



(12)发明专利

(10)授权公告号 CN 109388609 B

(45)授权公告日 2020.02.21

(21)申请号 201811158729.6

G06F 15/16(2006.01)

(22)申请日 2018.09.30

G06F 15/76(2006.01)

(65)同一申请的已公布的文献号

审查员 李晓阳

申请公布号 CN 109388609 A

(43)申请公布日 2019.02.26

(73)专利权人 中科驭数(北京)科技有限公司

地址 100190 北京市海淀区科学院南路6号

中国科学院计算技术研究所科研综合楼

(72)发明人 龚施俊 卢文岩 鄢贵海

(74)专利代理机构 北京金咨知识产权代理有限公司 11612

代理人 宋教花

(51)Int.Cl.

G06F 15/82(2006.01)

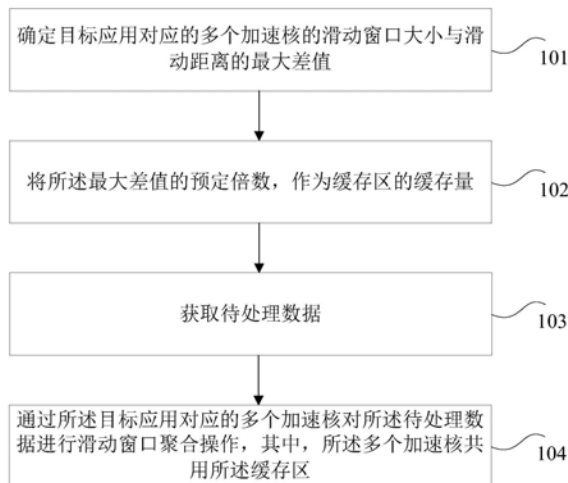
权利要求书2页 说明书7页 附图3页

(54)发明名称

基于加速核的数据处理方法和装置

(57)摘要

本发明提供了一种基于加速核的数据处理方法和装置,其中,该方法包括:确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值;将所述最大差值的预定倍数,作为缓存区的缓存量;获取待处理数据;通过所述目标应用对应的多个加速核对所待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区。通过上述方案解决了现有的每个加速核独立使用一个缓存区所导致的缓存资源需求过大、缓存区浪费严重的问题,达到了有效降低缓存资源需求,提升现有的滑动窗聚合操作中缓存资源的利用率的技术效果。



1. 一种基于加速核的数据处理方法,其特征在于,该方法包括以下步骤:
 - 确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值;
 - 将所述最大差值的预定倍数,作为缓存区的缓存量;
 - 获取待处理数据;
 - 通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区;
 - 通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作的步骤包括:
 - 所述加速核通过内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;
 - 所述加速核根据所述地址,从所述缓存区中获取历史数据;
 - 所述加速核基于所述历史数据进行滑动窗口聚合操作。
2. 根据权利要求1所述的方法,其特征在于,通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作的步骤包括:
 - 当所述待处理数据流入所述加速核的时候,所述待处理数据先流入所述缓存区。
3. 根据权利要求1所述的方法,其特征在于,确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值的步骤包括:
 - 对历史处理过程进行分析,确定多个加速核可能出现的滑动窗口大小与滑动距离的最大差值;
 - 将确定出的可能出现的滑动窗口大小与滑动距离的最大差值,作为所述目标应用的滑动窗口大小与滑动距离的最大差值。
4. 根据权利要求1至3中任一项所述的方法,其特征在于,所述聚合操作包括以下至少之一:求和、求最大值、求最小值、求平均值、数值排序。
5. 一种基于加速核的数据处理装置,其特征在于,该装置包括:
 - 确定模块,用于确定目标应用的滑动窗口大小与滑动距离的最大差值;
 - 生成模块,用于将所述最大差值的预定倍数,作为缓存区的缓存量;
 - 获取模块,用于获取待处理数据;
 - 处理模块,用于通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区;
 - 所述处理模块包括:
 - 计算单元,用于通过加速核内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;
 - 获取单元,用于控制所述加速核根据所述地址,从所述缓存区中获取历史数据;
 - 处理单元,用于控制所述加速核基于所述历史数据进行滑动窗口聚合操作。
6. 根据权利要求5所述的装置,其特征在于,当所述待处理数据流入所述加速核的时候,所述待处理数据先流入所述缓存区。
7. 一种网络设备,包括:存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至4中任一项所述方法的步骤。

