
5            return p;
        }
        return text;
    case BOOLEAN : return Boolean.valueOf(node.getText());
    case STRINGARRAY :
10        text = node.getText();
        text = text.substring(1, text.length()-1);
        String[] splits = text.split(",");
        Set<String> set_s = new HashSet<String>();
        for(String spllit : splits)
15
            set_s.add(spllit.substring(spllit.indexOf('"')+1,spllit.lastIndexOf('"')));
            return set_s;
    case NUMBERARRAY:
        text = node.getText();
20        text = text.substring(1, text.length()-1);
        splits = text.split(",");
        Set<Double> set_n = new TreeSet<Double>(new RealComparator());
        for(String spllit : splits)
            set_n.add(Double.parseDouble(spllit));
25        return set_n;
    }

    return o;
}

```

30 当然本领域技术人员应当能够理解，上述示例代码仅为举例，今后可能出现的进行预计算的方法、代码等其他形式，如适用本申请，均能够以引用的方式包含于本申请的保护范围之内。

具体地，用于对所述抽象语法树进行预计算，以获得所述可运行抽象语法树的模块，

其中，该模块用于：根据所述抽象语法树创建运行堆栈，将所述抽象语法树中的元素传入所述运行堆栈中，当所述元素为操作符时，将操作符对应的两个操作数传出所述运行堆栈，计算以获得计算结果，用于当所述元素为特殊元素，则将所述特殊元素转换为程序语言数据结构元素后传入所述运行堆栈中。

5           进一步地，所述第四装置 14 还包括将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的单元。

          利用若干所述可运行抽象语法树进行并行匹配计算运算的过程与一趟预计算相同，在正常情况下运行时 AST 中的所有表达式均为可计算表达式，所以最终的计算结果是一个确定的值，该值为布尔值 FALSE 或者 TRUE，如果计算结果的布尔值为 TRUE，则该  
10   结构化数据则判断为满足该过滤规则。

          在此，通过利用所述可运行抽象语法树对所述数据进行匹配计算，当本申请所述设备 1 被分配到了 1000 个过滤规则，则对于每一所述结构化待过滤数据，在所述设备 1 的线程池中对这 1000 个过滤规则并发执行匹配运算，以充分利用多核 CPU 的性能来并发  
15   计算过滤规则。

          具体地，所述用于将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的单元包括：用于将所述可运行抽象语法树的变量替换为所述数据体中的参数的模块；用于利用所述运行堆栈对所述可运行抽象语法树进行匹配计算的模块。

20           其中，将所述可运行抽象语法树的变量替换为所述数据体中的参数的示例代码如下：

```
protected Object processNode(Object o) {  
    if(!(o instanceof CommonTree))  
        return o;  
    CommonTree node = (CommonTree) o;  
25     if(isToken(node, Identifier)){  
        return variables.get().get(node.getText());  
    }  
    return o;  
}
```

30           利用所述运行堆栈对所述可运行抽象语法树进行匹配计算，其中，对 AST 每个节点进行处理，并放入运行时栈的示例代码如下：

```

protected ArrayDeque<Object> compute(){
    /**
     * 运行时栈
     */
5   ArrayDeque<Object> stack = new ArrayDeque<Object>();
    Iterator<Object> it = enter();
    while(it.hasNext()){
        Object o = it.next();
        o = processNode(o);
10   pushStack(stack,o);
    }
    return stack;
}

```

