

(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
2020 年 9 月 17 日 (17.09.2020)



(10) 国际公布号
WO 2020/181820 A1

(51) 国际专利分类号:
G06F 12/0802 (2016.01)

(21) 国际申请号: PCT/CN2019/118426

(22) 国际申请日: 2019 年 11 月 14 日 (14.11.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910175754.3 2019 年 3 月 8 日 (08.03.2019) CN

(71) 申请人: 平安科技(深圳)有限公司(PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) [CN/CN];
中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(72) 发明人: 李桃(LI, Tao); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(74) 代理人: 深圳市明日今典知识产权代理有限公司(普通合伙)(SHENZHEN MINGRIJINDIAN INTELLECTUAL PROPERTY AGENCY FIRM (GENERAL)); 中国广东省深圳市南山区南头街道安乐社区关口二路 15 号智恒产业园 30 栋 4 层 406 室, Guangdong 518000 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK,

(54) Title: DATA CACHE METHOD AND APPARATUS, COMPUTER DEVICE AND STORAGE MEDIUM

(54) 发明名称: 数据缓存方法、装置、计算机设备和存储介质

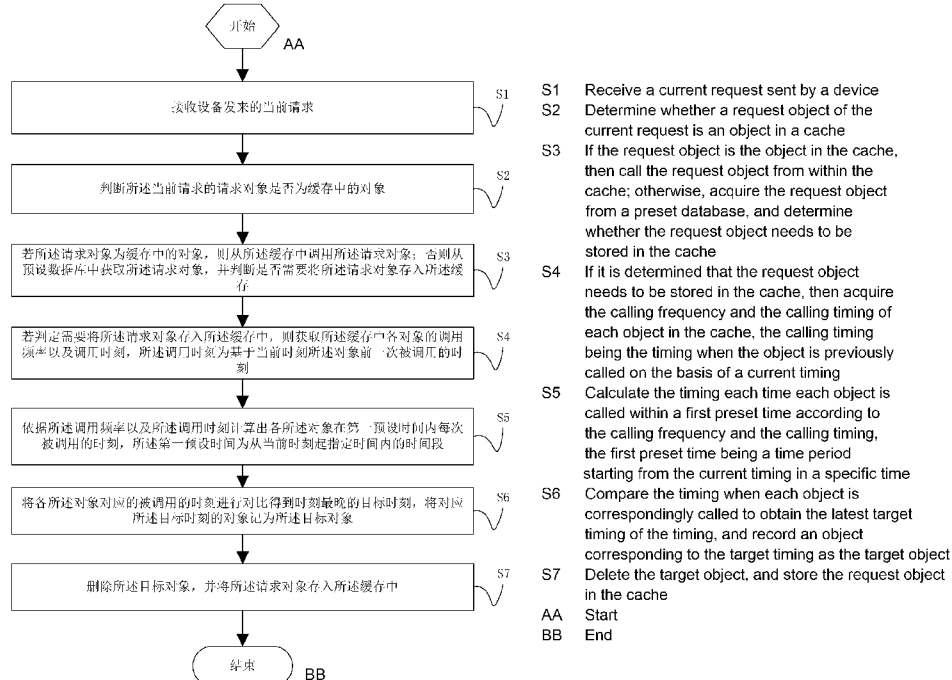


图 1

(57) Abstract: A data caching method and apparatus, a computer device and a storage medium set forth in the present application, which are used to cache data that has a fixed request frequency. The method comprises: receiving a current request sent by a device; determining whether a request object of the current request is an object in a cache; if the request object is the object in the cache, then calling the request object from within the cache; otherwise, acquiring the request object from a preset database, and determining whether the request object needs to be stored in the cache; and if it is determined that the request object needs to be stored in the cache,

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)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

deleting a target object in the cache, and storing the request object in the cache, the target object being the latest object in the timing when each object is called within a first preset time. The described method may increase the cache hit rate of the scene at which the call cache frequency is basically fixed, and make full use of cache resources. Compared to existing cache algorithms such as LRU, LFU, and FIFO, the present method costs less time.

(57) 摘要: 本申请提出的数据缓存方法、装置、计算机设备和存储介质, 用于缓存请求频率固定的数据, 其中方法包括: 接收设备发来的当前请求; 判断所述当前请求的请求对象是否为缓存中的对象; 若所述请求对象为缓存中的对象, 则从所述缓存中调用所述请求对象; 否则从预设数据库中获取所述请求对象, 并判断是否需要将所述请求对象存入所述缓存; 若判定需要将所述请求对象存入所述缓存中, 则删除所述缓存中的目标对象, 并将所述请求对象存入所述缓存中, 所述目标对象为在第一预设时间内各对象被调用的时刻当中最晚的对象, 该方法能够提高调用缓存频率基本固定场景的缓存命中率, 充分利用缓存资源, 与现有LRU、LFU以及FIFO等缓存算法相比, 耗费的时间代价更少。

数据缓存方法、装置、计算机设备和存储介质

本申请要求于 2019 年 3 月 8 日提交中国专利局、申请号为 201910175754.3, 申请名称为“数据缓存方法、装置、计算机设备和存储介质”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

本申请涉及到数据处理的技术领域, 特别是涉及到一种数据缓存方法、装置、计算机设备和存储介质。

背景技术

缓存通常为各设备系统的备份内存, 由于内存资源有限, 通常会设定一个固定内存值, 当缓存大小超过固定值后, 再按照 LRU(Least Recently Used)、LFU(Least Frequently Used) 或者 FIFO(First In First Out) 等算法对缓存进行清理。如规则引擎中, 会缓存规则引擎实例, 用于对设备请求的数据进行处理, 而不需要每次请求都重新创建对象, 当缓存达到设定的固定值时, 会根据缓存算法进行清理。

但是, 在物联网场景中, 设备上报数据或请求数据的频率往往是固定的, 而这种场景中若依据上述 LRU、LFU 或者 FIFO 等算法进行清理缓存并不适合, 缓存命中率并不高, 且缓存资源利用也不够充分, 鉴于上述各种缓存算法在物联网场景中的不足之处, 亟需一种针对调用缓存频率基本固定的场景的缓存方法。

技术问题

本申请的主要目的为提供一种数据缓存方法、装置、计算机设备和存储介质, 旨在解决在请求频率基本固定的场景中, 缓存资源利用不充分的技术问题。

技术解决方案

基于上述发明目的, 本申请提出一种数据缓存方法, 用于缓存请求频率固定的数据, 包括:

接收设备发来的当前请求;

判断所述当前请求的请求对象是否为缓存中的对象;