8. 一种非易失性计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至4中任一项所述方法的步骤。

基于加速核的数据处理方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,特别涉及一种基于加速核的数据处理方法和装置。

背景技术

[0002] 数据流处理是计算机领域很重要的一环,在许多应用中,都需要能够提供快速地响应基于快速持续到达的数据流。以一个具体场景(股票市场应用)为例,有多个客户端监听股票市场的价格波动,然后,就需要股票市场应用系统能够有效地响应不同客户端的分析请求。在这种场景中,每一个客户端的分析请求也许都是不同的,例如,可能是:每周股价波动率、每月平均股价等。除了金融领域,有效的数据流处理在医疗健康、商业决策、科学计算、社交媒体和网络控制等领域都很重要。

[0003] 数据流管理系统也就应运而生,在数据流管理系统中,客户端通过在即将到来的数据流上注册分析请求,然后,配置窗口大小(range)以及滑动距离(slide),来产生持续的分析结果。其中,滑动距离(slide)用于表征更新结果的时间跨度,窗口大小(range)用于表征进行聚合操作的数据大小。例如,股票应用配置range等于5分钟,slide等于3分钟,就表明应用需要每3分钟更新一次结果,并且这结果来源于过去5分钟的数据。

[0004] 然而,有效处理聚合操作成为了整个系统的瓶颈,尤其对于计算繁重的金融以及科学计算领域。例如:求最大值、求最小值、平均值、求前N等聚合操作。为了对这些聚合操作进行计算,一般会通过FPGA、ASIC等专用芯片来加速滑动窗聚合。然而,由于滑动窗聚合操作的固有属性,使得这些优化方式难以解决通用性和可扩展性的问题,也将导致缓存资源的利用率低。

[0005] 针对上述问题,目前尚未提出有效的解决方案。

发明内容

[0006] 本发明实施例提供了一种基于加速核的数据处理方法和装置,以提升现有的滑动窗聚合操作中缓存资源的利用率。

[0007] 根据本发明的一方面,提供了一种基于加速核的数据处理方法,该方法包括:

[0008] 确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值;

[0009] 将所述最大差值的预定倍数,作为缓存区的缓存量;

[0010] 获取待处理数据;

[0011] 通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区。

[0012] 在一个实施方式中,通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作的步骤包括:

[0013] 当所述待处理数据流入所述加速核的时候,所述待处理数据先流入所述缓存区。

[0014] 在一个实施方式中,通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作的步骤包括:

[0015] 所述加速核通过内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;

[0016] 所述加速核根据所述地址,从所述缓存区中获取历史数据;

[0017] 所述加速核基于所述历史数据进行滑动窗口聚合操作。

[0018] 在一个实施方式中,确定目标应用的滑动窗口大小与滑动距离的最大差值的步骤包括:

[0019] 对历史处理过程进行分析,确定出可能出现的滑动窗口大小与滑动距离的最大差值;

[0020] 将确定出的可能出现的滑动窗口大小与滑动距离的最大差值,作为所述目标应用的滑动窗口大小与滑动距离的最大差值。

[0021] 在一个实施方式中,所述聚合操作包括以下至少之一:求和、求最大值、求最小值、求平均值、数值排序。

[0022] 另一方面,提供了一种基于加速核的数据处理装置,包括:

[0023] 确定模块,用于确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值;

[0024] 生成模块,用于将所述最大差值的预定倍数,作为缓存区的缓存量;