其中，对 AST 中的操作符节点进行相应的运算的示例代码如下：

```

15  protected void pushStack(ArrayDeque<Object> stack, Object o){
    stack.add(o);
    if(isToken(o,UP)){
        ArrayDeque<Object> oprands = new ArrayDeque<Object>();
        while(true){
20      o = stack.pollLast();
        oprands.add(o);
        if(isToken(o,DOWN))
            break;
        }
25  CommonTree operator = (CommonTree) stack.pollLast();
    /**
     * 运行时计算的，则原样返回
     */
    if(!shouldCompute(oprands)){
30      stack.add(operator);
        while(!oprands.isEmpty())
            stack.add(oprands.pollLast());
    }else{

```

```

        oprands.pollLast();//把 DOWN 退掉
        o = computeExpression(operator, oprands);
        stack.add(o);
    }
5      }
    }

```

其中，进行匹配计算的示例代码如下：

```

private Object mathsEval(CommonTree operator, ArrayDeque<Object> oprands){
    Double operand1 = ((Number) oprands.pollLast()).doubleValue();
10    Double operand2 = ((Number) oprands.pollLast()).doubleValue();
    switch (operator.getType()) {
        case MINUS: return operand1 - operand2;
        case MUL: return operand1 * operand2;
        case DIV: return operand1 / operand2;
15    case MOD: return operand1 % operand2;
    }
    return null;
}

```

当然本领域技术人员应当能够理解，上述示例代码仅为举例，今后可能出现的方法、  
20 代码等其他形式，如适用本申请，均能够以引用的方式包含于本申请的保护范围之内。

此后，所述设备 1 可以对该结构化待过滤数据进行进一步处理，例如报警等。

图 2 示出根据本申请一优选实施例提供的一种用于过滤数据的设备的设备示意图，  
所述设备 1 包括：第一装置 11'、第二装置 12'、第三装置 13'、第四装置 14' 和第五装  
置 15'。

25 所述第一装置 11'、第三装置 13' 和第四装置 14' 的内容与图 1 所示的设备 1 的第一  
装置 11、第三装置 13 和第四装置 14 的内容相同或基本相同，为简明起见，不再赘述，  
仅以引用的方式包含于此。

优选的，所述第二装置 12' 在引用图 1 所示的第二装置 12 的内容的基础上，所述第  
二装置 12' 还包括：建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则  
30 列表的单元；所述第二装置 12' 按照所述过滤规则的规则领域标识和所述过滤规则的规则  
名称为两个维度的索引建立第一规则列表和第二规则列表，其中，以所述过滤规则的规  
则领域标识为索引的第一规则列表供过滤数据时查找，以所述过滤规则的规则名称为索

引的第二规则列表供过滤规则的管理和维护时查找。获得所述结构化待过滤数据时，根据数据领域标识匹配查找第一规则列表中的过滤规则，找到对应的过滤规则的列表，并遍历该过滤规则的列表，将格式化待过滤数据的数据体作为输入参数，对列表中的每一规则做并发匹配计算。所述第二规则列表便于对过滤规则进行管理。

5 所述第五装置 15' 用于新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

具体地，所述第五装置 15' 包括用于将新增的过滤规则增加至所述第二规则列表中的单元；用于从所述第二规则列表中删除相应的过滤规则的单元；用于从所述第二规则列表中查找过滤规则，并对所查找的过滤规则进行修改编译的单元。所述第五装置 15' 10 能够对过滤规则进行修改和增删操作，提高过滤规则的灵活性。

图 3 示出根据本申请另一优选实施例提供的一种用于过滤数据的设备的设备示意图，其中，所述设备 1 包括第一装置 11''、第二装置 12''、第三装置 13''、第四装置 14''、第五装置 15'' 和第六装置 16''。