若所述请求对象为缓存中的对象, 则从所述缓存中调用所述请求对象; 否则从预设数据库中获取所述请求对象, 并判断是否需要将所述请求对象存入所述缓存;

若判定需要将所述请求对象存入所述缓存中, 则获取所述缓存中各对象的调用频率以及调用时刻, 所述调用时刻为基于当前时刻所述对象前一次被调用的时刻;

依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻, 所述第一预设时间为从当前时刻起指定时间内的时间段;

将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

删除所述目标对象，并将所述请求对象存入所述缓存中。

本申请还提供一种数据缓存装置，用于缓存请求频率固定的数据，包括：

5 接收请求单元，用于接收设备发来的当前请求；

判断对象单元，用于判断所述当前请求的请求对象是否为缓存中的对象；

调用对象单元，用于若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象；否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

10 获取时刻单元，用于若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；

计算时刻单元，用于依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；

15 对比时刻单元，用于将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

删除对象单元，用于删除所述目标对象，并将所述请求对象存入所述缓存中。

本申请还提供一种计算机设备，包括存储器和处理器，所述存储器存储有计算机可读指令，所述处理器执行所述计算机可读指令时实现上述方法的步骤。

20 本申请还提供了一种计算机可读存储介质，其上存储有计算机可读指令，所述计算机可读指令被处理器执行时实现上述方法的步骤。

有益效果

先判断请求的请求对象是否为缓存中的对象，若不是，则进一步判断需不需要将请求对象存入缓存中，若需要，则通过预设规则得到在第一预设时间内各对象被调用的时刻当
25 中最晚的对象，然后删除该对象并将请求对象存入缓存中，该方法能够提高请求频率基本固定场景的缓存命中率，充分利用缓存资源，与现有 LRU、LFU 以及 FIFO 等缓存算法相比，耗费的时间代价更少。

附图说明

图1 为本申请一实施例中数据缓存方法的步骤示意图；

30 图2 为本申请一实施例中数据缓存装置的结构示意框图；

图3 为本申请一实施例的计算机设备的结构示意框图。

本申请目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

本发明的最佳实施方式

应当理解，此处所描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

参照图 1，本实施例中的数据缓存方法，包括：

5 步骤 S1：接收设备发来的当前请求；

步骤 S2：判断所述当前请求的请求对象是否为缓存中的对象；

步骤 S3：若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

10 步骤 S4：若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；

步骤 S5：依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；

步骤 S6：将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

15 步骤 S7：删除所述目标对象，并将所述请求对象存入所述缓存中。

本实施例中，上述数据缓存方法用于缓存请求频率固定的数据，主要应用于请求频率固定的场景，例如物联网场景。上述当前请求至少包括请求头和请求体，其中请求头包括请求方法、版本、使用的协议等信息，请求体包括服务端的参数（如设备 ID）、请求内容等信息，依据请求内容得到请求对象，请求对象即为用于处理该请求的数据或者工具。例如，在一个处理设备状态的场景中，设备需要上报状态数据，这些数据包括温度、湿度、电池状态之类的等等，设备发出请求后，服务端接收到该请求并初始化一个规则引擎，这个规则引擎负责对上述数据进行分析和处理，这个例子中规则引擎可以为存在缓存中对象，本实施例中，为了便于表述，将当前系统接收到设备发送的请求记为当前请求。

如步骤 S2 所述，判断当前请求的请求对象是否为缓存中的对象，已知的是，由于接收
25 到当前请求之后，需要对请求进行响应，若请求的对象（用于响应的数据）存于缓存当中，那么就无需从预设的数据库查找再获取，而是直接从缓存中调用即可，这样效率会提高，但是存在请求对象不在缓存中的情况，例如，请求对象为 a，但是缓存中有 bcd 三个，那么这时便无法调用缓存，故而获取到请求之后，判断对应的请求对象是否为缓存中的对象，如在缓存中查找请求对象，若找到即可判定缓存中具有请求对象，这时直接调用该请求对象，如果在缓存中没有查找到请求对象，那么可判定缓存中不具有请求对象，这时需要从
30 系统的数据库中获取请求对象，该数据库为预设的、存有对应各个请求的请求对象（响应

数据)的数据库,为了后续系统能够提高效率,可进一步判断是否需要将该请求对象存入到缓存中,如根据预设的规则或算法,以耗费更少代价的基准来进一步判断是否需要将该请求对象存入缓存,需知,上述缓存为容量已经存储满的缓存,上述频率固定是指请求的频率是固定的,所以调用缓存的频率也是固定的,如请求 a 每隔一小时请求一次,请求 b 每隔 15 分钟请求一次。

本实施例中,当判定需要将请求对象存入到缓存中,即表明在后续系统运行,响应请求的过程中,从缓存中调用请求对象效率更好,耗费的代价更少。由于缓存的容量固定,若需要将请求对象存入缓存,那么则需要将缓存中原来的其中一个对象删除,具体可通过删除的目标对象来实现,该目标对象为在第一预设时间内,各对象下次被调用的时刻当中最晚的对象,删除这个目标对象之后,将上述请求对象存入缓存中,以便下次请求调用。

如上述步骤 S4-S7 所述,上述请求频率即为调用请求对象的频率,而在物联网场景中,一般都为调用频率基本固定的场景,故而可直接从系统中获取到缓存中各对象的调用频率,以及各对象上一次的调用时刻,此处的上一次调用时刻为基于当前时刻,对象前一次被调用的时刻。然后依据每个对象的调用频率以及上一次的调用时刻,计算出在第一预设时间内各次被调用的时刻,此处的第一预设时间为从当前时刻起指定时间内的时间段,最后将计算出来的时刻进行对比得到其中最晚的目标时刻,该目标时刻对应的对象即为上述目标对象,由此可将上述缓存方法称为最晚到达(Lastly Arrived,简称 LA)的缓存算法,该 LA 算法为本申请自定义的算法。

本实施例中利用如下公式,计算出各对象在第一预设时间内每次被调用的时刻:
$$T_i = \frac{1}{f_i} + t_i$$
其中, i 为上述缓存中的任一个对象, T_i 为对象 i 的调用时刻, f_i 为对象 i 的调用频率, t_i 为前一次调用对象 i 的时刻。

例如,缓存中的对象 A,对应的调用频率为 0.05 次/秒,对应上一次调用的时刻为 12:10:15,通过上述公式下次调用的时刻为 12:10:35,再下次调用的时刻为 12:10:55,如此计算出在第一预设时间内对象 A 的所有调用时刻,以此类推,计算出缓存中每个对象在第一预设时间内的所有调用时刻,然后将这些调用时刻进行对比,得到当中最晚的那个目标时刻,对应该目标时刻的对象即为应该删除的对象,为了便于描述将其记为目标对象。