[0025] 获取模块,用于获取待处理数据;

[0026] 处理模块,用于通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区。

[0027] 在一个实施方式中,当所述待处理数据流入所述加速核的时候,所述待处理数据先流入所述缓存区。

[0028] 在一个实施方式中,所述处理模块包括:

[0029] 计算单元,用于通过加速核内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;

[0030] 获取单元,用于控制所述加速核根据所述地址,从所述缓存区中获取历史数据;

[0031] 处理单元,用于控制所述加速核基于所述历史数据进行滑动窗口聚合操作。

[0032] 又一方面,提供了一种网络设备,包括:存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现上述方法的步骤。

[0033] 又一方面,提供了一种非易失性计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现上述方法的步骤。

[0034] 在本发明实施例中,多个加速核共用缓存区,缓存区的缓存量基于目标应用的滑动窗口大小与滑动距离的最大差值确定,这样使得多个加速核都基于该缓存区进行滑动窗口聚合操作,从而可以避免每个加速核独立使用一个缓存区所导致的缓存资源需求过大、缓存区浪费严重的问题,达到了有效降低缓存资源需求,提升现有的滑动窗聚合操作中缓存资源的利用率的技术效果。

[0035] 本领域技术人员应当理解的是,能够用本发明实现的目的和优点不限于以上具体所述,并且根据以下详细说明将更清楚地理解本发明能够实现的上述和其他目的。

[0036] 并且,应当理解,前述大体的描述和后续详尽的描述均为示例性说明和解释,并不

应当用作对本发明所要求保护内容的限制。

附图说明

[0037] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,并不构成对本发明的限定。在附图中:

[0038] 图1是根据本发明实施例的基于加速核的数据处理方法的方法流程图;

[0039] 图2是根据本发明实施例的滑动窗聚合阶段属性示意图;

[0040] 图3是现有的四个加速核的传统实现示意图;

[0041] 图4是根据本发明实施例的实现缓冲区融合的结构示意图;

[0042] 图5是根据本发明实施例的缓冲区融合实现的方法流程图;

[0043] 图6是根据本发明实施例的基于加速核的数据处理装置的结构框图。

具体实施方式

[0044] 为使本发明的目的、技术方案和优点更加清楚明白,下面结合实施方式和附图,对本发明做进一步详细说明。在此,本发明的示意性实施方式及其说明用于解释本发明,但并不作为对本发明的限定。

[0045] 应当注意,为了清楚的目的,附图和说明中省略了与本发明无关的、本领域普通技术人员已知的部件和处理的表示和描述。

[0046] 针对一种实施方式描述和/或示出的特征可以以相同或类似的方式在一个或多个其它实施方式中使用,与其它实施方式中的特征相组合,或替代其它实施方式中的特征。

[0047] 应该强调,术语“包括/包含”在本文使用时指特征、要素、步骤或组件的存在,但并不排除一个或多个其它特征、要素、步骤或组件的存在或附加。

[0048] 针对现有的数据流管理系统由于滑动窗聚合操作的固有属性,使得这些优化方式难以解决通用性和可扩展性的问题更清楚,下面对涉及到的可扩展性和通用性的概念说明如下,以便更好地理解本申请:

[0049] 1) 可扩展性:在现实应用中,滑动窗聚合操作的两个关键属性,range和slide的大小,会根据用户的不同需求而变化,这就需要加速芯片能够提供一定的可配置性。除此以外,根据不同的range和slide,在计算单次聚合操作时,需要的历史数据量也是变化,这就需要加速芯片有足够大的缓冲区来满足不同的需求。

[0050] 2) 通用性:滑动窗聚合操作有多种类型,例如:求极值、最平均、求协方差等,不可能实现所有的操作,这就需要能够实现基本操作,然后对基本操作进行组合来完成相应的操作,从而实现最大的通用性。