其中，所述第一装置 11''、第二装置 12''、第三装置 13''、第四装置 14'' 和第五装置 15'' 与图 2 所示的设备 1 的第一装置 11'、第二装置 12'、第三装置 13'、第四装置 14' 15 和第五装置 15'' 的内容相同或基本相同，为简明起见，不再赘述，并以引用的方式包含于此。

在此，每一所述过滤规则还包括：所述过滤规则所绑定的通知器的信息；所述第六装置 16' 用于将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的 20 通知器，以备传输。其中，在此，所述通知器是对预留通知接口的一组实现，可实现自定义的通知方式，比如使用不同的传输协议，不同的压缩算法，不同的序列化算法传输至下游系统集群中不同的系统中。其中，所述通知器可以在过滤规则创建时，即进行自由地组合装配绑定至任意过滤规则。

图 4 示出根据本申请一方面提供的一种用于过滤数据的方法流程图，其中，所述方法 25 包括：步骤 S11、步骤 S12、步骤 S13 和步骤 S14。

具体地，所述步骤 S11 包括：获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；所述步骤 S12 包括：加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称、规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤 30 规则的第一规则列表；所述步骤 S13 包括：获取所述结构化待过滤数据，并根据所述数

据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；所述步骤 S14 包括：利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

进一步地，在所述步骤 S11 中：获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体在此，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体。

其中，所述数据领域标识用于表明所述结构化待过滤数据的类别，其中，所述类别例如且不限于：主机的 CPU 占用率、某网站的访问延迟时间等，所述数据领域标识可以采用数据或文字等进行标识，此外，任何能够被计算机识别的标识的方式都可以作为所述数据领域标识的实施方式，并以引用的方式包含于此。其中，所述键-值对格式的数据体记录所述结构化待过滤数据的键-值对格式（Key-Value 格式）的详细信息，所述数据体例如（仅为举例，并不限于此）：instanceId = AY123456, clusterId = Hangzhou , value = 92 , bizTime = 1427041923825 , unit = Percent，其中，每一等号左侧表示键（Key），每一等号右侧表示值（Value），等号左右两侧的信息组成键-值对格式的数据体，在此，所述数据体所包括的键-对可包括一个或多个，其键-对的数量并不受限制。

优选的，所述初始待过滤数据从海量数据中获取，所述步骤 S11 还包括：从分布式消息中间件中获取所述初始待过滤数据，通过分布式消息中间件，优选的，MetaQ（一种分布式消息中间件）是一款分布式、队列模型的消息中间件，具有以下特点：能够保证严格的消息顺序；提供丰富的消息拉取模式，高效的订阅者水平扩展能力，实时的消息订阅机制，亿级消息堆积能力，利用了 MetaQ 的集群数据 Sharding（分片）的特性，图 7 示出根据本申请一优选的实施例提供的一种应用所述用于过滤数据设备的系统的设备示意图，多个设备 1 形成多个功能完全相同的对等节点进行集群，并使使集群具备了负载均衡能力，满足在海量数据背景下的可扩展性，高可用性和性能要求。

优选的，所述步骤 S11 还包括：将所述结构化待过滤数据发送至阻塞队列；相应地，所述步骤 S13 包括：从所述阻塞队列中获取所述结构化待过滤数据。

在此，所述阻塞队列能够在队列满的时候阻塞进一步的入队操作直至所述阻塞队列的队列不满。具体地，所述步骤 S11 将所述结构化待过滤数据发送至阻塞队列，则所述结构化待过滤数据进入阻塞队列等待，所述步骤 S13 根据所述结构化待过滤数据的等待次序，依此从所述阻塞队列中获取所述结构化待过滤数据，当所述结构化待过滤数据被获取后即从所述阻塞队列删除。在此，所述阻塞队列中等待的结构化待过滤数据占满所

述阻塞队列时，所述阻塞队列阻塞所述步骤 S11 发送过滤数据进入阻塞队列的操作，从而能够避免处理能力不足时，内存占用过大，从而在海量数据过滤过程中起到削峰填谷的作用，避免处理故障。

进一步地，在所述步骤 S12 中，加载过滤规则，其中，每一所述过滤规则包括：规则领域标识、规则名称及规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表。

在此，所述规则领域标识用于表明所述过滤规则的类别，其中，所述的类别例如且不限于：主机的 CPU 占用率、某网站的访问延迟时间等，所述规则领域标识可以采用数据或文字等进行标识，此外，任何能够被计算机识别的标识的方式都可以作为所述数据领域标识的实施方式，并以引用的方式包含于此。优选地，所述规则领域标识的内容与所述数据领域标识的内容相同或基本相同，以便所述步骤 S13 根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则。其中，所述规则名称可以是全局唯一识别的规则名称，以便于过滤规则的管理维护。其中，规则运算表达式可以是数字、字符串形式组成的规则表达式，例如（仅为举例，并不限于此）：`instanceId = 'AY123456' || clusterId = hangzhou )&& value > 80`，规则运算表达式还可以包括非数字、字符串等原生类型组成的数据集合类型，例如（仅为举例，并不限于此）：数组、哈希集合等。