本申请提供的缓存方法,在调用频率基本固定的场景中,由于以耗费更少代价为基准来进一步判断需要将请求对象存入缓存,而当需要将请求对象存入缓存时,替换的目标对象为第一预设时间内各对象下次被调用的时刻当中最晚的对象,使得缓存的命中率更高,且耗费代价更少,效率更高。

以下从缓存命中率以及耗费代价等方面来说明为何从缓存中删除的是上述目标对象:

首先定义代价 c 为一次缓存未命中所付出的额外时间开销（即对于某一请求，缓存中具有请求对象且进行调用的响应耗费时间，与缓存中没有请求对象需从数据库获取进而响应的耗费时间的差值）。总代价 S 则为一定时间内的所有开销之和。

那么在 t 时间内，缓存命中次数为 h ，缓存调用总次数为 m ，因此，命中率可以表示为：

$$5 \quad R = \frac{h}{m} * 100\%, \text{ 总代价可以表示为: } S = (m - h) * c. \text{ 假设被缓存的对象有 } 1, 2, 3, 4 \dots k, \text{ 对}$$

应被调用的频率依次为 $f_1, f_2, f_3, \dots, f_k$ ，上一次被调用时刻分别记为 $t_1, t_2, t_3 \dots t_k$ 。那么当时刻为 t 时， i 对象后续被调用的时刻为： $T_i = \frac{1}{f_i} + t_i$ ，因此该段时间内最晚时刻可以表示为：

$$T_{LA} = \max \left\{ \frac{1}{f_i} + t_i - t \right\}, i \in (1, 2, 3 \dots k), \text{ 即最后 } T_{LA} \text{ 对应的 } i \text{ 对象即为需要被清理出缓存的对象。}$$

10 现有的缓存算法中包括 LRU, LFU, FIFO 算法，分别可用公式表示需要被清理出缓存的对象： $T_{LRU} = \max\{t - t_i\}, i \in (1, 2, 3 \dots k)$ ， $T_{LFU} = \min\{[t * f_i]\}, i \in (1, 2, 3 \dots k)$ ， $T_{FIFO} = \max\{t_i\}, i \in (1, 2, 3 \dots k)$ 。将上述现有算法与本方案提供的 LA 算法进行比较，如下：

15 举例地，缓存有对象 a,b,c,d，对应的调用频率分别为 $f_a=1/5, f_b=1/4, f_c=1/3, f_d=1/2$ ，缓存中只能存放 3 个对象，上述代价 c 为 1，缓存中已有的对象集合用 U 表示。假如缓存中初始对象有 a,b,c，对象 d 在 $t=2$ 时准备加入缓存，那么在 t 时间内，每个时间点对应的命中次数 h 和总代价 S 如下表：

t	需要调用的对象	LRU/LFU			FIFO			LA		
		U	h	c	U	h	c	U	h	c
0		a, b, c	0	3	a, b, c	0	3	a, b, c	0	3
2	d	b, c, d	0	4	b, c, d	0	4	b, c, d	0	4
3	c	b, c, d	1	4	b, c, d	1	4	b, c, d	1	4
4	b, d	b, c, d	3	4	b, c, d	3	4	b, c, d	3	4
5	a	a, b, d	3	5	a, c, d	3	5	a, c, d	3	5
6	c, d	a, c, d	4	6	a, c, d	5	5	a, c, d	5	5
8	b, d	b, c, d	5	7	a, b, d	6	6	b, c, d	6	6
9	c	b, c, d	6	7	a, b, c	6	7	b, c, d	7	6
10	a, d	a, c, d	7	8	b, c, d	7	8	b, c, d	8	7
12	b, c, d	b, c, d	9	9	b, c, d	10	8	b, c, d	11	7
14	d	b, c, d	10	9	b, c, d	11	8	b, c, d	12	7

15	a, c	a, c, d	11	10	a, c, d	12	9	b, c, d	13	8
16	b, d	b, c, d	12	11	a, b, d	13	10	b, c, d	15	8
18	c, d	b, c, d	14	11	a, c, b	14	11	b, c, d	17	8
20	a, b	a, b, d	15	12	a, c, b	16	11	a, b, c	18	9

由表格可看出，与现有中的 LRU，LFU，FIFO 算法对比，本申请提供的 LA 算法在每个时间点缓存的命中次数最高，且耗费的总代价最少。

在一个实施例中，上述步骤 S3，包括：

5 步骤 S31：依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序；

步骤 S32：计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间，并记为第一耗时；

步骤 S33：计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间，并记为第二耗时；

10 步骤 S34：将所述第一耗时与所述第二耗时进行对比；

步骤 S35：若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；

步骤 S36：若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

15 本实施例中，判断是否需要将请求对象存入缓存通过上述步骤 S31-36 实现，如上述步骤 S31 所述，由于，预设的所有请求的请求频率固定，故而可获取每个请求的请求频率，再依据请求频率计算出在第二预设时间内各请求按请求时间的先后顺序进行排序的排列顺序。

在一个实施例中，上述步骤 S31，包括：

20 步骤 S310：依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；

步骤 S310：将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序；其中，通过以下公式计算得到请求 j 的所有请求时刻：

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

25 其中，j 为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求 j 的第 n 次请求的请求时刻， f_j 为请求 j 的请求频率， t_0 为所述第二预设时间内的初始时刻。

本实施例中, 由于在第二预设时间内所有请求的请求频率已知, 故可通过请求频率计算出该段时间内所有请求的请求时刻, 然后将对应每个请求时刻的请求按时间顺序排序得到上述排列顺序, 例如, 设初始时间 t_0 , 请求 a 的频率为 f_a , 请求 b 的频率 f_b , 请求 c 的频率为 f_c , 则下一次请求 a 到达时刻为 $H_a=t_0+\frac{1}{f_a}-(t_0 \bmod \frac{1}{f_a})$, 下一次请求 b 到达时刻为:

5 $H_b=t_0+\frac{1}{f_b}-(t_0 \bmod \frac{1}{f_b})$, 下一次请求 c 到达时刻为: $H_c=t_0+\frac{1}{f_c}-(t_0 \bmod \frac{1}{f_c})$, 以此类推, 所有请求 a 到达时刻可以表示为: $H_a(n)=t_0+n*\frac{1}{f_a}-(t_0 \bmod \frac{1}{f_a})$, n 为自然整数; 所有请求 b 到达时刻可以表示为: $H_b(n)=t_0+n*\frac{1}{f_b}-(t_0 \bmod \frac{1}{f_b})$; 所有请求 c 到达时刻可以表示为: $H_c(n)=t_0+n*\frac{1}{f_c}-(t_0 \bmod \frac{1}{f_c})$ 。这时可依据这些时刻的从小到大的排列顺序得到对应请求的排列顺序, 如将 H_a , H_b , H_c 中所有时刻从小到大排序, 并记录每个时刻到达的请求, 得到

10 序列 Q, 例如 $Q=\{(2,a), (3,b), (5,c), (6,ab), (8,a), (9,b)\}$, 其中(2,a)表示在时刻 2 时, 到达的请求有 a; (3,b) 表示在时刻 3 时, 到达的请求有 b; (6,ab)表示时刻 6 时, 到达的请求有 a,b。

如上述步骤 S32-S33 所述, 在不将请求对象存入缓存以及将请求对象存入缓存的情况下, 分别按上述排列顺序遍历对应的请求得到对应耗时, 并分别记为第一耗时, 以及第二

15 耗时, 从而可依据第一耗时以及第二耗时来判断是否需要将请求对象存入缓存。如上述例子中, 在第二预设时间内, 如在 9 分钟中内各请求的排列顺序为 a-b-c-ab-a-b, 其中当前的请求对象为 a, 缓存中只能放置 bc, 若没有将对象 a 存入缓存, 系统按上述排列顺序对遍历每个请求的耗时之和即为上述第一耗时, 值得注意的是, 从数据库获取对象比从缓存获取对象的耗时长, 且响应每个请求的耗时、从资料库或者从缓存取出的耗时可通过代码统

20 计得到, 故而可统计得到上述第一耗时。同理, 当将对象 a 存入缓存中, 则从缓存中删除排在上述顺序最后的对象 b, 那么上述请求对象为 a, 缓存中有 ac, 9 分钟内各请求的顺序为 a-b-c-ab-a-b, 系统按该排列顺序对遍历每个请求的耗时之和为的第二耗时。

如上述步骤 S34-S36 所述, 获得上述第一耗时以及第二耗时后, 将两者进行比较, 当第一耗时比第二耗时长, 即表明在没有将上述请求对象存入缓存的情况下, 系统耗费的时间更多, 效率更慢, 耗费的代价更大, 故而这时判定需要将请求对象存入缓存。反之, 当

25 第一耗时比第二耗时短, 说明没有将上述请求对象存入缓存的情况下, 系统耗费的时间更少, 代价更小, 故而判定不需要将请求对象存入缓存。

在另一实施例中, 可以首先计算出每个请求时刻对应的请求对象在存入缓存以及不存入缓存的情况下的耗时, 按照选择耗时短原则, 得到每次选择将请求对象存入或者不存入

30 缓存的优选策略, 然后在响应请求时, 按照该优选策略去执行, 用以判断每一次请求的请求对象是否需要存入缓存。

其详细过程如下：第一步：上述请求的排列顺序可记为 Q ，则序列 $Q=\{(t_1,u_1),(t_2,u_2),(t_3,u_3),\dots,(t_n,u_n)\}$ ，其中 t 表示时刻， u 表示该时刻到达的请求集合，如上述例子中， $t_1=2$ 时， $u_1=\{a\}$ ； $t_4=6$ 时， $u_4=\{ab\}$ 。对于任意二叉树（请求到达的时将请求对象分成两种情况，一种是存入缓存，另一种是不存入缓存）的节点 $S(i)$ 由五部分组成：父节点索引，左子节点索引，右子节点索引， $t(i)$ 时刻的总耗时 $c(i)$ 以及当前缓存集合 $m(i)$ 。

对于 $S(i)$ 的左子节点 $Sl(i+1)$ 表示对象集合 $A \in u(i)$ 且 $A \notin m(i)$ 时，集合 A 不使用缓存的情况，右子节点为 $Sr(i+1)$ 则是集合 A 都使用缓存的情况。一般情况下，1 个对象命中缓存和未命中的耗时相差较大，所以将命中缓存的耗时设为 0，未命中缓存的耗时设为 1。

第二步：令二叉树的根节点为 S_0 ，表示缓存的初始状态，此时 $c(0)=0, m(0)=\emptyset$ ，父节点索引，子节点索引均等于 null；依次取出序列 Q 中的元素，先判断 t_1 时刻， $u_1 \in m(0)$ 是否为真，如果为真，则 $Sr(1)$ 节点中 $c(1)=0, m(1)=m(0)$ ， $S(0)$ 右子节点索引指向 $Sr(1)$ ， $Sr(1)$ 父节点索引指向 $S(0)$ 。如果为假，即 $u_1 \notin m(0)$ ，分为两种情况：集合 u_1 使用缓存和不使用缓存，当使用缓存时， $S(0)$ 右子节点索引指向 $Sr(1)$ ， $Sr(1)$ 父节点索引指向 $S(0)$ ， $c(1)$ 等于 $c(0)$ 与 $\overline{u_1 \cap (u_1 \cap m(0))}$ 集合中元素个数 k 相加， $m(1)=\overline{u_1 \cap (u_1 \cap m(0))} \cup p$ ，其中集合 p 是使用上述 LA 算法淘汰掉 k 个对象后剩下的缓存实例集合。当不使用缓存时， $S(0)$ 右子节点索引指向 $Sr(1)$ ， $Sr(1)$ 父节点索引指向 $S(0)$ ， $c(1)$ 等于 $c(0)$ 与 $\overline{u_1 \cap (u_1 \cap m(0))}$ 集合中元素个数 k 相加。

第三步：重复上述步骤，直到 Q 中元素取完，最后得到一颗二叉树，在上述第二预设时间内，找出对应时间段内该二叉树中所有叶节点中 $c(n)$ 最小的节点，然后根据其父节点索引，向上检索，直到根节点 $S(0)$ ，此路径即为最短耗时路径。再根据左子节点是不使用缓存，右子节点使用缓存的规则，判断该路径上哪些时刻请求对象需要存入缓存，从而得到该时间段每个请求时刻的是否将请求对象存入缓存的最优策略。

在另一实施例中，上述步骤 S3，包括：

步骤 S31'：获取所述缓存中各对象的调用频率；

步骤 S32'：依据所述调用频率计算出下一个请求的对象，并记为下次对象；

步骤 S33'：判断所述下次对象是否为所述当前请求的请求对象；

步骤 S34'：若是，则判定需要将所述请求对象存入所述缓存中，否则判定不需要将所述请求对象存入所述缓存中。

除了通过上述步骤 S31-S36 来实现判断是否需要将请求对象存入缓存，还可通过下次调用的对象是否为当前请求对象来判定。如上述步骤 S31'-S34' 所述，从系统中获取缓存中每个对象的调用频率，根据各对象的调用频率以及上一次调用时刻，计算出下一次请求的