[0051] 针对现有的滑动窗聚合操作所存在的每个加速核都独自使用一个缓存区用于缓存需要重复使用的数据所存在的资源利用率低下的问题,在本例中提供了一种基于加速核的数据处理方法。

[0052] 图1是本申请的基于加速核的数据处理方法一个实施例的方法流程图。虽然本申请提供了如下述实施例或附图所示的方法操作步骤或装置结构,但基于常规或者无需创造性的劳动在所述方法或装置中可以包括更多或者更少的操作步骤或模块单元。在逻辑性上不存在必要因果关系的步骤或结构中,这些步骤的执行顺序或装置的模块结构不限于本申

请实施例描述及附图所示的执行顺序或模块结构。所述的方法或模块结构的在实际中的装置或终端产品应用时,可以按照实施例或者附图所示的方法或模块结构连接进行顺序执行或者并行执行(例如并行处理器或者多线程处理的环境,甚至分布式处理环境)。

[0053] 具体的,如图1所示,本申请一种实施例提供的一种基于加速核的数据处理方法可以包括如下步骤:

[0054] 步骤101:确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值。

[0055] 步骤102:将所述最大差值的预定倍数,作为缓存区的缓存量。

[0056] 步骤103:获取待处理数据。该待处理的数据例如为流式数据,或称数据流,但本发明并不限于此。

[0057] 步骤104:通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区。

[0058] 在上例中,多个加速核共用缓存区,缓存区的缓存量基于目标应用的滑动窗口大小与滑动距离的最大差值确定,这样使得多个加速核都基于该缓存区进行滑动窗口聚合操作,从而可以避免每个加速核独立使用一个缓存区所导致的缓存资源需求过大、缓存区浪费严重的问题,达到了有效降低缓存资源需求,提升现有的滑动窗聚合操作中缓存资源的利用率的技术效果。

[0059] 其中,上述的多个加速核可以并行运行也可以串行运行,具体采用串行运行的方式,还是采用并行运行的方式可以根据实际情况和需求确定,本申请对此不作限定。

[0060] 具体的,在实现的时候,因为多个加速核共用一个缓存区,为了实现数据的有序获取,可以为每个加速核增加缓冲区寻址访问模块,并配置相应的数据通路,即,每个加速核对应一个数据通路,用于从缓存区中获取数据。当待处理数据流入加速核时,待处理数据先流入缓存区,在通过预设的数据通路流入对应的加速核。

[0061] 进一步的,可以基于地址来获取数据,因此,通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作的步骤可以包括:

[0062] S1:加速核通过内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;

[0063] S2:加速核根据所述地址,从所述缓存区中获取历史数据;

[0064] S3:加速核基于所述历史数据进行滑动窗口聚合操作。

[0065] 在上述步骤101中,确定目标应用的滑动窗口大小与滑动距离的最大差值,可以是对历史处理过程进行分析,确定出可能出现的滑动窗口大小与滑动距离的最大差值;然后,将确定出的可能出现的滑动窗口大小与滑动距离的最大差值,作为所述目标应用的滑动窗口大小与滑动距离的最大差值。即,确定出range和slide的差值的最大值。

[0066] 在确定出range和slide的差值的最大值之后,可以根据是并行处理还是串行处理等,来确定最终的缓存区的缓存量,例如,串行可以以最大值作为缓存量,两两并行可以以最大值的两倍作为缓存量,三三并行可以以最大值的三倍作为缓存量等。具体最终缓存量可以根据实际需求确定,本申请对此不作限定。

[0067] 上述的聚合操作可以包括但不限于以下至少之一:求和、求最大值、求最小值、求平均值、数值排序。其中,核(kernel)是执行这些加速基本操作的硬件实现。