进一步地，所述步骤 S12 包括：建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表，其中，所述第一规则列表用于为步骤 S13 获取过滤规则提供支持。

进一步地，在所述步骤 S13 中，获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则。具体地，所述步骤 S13 根据所述数据领域标识从所述第一规则列表中获取具有与数据领域标识相对应的规则领域标识的若干过滤规则。

进一步地，在所述步骤 S14 中，利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

优选的，对于每一所述结构化待过滤数据，所述步骤 S13 根据其数据领域标识获取若干具有相应相同的规则领域标识的若干过滤规则，则步骤 S14 利用每一所获取的过滤规则对所述结构化待过滤数据进行一次匹配运算，所述步骤 S14 对若干所获取的过滤规则进行并行匹配计算，以充分利用多核中央处理器的性能，提高过滤效率。

具体地，所述步骤 S14 包括：对所获取的过滤规则进行规则编译，以建立可运行抽

象语法树；将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算。

所述步骤 S14 实现了抽象语法树的功能，能够支持算术运算，字符串运算，关系运算，逻辑运算，正则表达式运算和集合运算等，并预留了扩展接口，可支持用户自定义的运算等。

进一步地，对所获取的过滤规则进行规则编译，以建立可运行抽象语法树（AST，Abstract Syntax Tree），在此，所述抽象语法树是规则表达式的抽象语法结构的树状表现形式。

其中，对所获取的过滤规则进行规则编译，以建立可运行抽象语法树包括：用于对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树，具体地，可以采用 Antlr（Another Tool for Language Recognition）来实现，能够将用户自定义的过滤规则表达式转化为抽象语法树；通过对规则表达式进行词法分析得到 AST 的 Token 流，Token 流（令牌）包括分析识别出的字符串规则各种运算操作符，运算操作符包括但不限于例如：操作符、数字、字符串、变量、正则表达式等。

其中，运算操作符的示例代码与图 1 所示设备 1 的第四装置 14 转化的抽象语法树的运算操作符的示例代码的内容相同或基本相同，为简明起见，不再赘述，仅以引用的方式包含于此。

在具体的应用场景中，例如过滤规则的规则表达式为以下字符串形式的内容：

`CPU > 90/100 and clusterId in ['hz','qd'] and instanceId like 'AK47\w+'`

通过编写 Antlr 词法分析规则，得到如图 9 所示的 AST token 流，并在系统中的保存形式采用运算表达式后缀表示法来解决优先级问题，如图 8 所示，保存形式为：OP:操作符，Num:数字，Var:变量，Regex:正则表达式，StrArray:字符串数组。

接着，对所述抽象语法树进行预计算，以获得所述可运行抽象语法树。其中，所述预计算用于将 AST token 流中的常量表达式进行预先计算，以判断子表达式是否可计算，并通过预计算检查所述抽象语法树中每个元素是否为特殊类型，将其中的特殊类型的元素转换为程序语言数据结构元素，例如但不限于，将 Like 操作参数元素解释转化为正则表达式，将 In 操作参数元素解释转化为集合。通过预计算能够将 AST 中的常量表达式进行预先预算，从而加快运行时处理速度，并将特殊类型的元素转化为程序语言数据结构元素，其中，所述特殊类型的元素是非数字、字符串组成的原生类型的元素，例如但不限于数据集合类型，例如但不限于数组、哈希 map、哈希集合等。

接上例，在具体的场景中，对图 9 所示的所述抽象语法树进行一次预计算，计算结果为可运行抽象语法树（AST），其中，AST 的 token 流如图 10 所示，其中，“0.9”、“Java.util.HashSet[‘hz’, ‘qd’]”及“Java.util.regex.pattern‘AK47\W+’”为经过预计算的计算结果。

5 进行预计算的示例代码可以与图 1 所示第四装置 14 进行预计算的示例代码的内容相同或基本相同，为简明起见，不再赘述，仅以引用的方式包含于此。

具体地，对所述抽象语法树进行预计算包括：

根据所述抽象语法树创建运行堆栈，将所述抽象语法树中的元素传入所述运行堆栈中；当所述元素为操作符时，将操作符对应的两个操作数传出运行堆栈，计算以获得计  
10 算结果；当所述元素为特殊元素，则将所述特殊元素转换为程序语言数据结构元素后传入运行堆栈中。

进一步地，将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算的过程，利用若干所述可运行抽象语法树进行并行匹配计算运算的过程与一趟预计算相同，在正常情况下运行  