请求对象，即计算出排序在当前请求的下一次请求的对象，计算方法参照上述步骤 S31 所述，此处不再赘述。当得到下次对象时，进一步判断下次对象是否为当前请求的请求对象，若是，则判定需要将请求对象存入缓存，以便下次调用，若否，则判定不需要将请求对象存入缓存中。

5 在一个实施例中，上述步骤 S2，包括：

步骤 S21：依据所述当前请求识别出所述请求对象；

步骤 S22：将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；

10 步骤 S23：若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

本实施例中，已知请求对象即为用于处理该请求的数据或者工具，由于每个请求均包括请求头、请求体，而请求体中包含有服务端的参数（如设备 ID）、请求内容等信息，可从请求内容获得处理该内容的信息，根据这些信息识别出当前请求的请求对象，然后将请求对象与缓存列表中的对象进行对比，缓存列表为存储在缓存中所有对象的列表，当在缓存
15 列表中找到与请求对象一致的对象，即可判定请求对象为缓存中的对象，当在缓存列表中找不到与请求对象一致的对象，则表明请求对象不为缓存中的对象。

参照图 2，本实施例中数据缓存装置，包括：

接收请求单元 100，用于接收设备发来的当前请求；

判断对象单元 200，用于判断所述当前请求的请求对象是否为缓存中的对象；

20 调用对象单元 300，用于若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

25 获取时刻单元 400，用于若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；

计算时刻单元 500，用于依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；

对比时刻单元 600，用于将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

30 删除对象单元 700，用于删除所述目标对象，并将所述请求对象存入所述缓存中。

本实施例中利用如下公式，计算出各对象在第一预设时间内下次被调用的时刻：

$T_i = \frac{1}{f_i} + t_i$, 其中, i 为上述缓存中的任一个对象, T_i 为对象 i 的调用时刻, f_i 为对象 i 的调用频率, t_i 为前一次调用对象 i 的时刻。

在一个实施例中, 上述调用对象单元 300, 包括:

5 计算顺序子单元, 用于依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序;

第一耗时子单元, 用于计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间, 并记为第一耗时;

第二耗时子单元, 用于计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间, 并记为第二耗时;

10 对比耗时子单元, 用于将所述第一耗时与所述第二耗时进行对比;

第一判定子单元, 用于所述第一耗时比所述第二耗时长时, 则判定需要将所述请求对象存入所述缓存;

第二判定子单元, 用于所述第一耗时比所述第二耗时短时, 则判定不需要将所述请求对象存入所述缓存。

15 在一个实施例中, 上述计算顺序子单元, 包括:

计算时刻模块, 用于依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻;

排序请求模块, 用于将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序;

20 其中, 通过以下公式计算得到请求 j 的所有请求时刻:

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

其中, j 为所述预设的所有请求中的任一个请求, $H_j(n)$ 为请求 j 的第 n 次请求的请求时刻, f_j 为请求 j 的请求频率, t_0 为所述第二预设时间内的初始时刻。

在另一实施例中, 上述调用对象单元 300, 包括:

调用频率子单元, 用于获取所述缓存中各对象的调用频率;

25 计算对象子单元, 用于依据所述调用频率计算出下一个请求的对象, 并记为下次对象;

判断对象子单元, 用于判断所述下次对象是否为所述当前请求的请求对象;

判定存入子单元, 用于判断所述下次对象为所述当前请求的请求对象, 则判定需要将所述请求对象存入所述缓存中, 否则判定不需要将所述请求对象存入所述缓存中。

在一个实施例中, 上述判断对象单元 200, 包括:

30 识别对象子单元, 用于依据所述当前请求识别出所述请求对象;

对比对象子单元，用于将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；

判定缓存子单元，用于若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

5 参照图3，本申请实施例中还提供一种计算机设备，该计算机设备可以是服务器，其内部结构可以如图3所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中，该计算机设计的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机可读指令和数据库。该内存储器为非易失性存储介质中的操作系统和计算机可读指令的运行提供
10 环境。该计算机设备的数据库用于存储上述调用缓存对象所需的所有数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机可读指令被处理器执行时以实现一种数据缓存方法。

上述处理器执行上述数据缓存方法的步骤：接收设备发来的当前请求；判断所述当前请求的请求对象是否为缓存中的对象；若所述请求对象为缓存中的对象，则从所述缓存中
15 调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；将各所述对象对应的被调用
20 的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；删除所述目标对象，并将所述请求对象存入所述缓存中。

在一个实施例中，上述判断是否需要将所述请求对象存入所述缓存的步骤，包括：依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序；计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间，并记为第一耗时；计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各
25 请求所需的时间，并记为第二耗时；将所述第一耗时与所述第二耗时进行对比；若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

在一个实施例中，上述依据预设的所有请求的请求频率计算得到在第二预设时间内各
30 请求按时间顺序排序的排列顺序的步骤，包括：依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；将对应每个所述请求时刻的请求按时

间顺序进行排序得到所述排列顺序；其中，通过以下公式计算得到请求j的所有请求时刻： $H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$ ；其中，j为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求j的第n次请求的请求时刻， f_j 为请求j的请求频率， t_0 为所述第二预设时间内的初始时刻。

- 5 在一个实施例中，上述判断是否需要将所述请求对象存入所述缓存的步骤，包括：获取所述缓存中各对象的调用频率；依据所述调用频率计算出下一个请求的对象，并记为下次对象；判断所述下次对象是否为所述当前请求的请求对象；若是，则判定需要将所述请求对象存入所述缓存中，否则判定不需要将所述请求对象存入所述缓存中。

- 10 在一个实施例中，上述判断所述当前请求的请求对象是否为缓存中的对象的步骤，包括：依据所述当前请求识别出所述请求对象；将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

- 15 本领域技术人员可以理解，图3中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的计算机设备的限定。

- 20 本申请一实施例还提供一种计算机可读存储介质，所述计算机可读存储介质，例如为非易失性的计算机可读存储介质，或者为易失性的计算机可读存储介质，其上存储有计算机可读指令，计算机可读指令被处理器执行时实现一种数据缓存方法，具体为：接收设备发来的当前请求；判断所述当前请求的请求对象是否为缓存中的对象；若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；删除所述目标对象，并将所述请求对象存入所述缓存中。