[0068] 下面结合一个具体实施例对上述方法进行说明,然而,值得注意的是,该具体实施例仅是为了更好地说明本申请,并不构成对本申请的不当限定。

[0069] 在本例中提出了一种通过融合所有核的缓冲区的缓存结构,来为多个并行或者串行执行的核提供数据缓存,其中,缓存结构大小的选择,可以根据n-for-N模型来决定。

[0070] 通过这种方式可以在不影响加速芯片的可扩展性的前提下,减少每个核需要的缓冲区大小,提高资源的利用率。具体的,提供了一种基于缓冲区融合的高效的缓存结构,具体采用如下原理:

[0071] 1) 根据常用的滑动窗聚合操作实现特定的加速核,这些加速核既可以并行运行也可以串行运行,

[0072] 2) 将需要实现的加速核的缓冲区融合起来,实现一个的缓存结构来统一管理;

[0073] 3) 改变之前的数据结构,数据到达之后,由缓存结构实现对每个核的数据供给。

[0074] 即,提供了统一的面向滑动窗聚合操作的历史数据管理缓存结构,可以大大减少在并行或者串行场景下,每个核需要的缓冲区大小,减少加速器的面积,提高了资源的利用率。

[0075] 基于上述的基于缓冲区融合的高效的缓存结构,可以按照如下方式进行数据核加速:

[0076] S1: 确定时间序列专用加速器的滑动窗聚合操作所需要的缓冲区大小;

[0077] S2: 将不同加速器核的缓冲区通过融合实现统一管理;

[0078] S3: 为每个核增加缓冲区寻址访问模块,并配置相应的数据通路;

[0079] S4: 根据特定需求,对缓冲区进行分段,并将分段后的缓冲区配置给不同的加速核使用。

[0080] 下面以一个具体实例进行说明:

[0081] 如图2所示是滑动窗聚合操作的阶段属性图,由图2可以看出滑动窗聚合这种操作,主要有两个属性变量范围(range)和间距(slide),其中,range减去slide决定了实现单个操作需要的缓冲区的大小,也就是历史数据的重用量。当slide等于1一个元组时,将面临最坏的情况,当然这时的数据重用率也是最高的。

[0082] 在本例中,就是为了减少在并行或串行情况下,整个加速器的资源利用率。首先,看一下现有的方式,如图3所示,是一个包含四个加速滑动窗聚合操作的核实现,其中,SWA表示滑动窗聚合加速核,按照现有的方式,假设range等于5,slide等于3,四个核都是实现求和操作,那么可以发现每个核都至少需要大小等于2的缓冲区,总共需要大小等于8的缓冲区。这样的实现当只有一个核处于利用状态时,存在的资源浪费是巨大的,同时存在可扩展性问题。当range和slide的差值增大时,需要同时增大每个核的缓冲区大小,这样资源上的浪费也将更大。

[0083] 针对现有方式所存在的可扩展性和资源利用率的问题,在本例中提供了一种基于缓冲区融合的缓存结构实现,如图4所示,四个核共用一个缓存,能够以最少的片上面积完成相应的可扩展性要求,其核心算法思想来源于,数据流中的n-of-N模型,对于此模型的数据流处理系统而言,可以解决所有长度小于等于N的滑动窗口问题。

[0084] 具体的,可以如图5所示,包括如下步骤:

[0085] S1: 首先通过对相关应用的执行历史进行分析,确定可能出现最大的range和

slide的差值,来确定n-of-N模型中的N值。

[0086] S2:根据N值可以确定cache的大小,其中,cache的大小为N的整数倍,这样可使得kernel可以同时处理多个时间序列。

[0087] S3:为每个kernel(核)增加取数单元。即,因为核要实现滑动窗聚合操作,计数单元是必不可少,因此,可以在此基础上加上时钟的推断,来确定历史数据在cache中的地址值。