15 时 AST 中的所有表达式均为可计算表达式，所以最终的计算结果是一个确定的值，该值为布尔值 FALSE 或者 TRUE，如果计算结果的布尔值为 TRUE，则该结构化数据则判断为满足该过滤规则。

在此，所述利用若干所述可运行抽象语法树进行并行匹配计算通过利用所述可运行抽象语法树对所述数据进行匹配计算，例如，当本申请所述设备 1 被分配到了 1000 个过  
20 滤规则，则对于每一所述结构化待过滤数据，在所述设备 1 的线程池中对这 1000 个过滤规则并发执行匹配运算，以充分利用多核 CPU 的性能来并发计算过滤规则。

具体地，利用若干所述可运行抽象语法树进行并行匹配计算包括：将所述可运行抽象语法树的变量替换为所述数据体中的参数；利用所述运行堆栈对所述可运行抽象语法树进行匹配计算。

25 将所述可运行抽象语法树的变量替换为所述数据体中的参数的示例代码与所述图 1 中的设备 1 的第四装置 14 替换示例代码的内容相同或基本相同，为简明起见，不再赘述，仅以引用的方式包含于此。

对 AST 中的操作符节点进行相应的运算的示例代码与所述图 1 中的设备 1 的第四装置 14 进行相应的运算的示例代码的内容相同或基本相同，为简明起见，不再赘述，仅以  
30 引用的方式包含于此。

同样地,进行匹配计算与所述图 1 中的设备 1 的第四装置 14 进行匹配计算的示例代码的内容相同或基本相同,为简明起见,不再赘述,仅以引用的方式包含于此。

此后,所述方法还可以对该结构化待过滤数据进行进一步处理,例如报警等。

图 5 示出根据本申请一优选实施例提供的一种用于过滤数据的方法流程示意图,所述方法包括:步骤 S11'、步骤 S12'、步骤 S13'、步骤 S14' 和步骤 S15'。

所述步骤 S11'、步骤 S13'和步骤 S14'的内容与图 4 所示的步骤 S11、步骤 S12 和步骤 S14 的内容相同或基本相同,为简明起见,不再赘述,仅以引用的方式包含于此。

优选的,所述步骤 S12'在引用图 4 所示的步骤 S12 的内容的基础上,所述步骤 S12'还包括:建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则列表;所述步骤 S12'按照所述过滤规则的规则领域标识和所述过滤规则的规则名称为两个维度的索引建立第一规则列表和第二规则列表,其中,以所述过滤规则的规则领域标识为索引的第一规则列表供过滤数据时查找,以所述过滤规则的规则名称为索引的第二规则列表供过滤规则的管理和维护时查找。获得所述结构化待过滤数据时,根据数据领域标识匹配查找第一规则列表中的过滤规则,找到对应的过滤规则的列表,并遍历该过滤规则的列表,将格式化待过滤数据的数据体作为输入参数,对列表中的每一规则做并发匹配计算。所述第二规则列表便于对过滤规则进行管理。

在所述步骤 S15'中,新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

具体地,所述步骤 S15'包括以下至少任一项:将新增的过滤规则增加至所述第二规则列表中;从所述第二规则列表中删除相应的过滤规则;从所述第二规则列表中查找过滤规则,并对所查找的过滤规则进行修改编译,所述步骤 S15'能够对过滤规则进行修改和增删操作,提高过滤规则的灵活性。

图 6 示出根据本申请另一优选实施例提供的一种用于过滤数据的方法流程图,其中,所述方法包括步骤 S11''、步骤 S12''、步骤 S13''、步骤 S14''、步骤 S15''和步骤 S16''。

其中,所述步骤 S11''、步骤 S12''、步骤 S13''、步骤 S14''和步骤 S15''与图 5 所示的步骤 S11'、步骤 S12'、步骤 S13'、步骤 S14'和步骤 S15'的内容相同或基本相同,为简明起见,不再赘述,并以引用的方式包含于此。

在此,每一所述过滤规则还包括:所述过滤规则所绑定的通知器的信息;在所述步骤 S16'中,将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的通知器,以备传输。在此,所述通知器是对预留通知接口的一组实现,可实现自定义的通

知方式，比如使用不同的传输协议，不同的压缩算法，不同的序列化算法传输至下游系统集群中不同的系统中。其中，所述通知器可以在过滤规则创建时，即进行自由地组合装配绑定至任意过滤规则。

与现有技术相比，根据本申请一实施例所提供的用于数据过滤的设备与方法采用流式运算方式，不会在内存中缓存也不会固化数据，即每次获取初始待过滤数据后将所述初始待过滤数据转换为结构化待过滤数据，并利用相应的过滤规则实时进行匹配计算，随即得到过滤结果，解决海量流式数据的过滤的实时性问题；

进一步地，根据本申请一实施例所提供的用于数据过滤的设备与方法所述过滤数据的方法和设备支持算术运算，字符串运算，关系运算，逻辑运算，正则表达式运算，集合运算，且预留了扩展接口，并且所述过滤规则为简单的带变量的运算表达式形式，解决了过滤规则描述复杂、扩展不易及管理困难的问题；

此外，本申请根据本申请一实施例所提供的用于数据过滤的设备与方法为自主设计开发，成本相对较低，且在任意代码路径上都可以监控和调优。