- 30 在一个实施例中，上述判断是否需要将所述请求对象存入所述缓存的步骤，包括：依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序；计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间，并记为第一耗时；计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各

请求所需的时间，并记为第二耗时；将所述第一耗时与所述第二耗时进行对比；若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

5 在一个实施例中，上述依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按时间顺序排序的排列顺序的步骤，包括：依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序；其中，通过以下公式计算得到请求j的所有请求时刻： $H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$ ；其中，j为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求j的第n次请求的请求时刻， f_j 为请求j的请求频率， t_0 为所述第二预设时间内的初始时刻。

10 在一个实施例中，上述判断所述当前请求的请求对象是否为缓存中的对象的步骤，包括：依据所述当前请求识别出所述请求对象；将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

15 以上所述仅为本申请的优选实施例，并非因此限制本申请的专利范围，凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本申请的专利保护范围内。

权利要求书

1、一种数据缓存方法，用于缓存请求频率固定的数据，其特征在于，包括：

接收设备发来的当前请求；

判断所述当前请求的请求对象是否为缓存中的对象；

5 若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；

10 依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；

将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

删除所述目标对象，并将所述请求对象存入所述缓存中。

15 2、根据权利要求1所述的数据缓存方法，其特征在于，所述依据所述调用频率以及所述调用时刻计算出各所述对象在所述第一预设时间内每次被调用的时刻的步骤，包括：

利用以下公式计算各所述对象在所述第一预设时间内每次被调用的时刻：

$$T_i = \frac{1}{f_i} + t_i$$

其中，i 为所述缓存中的任一个对象， T_i 为对象 i 的调用时刻， f_i 为对象 i 的调用频率， t_i 为前一次调用对象 i 的时刻。

20 3、根据权利要求1所述的数据缓存方法，其特征在于，所述判断是否需要将所述请求对象存入所述缓存的步骤，包括：

依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序；

计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间，并记为第一耗时；

25 计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间，并记为第二耗时；

将所述第一耗时与所述第二耗时进行对比；

若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；

若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

4、根据权利要求3所述的数据缓存方法，其特征在于，所述依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序的步骤，包括：

依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；

5 将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序；

其中，通过以下公式计算得到请求j的所有请求时刻：

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

其中，j为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求j的第n次请求的请求时刻， f_j 为请求j的请求频率， t_0 为所述第二预设时间内的初始时刻。

10 5、根据权利要求1所述的数据缓存方法，其特征在于，所述判断是否需要将所述请求对象存入所述缓存的步骤，包括：

获取所述缓存中各对象的调用频率；

依据所述调用频率计算出下一个请求的对象，并记为下次对象；

判断所述下次对象是否为所述当前请求的请求对象；

15 若是，则判定需要将所述请求对象存入所述缓存中，否则判定不需要将所述请求对象存入所述缓存中。

6、根据权利要求1所述的数据缓存方法，其特征在于，所述判断所述当前请求的请求对象是否为缓存中的对象的步骤，包括：

依据所述当前请求识别出所述请求对象；

20 将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；

若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

7、一种数据缓存装置，用于缓存请求频率固定的数据，其特征在于，包括：

接收请求单元，用于接收设备发来的当前请求；

25 判断对象单元，用于判断所述当前请求的请求对象是否为缓存中的对象；

调用对象单元，用于若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

30 获取时刻单元，用于若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的

时刻;

计算时刻单元, 用于依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻, 所述第一预设时间为从当前时刻起指定时间内的时间段;

对比时刻单元, 用于将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻, 将对应所述目标时刻的对象记为所述目标对象;

删除对象单元, 用于删除所述目标对象, 并将所述请求对象存入所述缓存中。

8、根据权利要求 7 所述的数据缓存装置, 其特征在于, 所述调用对象单元包括:

计算顺序子单元, 用于依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按时间顺序排序的排列顺序;

10 第一耗时子单元, 用于计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间, 并记为第一耗时;

第二耗时子单元, 用于计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间, 并记为第二耗时;

对比耗时子单元, 用于将所述第一耗时与所述第二耗时进行对比;

15 第一判定子单元, 用于所述第一耗时比所述第二耗时长时, 则判定需要将所述请求对象存入所述缓存;

第二判定子单元, 用于所述第一耗时比所述第二耗时短时, 则判定不需要将所述请求对象存入所述缓存。

9、根据权利要求 8 所述的数据缓存装置, 其特征在于, 所述计算顺序子单元, 包括:

20 计算时刻模块, 用于依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻;

排序请求模块, 用于将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序;

其中, 通过以下公式计算得到请求 j 的所有请求时刻:

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

25 其中, j 为所述预设的所有请求中的任一个请求, $H_j(n)$ 为请求 j 的第 n 次请求的请求时刻, f_j 为请求 j 的请求频率, t_0 为所述第二预设时间内的初始时刻。

10、根据权利要求 7 所述的数据缓存装置, 其特征在于, 所述调用对象单元, 包括:

调用频率子单元, 用于获取所述缓存中各对象的调用频率;

计算对象子单元, 用于依据所述调用频率计算出下一个请求的对象, 并记为下次对象;

30 判断对象子单元, 用于判断所述下次对象是否为所述当前请求的请求对象;

判定存入子单元,用于判断所述下次对象为所述当前请求的请求对象,则判定需要将所述请求对象存入所述缓存中,否则判定不需要将所述请求对象存入所述缓存中。

11、根据权利要求7所述的数据缓存装置,其特征在于,所述判断对象单元,包括:
识别对象子单元,用于依据所述当前请求识别出所述请求对象;

5 对比对象子单元,用于将所述请求对象与缓存列表中各对象进行对比,所述缓存列表为存储在所述缓存中所有对象的列表;

判定缓存子单元,用于若所述请求对象与所述缓存列表中的对象一致,则判定所述请求对象为所述缓存中的对象;否则判定所述请求对象不为所述缓存中的对象。

12、一种计算机设备,包括存储器和处理器,所述存储器存储有计算机可读指令,其特征
10 在于,所述处理器执行所述计算机可读指令时实现数据缓存方法,该数据缓存方法包括:

接收设备发来的当前请求;

判断所述当前请求的请求对象是否为缓存中的对象;

若所述请求对象为缓存中的对象,则从所述缓存中调用所述请求对象,否则从预设数
15 据库中获取所述请求对象,并判断是否需要将所述请求对象存入所述缓存;

若判定需要将所述请求对象存入所述缓存中,则获取所述缓存中各对象的调用频率以及调用时刻,所述调用时刻为基于当前时刻所述对象前一次被调用的时刻;