[0088] S4:当数据流向kernel(核)时,首先流入cache中,然后再流向相应的kernel。

[0089] 在上例中,n-for-N模型,其中,n小于等于N,通过该模型,可以解决所有长度小于等于N的滑动窗口问题。

[0090] 在本说明书中,诸如第一和第二这样的形容词仅可以用于将一个元素或动作与另一元素或动作进行区分,而不必要求或暗示任何实际的这种关系或顺序。在环境允许的情况下,参照元素或部件或步骤(等)不应解释为局限于仅元素、部件、或步骤中的一个,而可以是元素、部件、或步骤中的一个或多个等。

[0091] 基于同一发明构思,本发明实施例中还提供了一种基于加速核的数据处理装置,如下面的实施例所述。由于基于加速核的数据处理装置解决问题的原理与基于加速核的数据处理方法相似,因此基于加速核的数据处理装置的实施可以参见基于加速核的数据处理方法的实施,重复之处不再赘述。以下所使用的,术语“单元”或者“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现,但是硬件,或者软件和硬件的组合的实现也是可能并被构想的。图6是本发明实施例的基于加速核的数据处理装置的一种结构框图,如图6所示,可以包括:确定模块601、生成模块602、获取模块603和处理模块604,下面对该结构进行说明。

[0092] 确定模块601,用于确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值;

[0093] 生成模块602,用于将所述最大差值的预定倍数,作为缓存区的缓存量;

[0094] 获取模块603,用于获取待处理数据;

[0095] 处理模块604,用于通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,其中,所述多个加速核共用所述缓存区。

[0096] 在一个实施方式中,当所述待处理数据流入所述加速核时,所述待处理数据先流入所述缓存区。

[0097] 在一个实施方式中,通过所述目标应用对应的多个加速核对所述待处理数据进行滑动窗口聚合操作,可以包括:所述加速核通过内置的取数单元,计算历史数据在缓存中的地址,其中,所述历史数据为待处理数据中本次数据处理过程中需要重用的数据;所述加速核根据所述地址,从所述缓存区中获取历史数据;所述加速核基于所述历史数据进行滑动窗口聚合操作。

[0098] 在一个实施方式中,确定目标应用对应的多个加速核的滑动窗口大小与滑动距离的最大差值,可以包括:对历史处理过程进行分析,确定多个加速核可能出现的滑动窗口大小与滑动距离的最大差值;将确定出的可能出现的滑动窗口大小与滑动距离的最大差值,作为所述目标应用的滑动窗口大小与滑动距离的最大差值。

[0099] 在一个实施方式中,聚合操作可以包括但不限于以下至少之一:求和、求最大值、

求最小值、求平均值、数值排序。

[0100] 在另外一个实施例中,还提供了一种软件,该软件用于执行上述实施例及优选实施方式中描述的技术方案。

[0101] 在另外一个实施例中,还提供了一种存储介质,该存储介质中存储有上述软件,该存储介质包括但不限于:光盘、软盘、硬盘、可擦写存储器等。

[0102] 从以上的描述中,可以看出,本发明实施例实现了如下技术效果:多个加速核共用缓存区,缓存区的缓存量基于目标应用的滑动窗口大小与滑动距离的最大差值确定,这样使得多个加速核都基于该缓存区进行滑动窗口聚合操作,从而可以避免每个加速核独立使用一个缓存区所导致的缓存资源需求过大、缓存区浪费严重的问题,达到了有效降低缓存资源需求,提升现有的滑动窗聚合操作的通用性和可扩展性的技术效果。

[0103] 显然,本领域的技术人员应该明白,上述的本发明实施例的各模块或各步骤可以用通用的计算装置来实现,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上,可选地,它们可以用计算装置可执行的程序代码来实现,从而,可以将它们存储在存储装置中由计算装置来执行,并且在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本发明实施例不限制于任何特定的硬件和软件结合。

[0104] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明实施例可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