经多次性能测试，得到的性能指标大致为在单台 4 核 8G 配置的虚拟机可以支撑 50 万过滤规则，处理流式数据 TPS 达到 20000，过滤出有效数据 TPS 达到 2000，系统平均负载稳定在 load1-4 左右，CPU 资源得到了有效利用。

在本申请一个典型的配置中，终端、服务网络的设备和可信方均包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

需要注意的是，本申请可在软件和/或软件与硬件的组合体中被实施，例如，可采用

专用集成电路（ASIC）、通用目的计算机或任何其他类似硬件设备来实现。在一个实施例中，本申请的软件程序可以通过处理器执行以实现上文所述步骤或功能。同样地，本申请的软件程序（包括相关的数据结构）可以被存储到计算机可读记录介质中，例如，RAM 存储器，磁或光驱动器或软磁盘及类似设备。另外，本申请的一些步骤或功能可采用硬件来实现，例如，作为与处理器配合从而执行各个步骤或功能的电路。

另外，本申请的一部分可被应用为计算机程序产品，例如计算机程序指令，当其被计算机执行时，通过该计算机的操作，可以调用或提供根据本申请的方法和/或技术方案。而调用本申请的方法的程序指令，可能被存储在固定的或可移动的记录介质中，和/或通过广播或其他信号承载媒体中的数据流而被传输，和/或被存储在根据所述程序指令运行的计算机设备的工作存储器中。在此，根据本申请的一个实施例包括一个装置，该装置包括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该装置运行基于前述根据本申请的多个实施例的方法和/或技术方案。

对于本领域技术人员而言，显然本申请不限于上述示范性实施例的细节，而且在不背离本申请的精神或基本特征的情况下，能够以其他的具体形式实现本申请。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本申请的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本申请内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外，显然“包括”一词不排除其他单元或步骤，单数不排除复数。装置权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

## 权利要求书

1. 一种用于过滤数据的方法，其中，所述方法包括：

获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；

5        加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称及规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表；

获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；

利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算。

10       2. 根据权利要求1所述的方法，其中，获取初始待过滤数据包括：

从分布式消息中间件中获取所述初始待过滤数据。

3. 根据权利要求1或2所述的方法，其中，将所述初始待过滤数据转换为结构化待过滤数据还包括：

将所述结构化待过滤数据发送至阻塞队列；

15       获取所述结构化待过滤数据包括：

从所述阻塞队列中获取所述结构化待过滤数据。

4. 根据权利要求1至3中任一项所述的方法，其中，利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配运算包括：

对所获取的过滤规则进行规则编译，以建立可运行抽象语法树；

20       将所述结构化待过滤数据的数据体作为输入参数，遍历若干所述可运行抽象语法树，并利用若干所述可运行抽象语法树进行并行匹配计算。

5. 根据权利要求4所述的方法，其中，对所获取的过滤规则进行规则编译，以建立可运行抽象语法树包括：

对所获取的过滤规则的规则表达式进行分析，以转化为抽象语法树；

25       对所述抽象语法树进行预计算，以获得所述可运行抽象语法树；

其中，对所述抽象语法树进行一次预计算包括：

根据所述抽象语法树创建运行堆栈，将所述抽象语法树中的元素传入所述运行堆栈中；

30       当所述元素为操作符时，将所述操作符对应的两个操作数传出所述运行堆栈，计算以获得计算结果；

当所述元素为特殊元素，则将所述特殊元素转换为程序语言数据结构元素后传入所述运行堆栈中。

6. 根据权利要求 4 或 5 所述的方法，其中，利用若干所述可运行抽象语法树进行并行匹配计算包括：

5 将所述可运行抽象语法树的变量替换为所述数据体中的参数；  
利用所述运行堆栈对所述可运行抽象语法树进行匹配计算。

7. 根据权利要求 1 至 6 中任一项所述的方法，其中，所述方法还包括：  
新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

8. 根据权利要求 7 所述的方法，其中，建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表还包括：

建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则列表；