依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻,所述第一预设时间为从当前时刻起指定时间内的时间段;

20 将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻,将对应所述目标时刻的对象记为所述目标对象;

删除所述目标对象,并将所述请求对象存入所述缓存中。

13、根据权利要求12所述的计算机设备,其特征在于,所述判断是否需要将所述请求对象存入所述缓存的步骤,包括:

25 依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序;

计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间,并记为第一耗时;

30 计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间,并记为第二耗时;

将所述第一耗时与所述第二耗时进行对比;

若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；

若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

14、根据权利要求 13 所述的计算机设备，其特征在于，所述依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序的步骤，包括：

5 依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；

将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序；

其中，通过以下公式计算得到请求 j 的所有请求时刻：

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

10 其中，j 为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求 j 的第 n 次请求的请求时刻， f_j 为请求 j 的请求频率， t_0 为所述第二预设时间内的初始时刻。

15、根据权利要求 12 所述的计算机设备，其特征在于，所述判断是否需要将所述请求对象存入所述缓存的步骤，包括：

获取所述缓存中各对象的调用频率；

依据所述调用频率计算出下一个请求的对象，并记为下次对象；

15 判断所述下次对象是否为所述当前请求的请求对象；

若是，则判定需要将所述请求对象存入所述缓存中，否则判定不需要将所述请求对象存入所述缓存中。

16、根据权利要求 12 所述的计算机设备，其特征在于，所述判断所述当前请求的请求对象是否为缓存中的对象的步骤，包括：

20 依据所述当前请求识别出所述请求对象；

将所述请求对象与缓存列表中各对象进行对比，所述缓存列表为存储在所述缓存中所有对象的列表；

若所述请求对象与所述缓存列表中的对象一致，则判定所述请求对象为所述缓存中的对象；否则判定所述请求对象不为所述缓存中的对象。

25 17、一种计算机可读存储介质，其上存储有计算机可读指令，其特征在于，所述计算机可读指令被处理器执行时实现数据缓存方法，该数据缓存方法包括：

接收设备发来的当前请求；

判断所述当前请求的请求对象是否为缓存中的对象；

30 若所述请求对象为缓存中的对象，则从所述缓存中调用所述请求对象，否则从预设数据库中获取所述请求对象，并判断是否需要将所述请求对象存入所述缓存；

若判定需要将所述请求对象存入所述缓存中，则获取所述缓存中各对象的调用频率以及调用时刻，所述调用时刻为基于当前时刻所述对象前一次被调用的时刻；

依据所述调用频率以及所述调用时刻计算出各所述对象在第一预设时间内每次被调用的时刻，所述第一预设时间为从当前时刻起指定时间内的时间段；

- 5 将各所述对象对应的被调用的时刻进行对比得到时刻最晚的目标时刻，将对应所述目标时刻的对象记为所述目标对象；

删除所述目标对象，并将所述请求对象存入所述缓存中。

18、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述判断是否需要将所述请求对象存入所述缓存的步骤，包括：

- 10 依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序；

计算在所述请求对象没有存入所述缓存的情况下按所述排列顺序遍历各请求所需的时间，并记为第一耗时；

- 15 计算在所述请求对象存入所述缓存的情况下按所述排序顺序遍历各请求所需的时间，并记为第二耗时；

将所述第一耗时与所述第二耗时进行对比；

若所述第一耗时比所述第二耗时长，则判定需要将所述请求对象存入所述缓存；

若所述第一耗时比所述第二耗时短，则判定不需要将所述请求对象存入所述缓存。

- 19、根据权利要求 18 所述的计算机可读存储介质，其特征在于，所述依据预设的所有请求的请求频率计算得到在第二预设时间内各请求按请求时间排序的排列顺序的步骤，包括：

依据预设的所有请求的请求频率计算得到在所述第二预设时间内各所述请求的所有请求时刻；

将对应每个所述请求时刻的请求按时间顺序进行排序得到所述排列顺序；

- 25 其中，通过以下公式计算得到请求 j 的所有请求时刻：

$$H_j(n) = t_0 + n * \frac{1}{f_j} - (t_0 \bmod \frac{1}{f_j})$$

其中，j 为所述预设的所有请求中的任一个请求， $H_j(n)$ 为请求 j 的第 n 次请求的请求时刻， f_j 为请求 j 的请求频率， t_0 为所述第二预设时间内的初始时刻。

20、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述判断是否需要将所述请求对象存入所述缓存的步骤，包括：