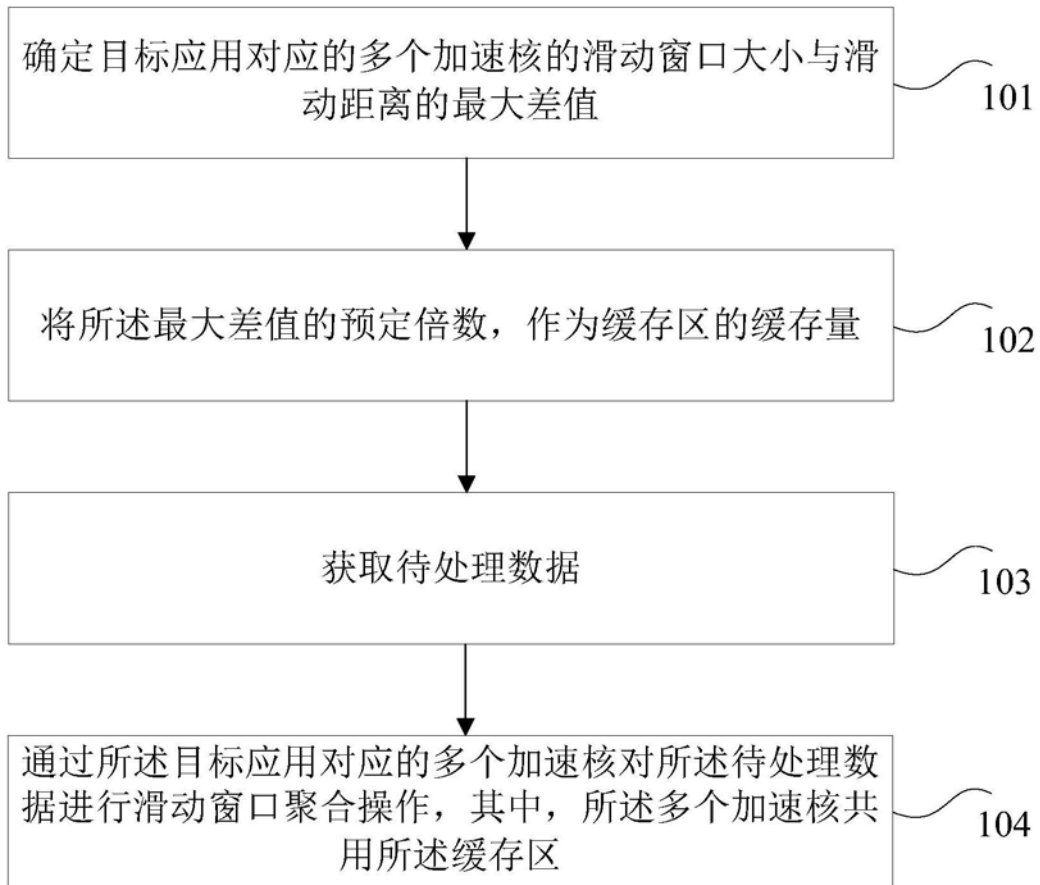


图1

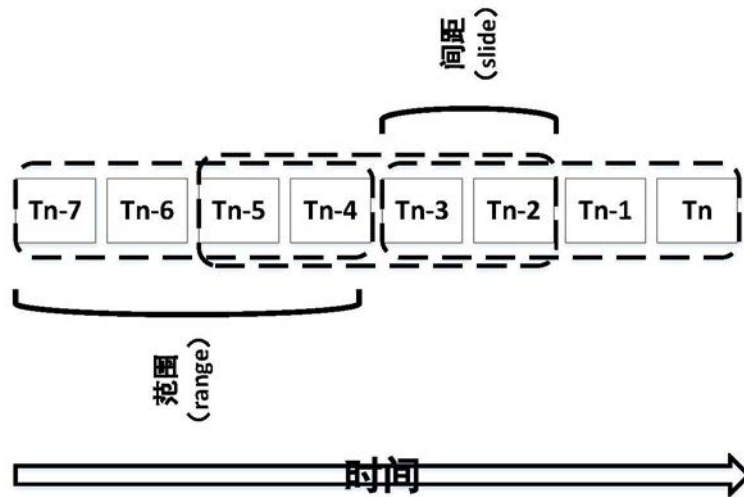


图2

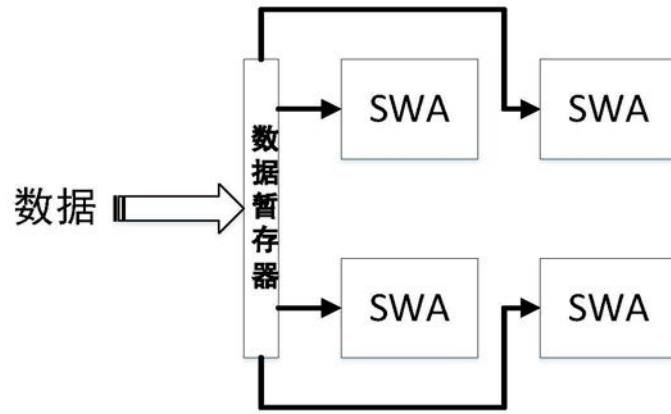