所述新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译包括至少以下任一项：

将新增的过滤规则增加至所述第二规则列表中；

15 从所述第二规则列表中删除相应的过滤规则；

从所述第二规则列表中查找过滤规则，并对所查找的过滤规则进行修改编译。

9. 根据权利要求 1 至 8 中任一项所述的方法，其中，每一所述过滤规则还包括：所述过滤规则所绑定的通知器的信息；

所述方法还包括：

20 将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的通知器，以备传输。

10. 一种用于过滤数据的设备，其中，所述设备包括：

第一装置，用于获取初始待过滤数据，并将所述初始待过滤数据转换为结构化待过滤数据，其中，所述结构化待过滤数据包括数据领域标识和键-值对格式的数据体；

25 第二装置，用于加载过滤规则，其中，每一所述过滤规则包括规则领域标识、规则名称及规则运算表达式，并建立以所述过滤规则的领域标识为索引的所述过滤规则的第一规则列表；

第三装置，用于获取所述结构化待过滤数据，并根据所述数据领域标识从所述第一规则列表中获取具有与所述数据领域标识相对应的规则领域标识的若干过滤规则；

30 第四装置，用于利用所获取的若干过滤规则对所述结构化待过滤数据进行并行匹配

运算。

11. 根据权利要求 10 所述的设备, 其中, 所述第一装置包括:

从分布式消息中间件中获取所述初始待过滤数据的单元。

12. 根据权利要求 10 或 11 所述的设备, 其中, 所述第一装置包括:

5 用于将所述结构化待过滤数据发送至阻塞队列的单元;

所述第三装置包括:

从所述阻塞队列中获取所述结构化待过滤数据的单元。

13. 根据权利要求 10 至 12 中任一项所述的设备, 其中, 所述第四装置包括:

用于对所获取的过滤规则进行规则编译, 以建立可运行抽象语法树的单元;

10 用于将所述结构化待过滤数据的数据体作为输入参数, 遍历若干所述可运行抽象语法树, 并利用若干所述可运行抽象语法树进行并行匹配计算的单元。

14. 根据权利要求 13 所述的设备, 其中, 所述用于对所获取的过滤规则进行规则编译, 以建立可运行抽象语法树的单元包括:

用于对所获取的过滤规则的规则表达式进行分析, 以转化为抽象语法树的模块;

15 用于对所述抽象语法树进行预计算, 以获得所述可运行抽象语法树的模块, 其中, 该模块用于:

根据所述抽象语法树创建运行堆栈, 将所述抽象语法树中的元素传入所述运行堆栈中,

20 当所述元素为操作符时, 将操作符对应的两个操作数传出所述运行堆栈, 计算以获得计算结果,

用于当所述元素为特殊元素, 则将所述特殊元素转换为程序语言数据结构元素后传入所述运行堆栈中。

15. 根据权利要求 13 或 14 所述的设备, 其中, 所述用于将所述结构化待过滤数据的数据体作为输入参数, 遍历若干所述可运行抽象语法树, 并利用若干所述可运行抽象语法树进行并行匹配计算的单元包括:

用于将所述可运行抽象语法树的变量替换为所述数据体中的参数的模块;

用于利用所述运行堆栈对所述可运行抽象语法树进行匹配计算的模块。

16. 根据权利要求 10 至 15 中任一项所述的设备, 其中, 所述设备还包括:

第五装置, 用于新增过滤规则、删除过滤规则或对已有的过滤规则进行修改编译。

30 17. 根据权利要求 16 所述的设备, 其中, 所述第二装置还包括:

建立根据所述过滤规则的规则名称为索引的所述过滤规则的第二规则列表的单元；  
所述第五装置包括：

用于将新增的过滤规则增加至所述第二规则列表中的单元；

用于从所述第二规则列表中删除相应的过滤规则的单元；

- 5       用于从所述第二规则列表中查找过滤规则，并对所查找的过滤规则进行修改编译的单元。

18. 根据权利要求 10 至 17 中任一项所述的设备，其中，每一所述过滤规则还包括：  
所述过滤规则所绑定的通知器的信息；

所述设备还包括：

- 10       第六装置，用于将满足相应所述过滤规则的结构化待过滤数据发送至该过滤规则所绑定的通知器，以备传输。

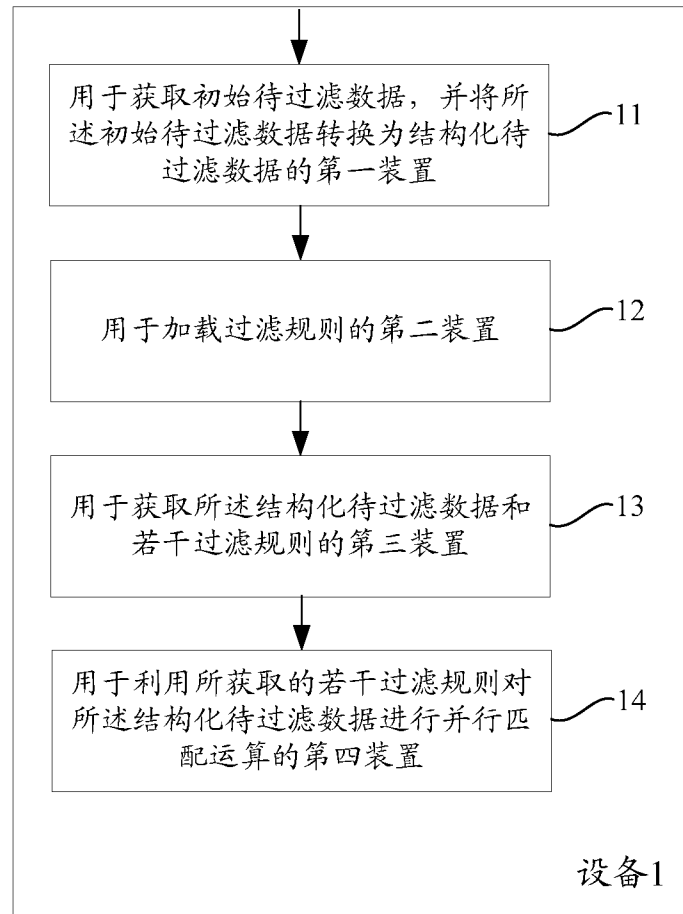


图 1

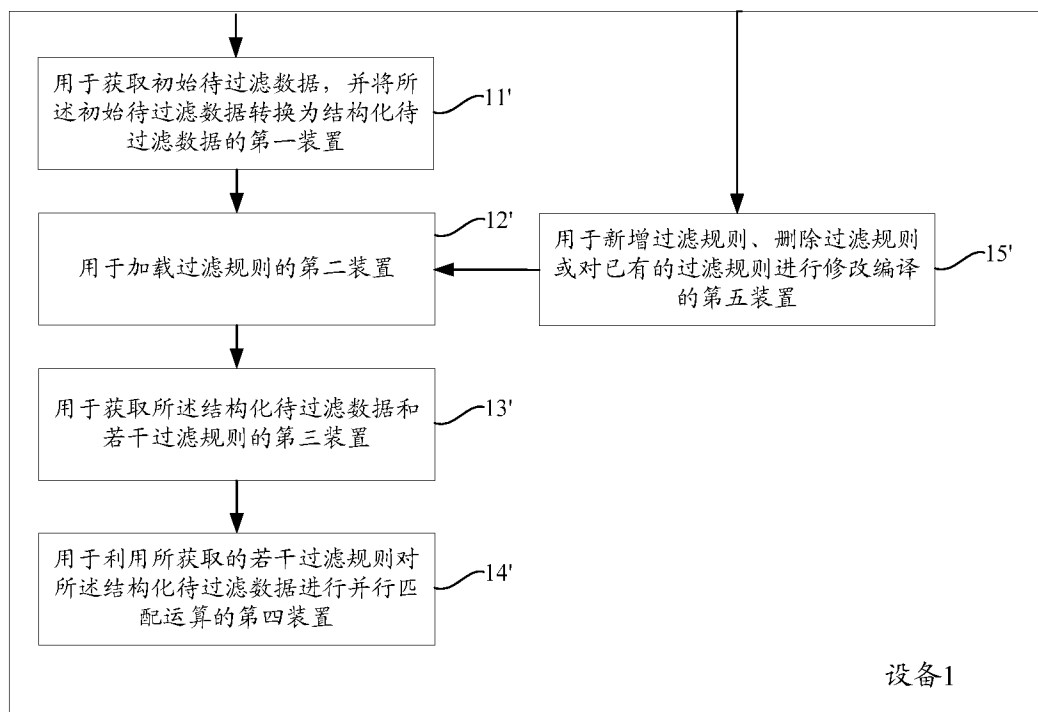


图 2

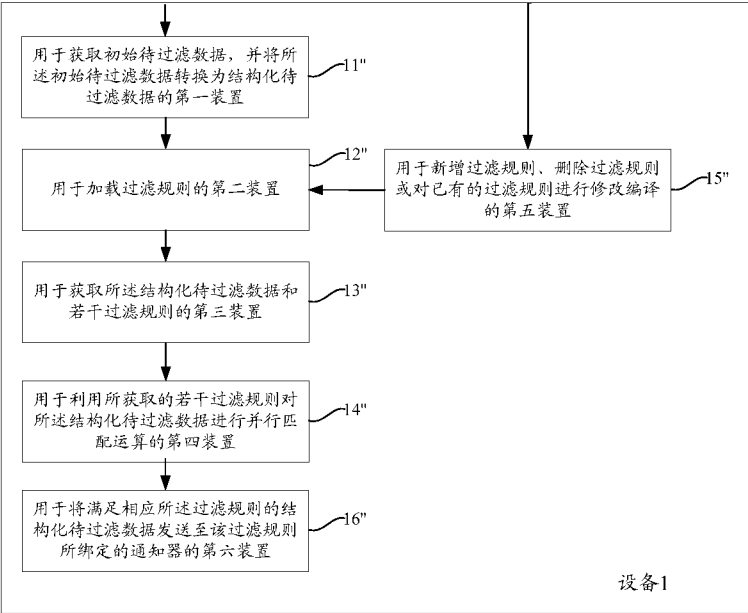


图 3

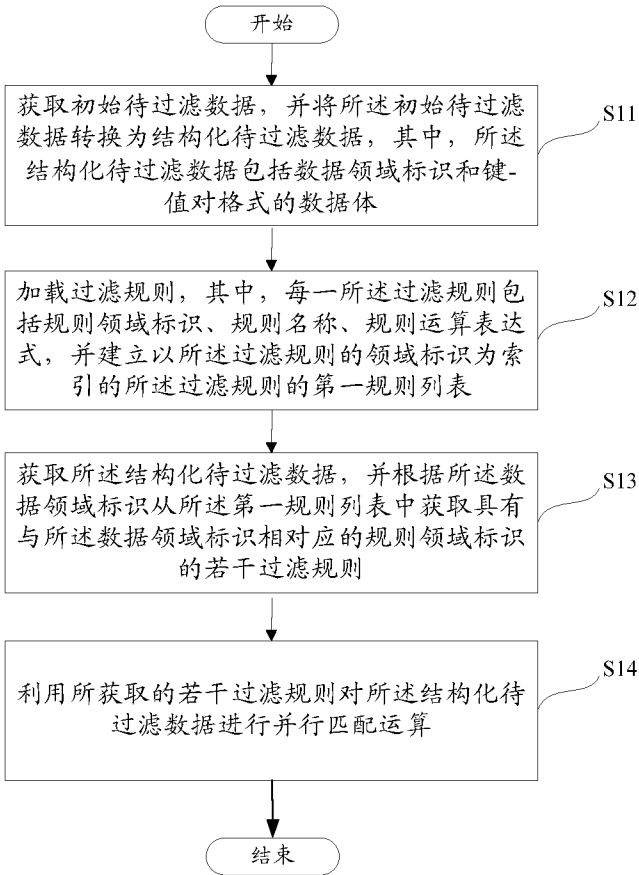


图 4

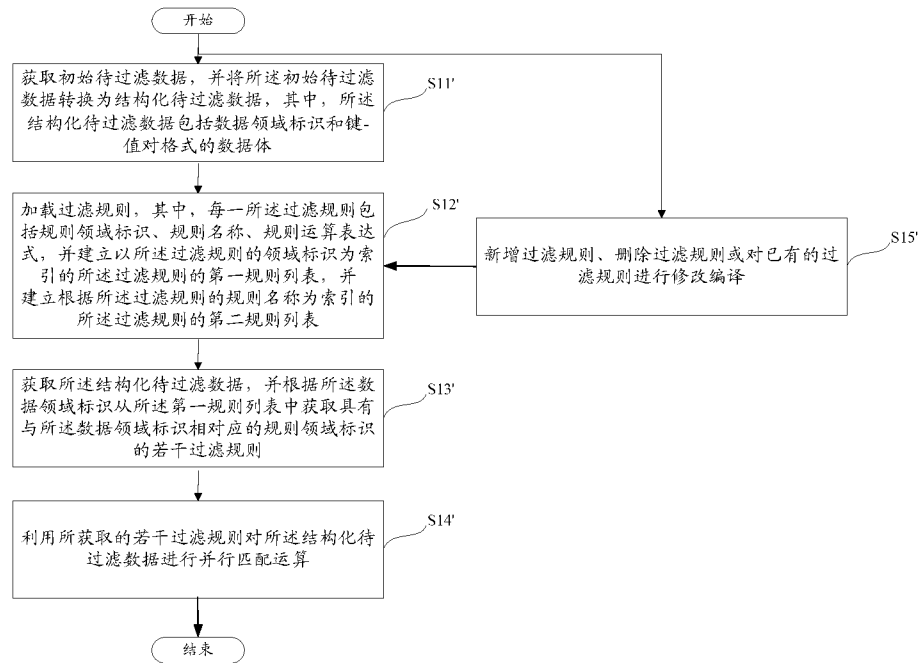


图 5

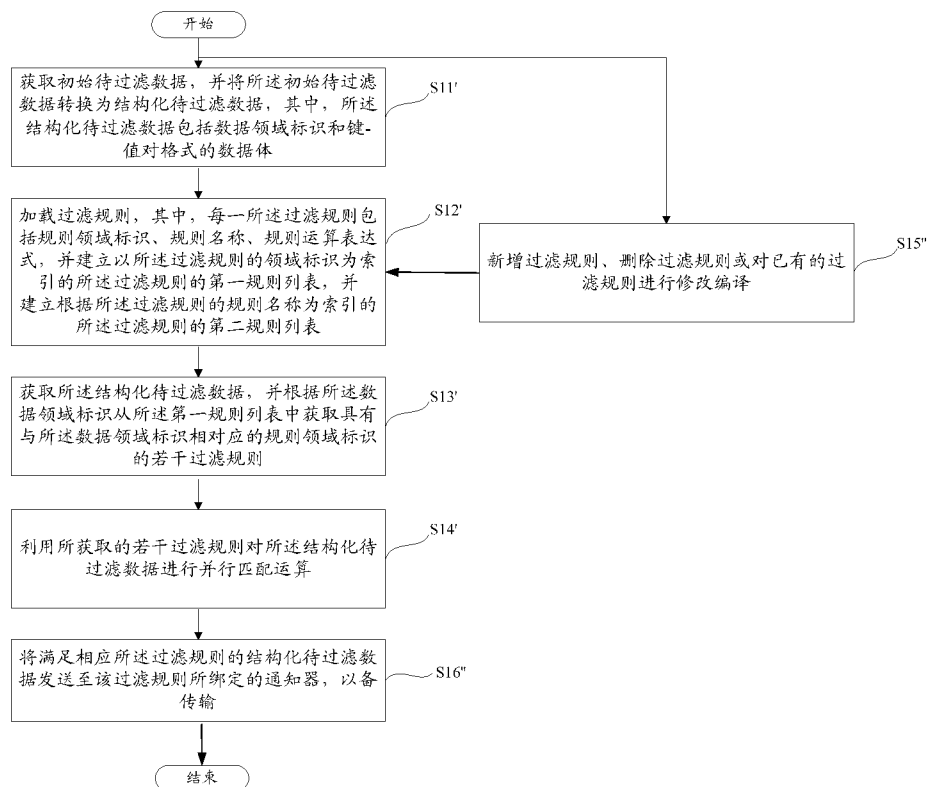


图 6

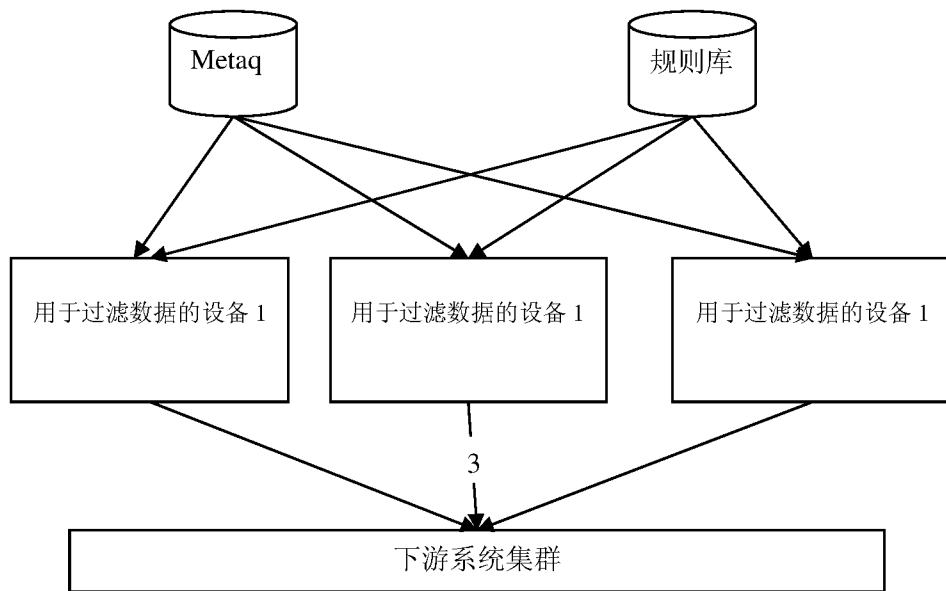


图 7

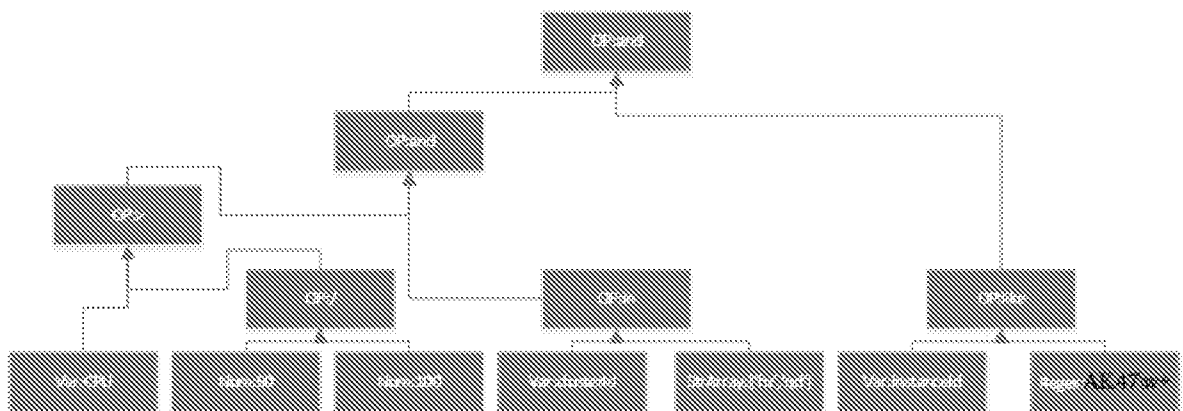


图 8

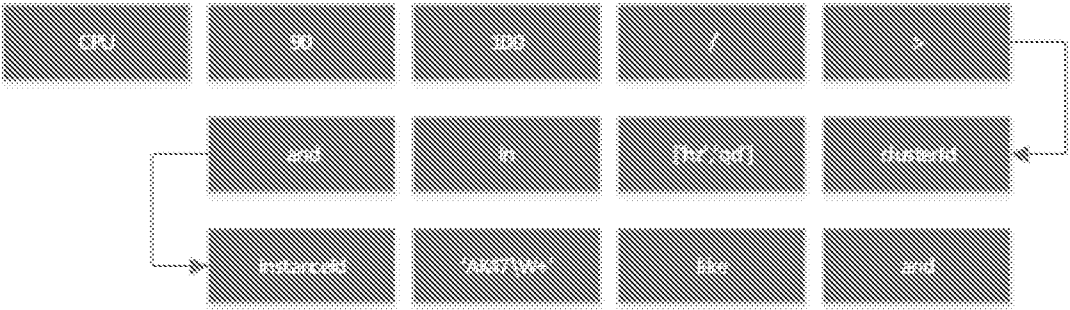


图 9

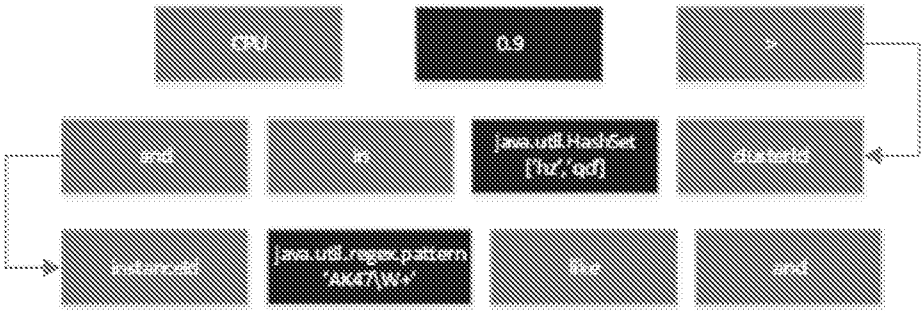


图 10

# INTERNATIONAL SEARCH REPORT

International application No.