图3

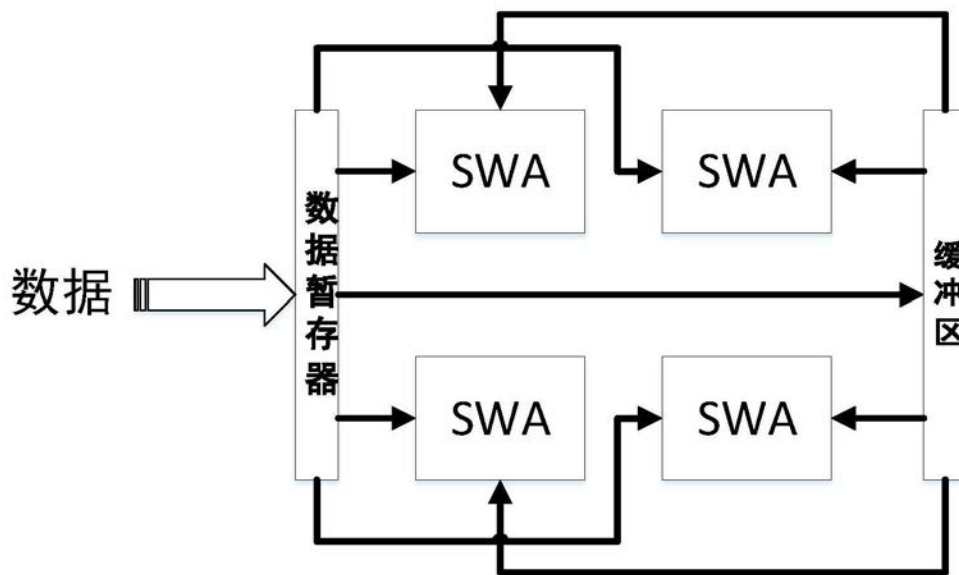


图4

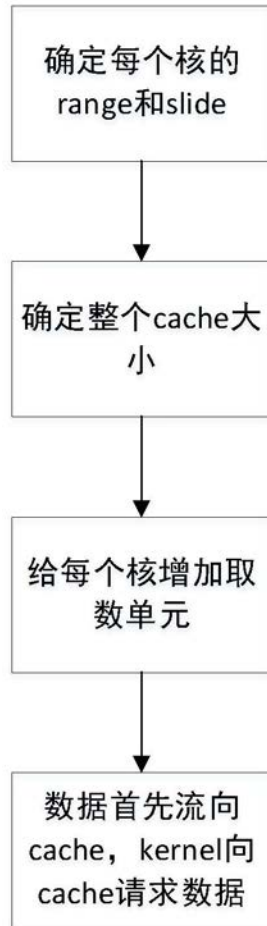


图5



图6