**PCT/CN2016/088302**

## A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI, IEEE, GOOGLE: screen, structured, mark, marker, category, syntax tree, filtrate, structure, id, rule, condition, map, match, syntax, tree, list, stack

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                            | Relevant to claim No. |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Y         | CN 103034700 A (BEIJING QIHOO TECHNOLOGY CO., LTD. et al.), 10 April 2013 (10.04.2013), description, paragraphs [0006]-[0012], and figure 2   | 1-3, 7-12, 16-18      |
| Y         | CN 102467561 A (KINGDEE SOFTWARE (CHINA) CO., LTD.), 23 May 2012 (23.05.2012), description, paragraphs [0050]-[0056] and [0069], and figure 2 | 1-3, 7-12, 16-18      |
| A         | CN 103618733 A (BEIJING ZHONGCHUANG TELECOM TEST CO., LTD.), 05 March 2014 (05.03.2014), the whole document                                   | 1-18                  |
| A         | CN 104317947 A (FIBERHOME TELECOMMUNICATION TECHNOLOGIES CO., LTD.), 28 January 2015 (28.01.2015), the whole document                         | 1-18                  |
| A         | US 8949371 B1 (SYMANTEC CORPORATION), 03 February 2015 (03.02.2015), the whole document                                                       | 1-18                  |

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

|                                                                                                                                                                         |                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| * Special categories of cited documents:                                                                                                                                | “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention                                              |
| “A” document defining the general state of the art which is not considered to be of particular relevance                                                                | “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone                                                                     |
| “E” earlier application or patent but published on or after the international filing date                                                                               | “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | “&” document member of the same patent family                                                                                                                                                                                                    |
| “O” document referring to an oral disclosure, use, exhibition or other means                                                                                            |                                                                                                                                                                                                                                                  |
| “P” document published prior to the international filing date but later than the priority date claimed                                                                  |                                                                                                                                                                                                                                                  |

|                                                                                                                                                                                                                    |                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Date of the actual completion of the international search<br>23 August 2016 (23.08.2016)                                                                                                                           | Date of mailing of the international search report<br><b>11 October 2016 (11.10.2016)</b> |
| Name and mailing address of the ISA/CN:<br>State Intellectual Property Office of the P. R. China<br>No. 6, Xitucheng Road, Jimenqiao<br>Haidian District, Beijing 100088, China<br>Facsimile No.: (86-10) 62019451 | Authorized officer<br><b>ZHOU, Ya’nan</b><br>Telephone No.: (86-10) <b>62413669</b>       |

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.

**PCT/CN2016/088302**

| Patent Documents referred<br>in the Report | Publication Date | Patent Family | Publication Date |
|--------------------------------------------|------------------|---------------|------------------|
| CN 103034700 A                             | 10 April 2013    | None          |                  |
| CN 102467561 A                             | 23 May 2012      | None          |                  |
| CN 103618733 A                             | 05 March 2014    | None          |                  |
| CN 104317947 A                             | 28 January 2015  | None          |                  |
| US 8949371 B1                              | 03 February 2015 | None          |                  |

## 国际检索报告

国际申请号

PCT/CN2016/088302

## A. 主题的分类

G06F 17/30(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

## B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, WPI, EPDOC, CNKI, IEEE, GOOGLE: 过滤, 筛选, 结构化, 标识, 标记, 类别, 规则, 条件, 匹配, 语法树, 列表, 堆栈, filtrate, structure, id, rule, condition, map, match, syntax, tree, list, stack

## C. 相关文件

| 类 型* | 引用文件, 必要时, 指明相关段落                                                                           | 相关的权利要求        |
|------|---------------------------------------------------------------------------------------------|----------------|
| Y    | CN 103034700 A (北京奇虎科技有限公司 等) 2013年 4月 10日 (2013 - 04 - 10)<br>说明书第[0006]-[0012]段, 附图2      | 1-3、7-12、16-18 |
| Y    | CN 102467561 A (金蝶软件中国有限公司) 2012年 5月 23日 (2012 - 05 - 23)<br>说明书第[0050]-[0056]、[0069]段, 附图2 | 1-3、7-12、16-18 |
| A    | CN 103618733 A (北京中创信测科技股份有限公司) 2014年 3月 5日 (2014 - 03 - 05)<br>全文                          | 1-18           |
| A    | CN 104317947 A (烽火通信科技股份有限公司) 2015年 1月 28日 (2015 - 01 - 28)<br>全文                           | 1-18           |
| A    | US 8949371 B1 (SYMANTEC CORPORATION) 2015年 2月 3日 (2015 - 02 - 03)<br>全文                     | 1-18           |

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

\* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&amp;” 同族专利的文件

国际检索实际完成的日期

2016年 8月 23日

国际检索报告邮寄日期

2016年 10月 11日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)  
中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

受权官员

周亚楠

电话号码 (86-10)62413669

国际检索报告  
关于同族专利的信息

国际申请号  
PCT/CN2016/088302

| 检索报告引用的专利文件 |           |    | 公布日<br>(年/月/日) | 同族专利 | 公布日<br>(年/月/日) |
|-------------|-----------|----|----------------|------|----------------|
| CN          | 103034700 | A  | 2013年 4月 10日   | 无    |                |
| CN          | 102467561 | A  | 2012年 5月 23日   | 无    |                |
| CN          | 103618733 | A  | 2014年 3月 5日    | 无    |                |
| CN          | 104317947 | A  | 2015年 1月 28日   | 无    |                |
| US          | 8949371   | B1 | 2015年 2月 3日    | 无    |                |

表 PCT/ISA/210 (同族专利附件) (2009年7月)