- 30 获取所述缓存中各对象的调用频率；

依据所述调用频率计算出下一个请求的对象，并记为下次对象；

判断所述下次对象是否为所述当前请求的请求对象；

若是，则判定需要将所述请求对象存入所述缓存中，否则判定不需要将所述请求对象存入所述缓存中。

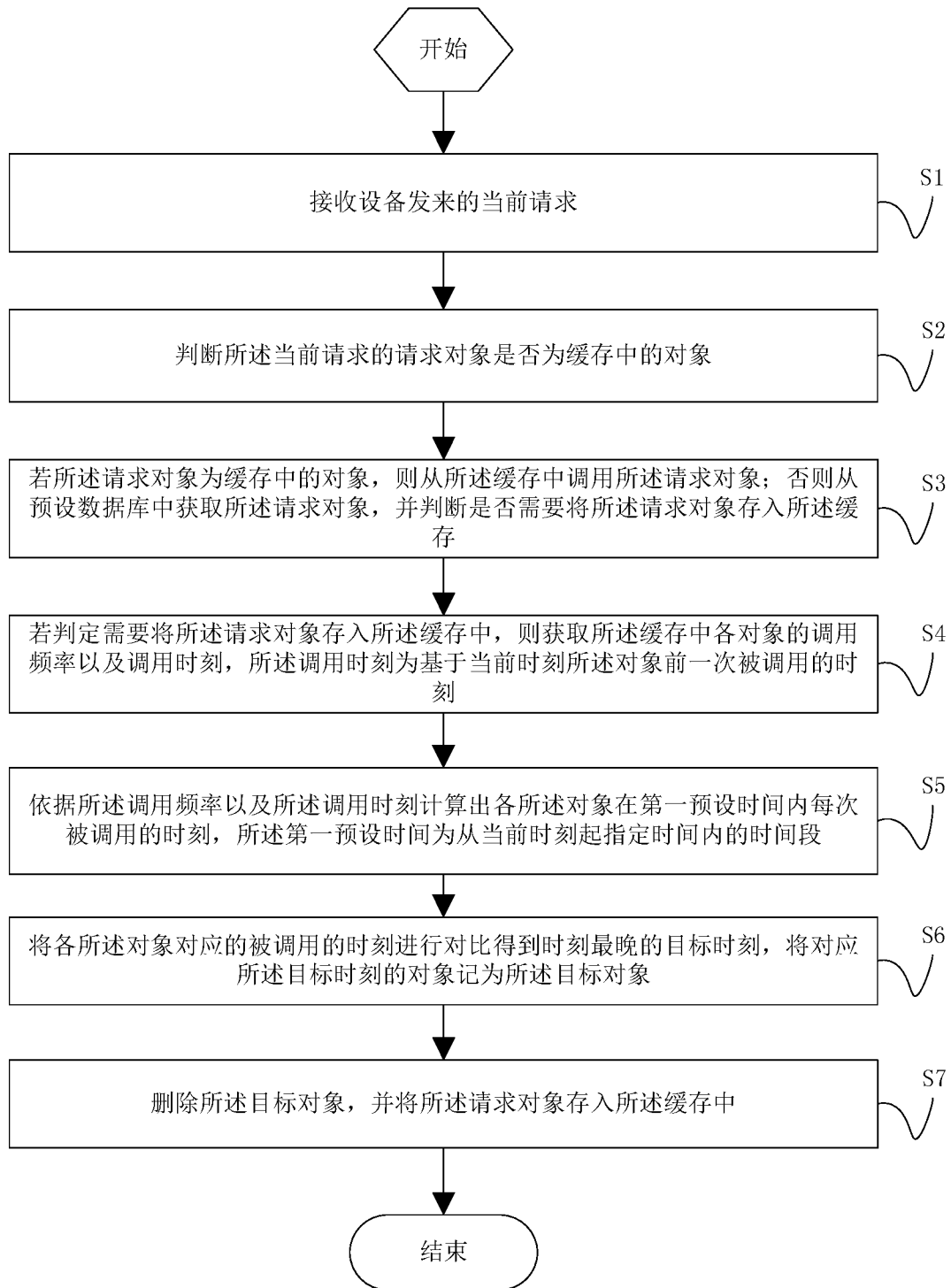


图 1

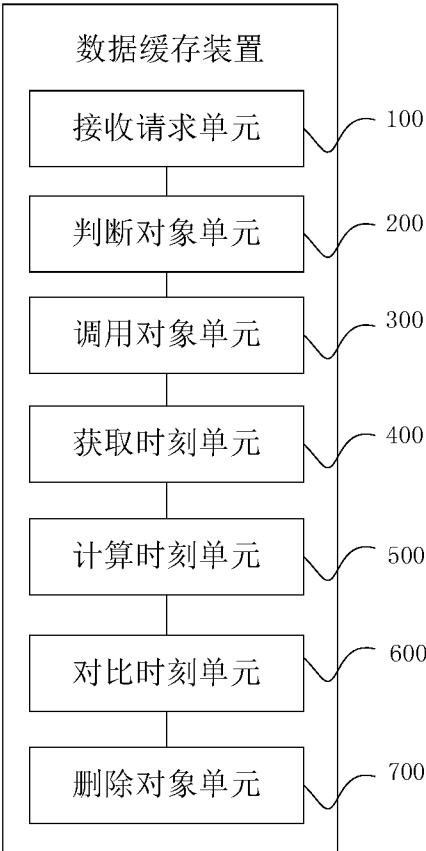


图 2

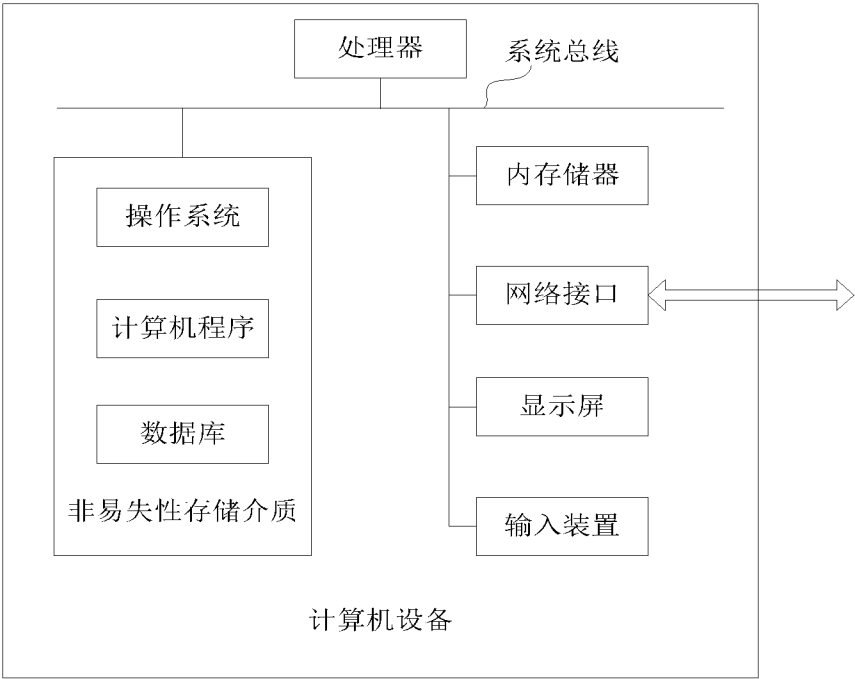


图 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/118426

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/0802(2016.01)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; H04L

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; CNKI; WPI; EPODOC; IEEE: 日志, 问题, 故障, 服务链, 多个, 若干个, log+, problem, trouble, service, link, multi+, several

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 110018969 A (PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) 16 July 2019 (2019-07-16) claims 1-10	1-20
A	CN 103544119 A (GPGC ELECTRIC POWER RESEARCH INSTITUTE) 29 January 2014 (2014-01-29) description, paragraphs [0030]-[0063]	1-20
A	CN 102223681 A (ZTE CORPORATION) 19 October 2011 (2011-10-19) entire document	1-20
A	CN 106888262 A (BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS) 23 June 2017 (2017-06-23) entire document	1-20
A	US 2017185645 A1 (SYBASE, INC.) 29 June 2017 (2017-06-29) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“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)

“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

“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

“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

“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

“&” document member of the same patent family

Date of the actual completion of the international search

14 January 2020

Date of mailing of the international search report

01 February 2020

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/118426

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	110018969	A	16 July 2019	None			
CN	103544119	A	29 January 2014	None			
CN	102223681	A	19 October 2011	WO	2011131066	A1	27 October 2011
CN	106888262	A	23 June 2017	None			
US	2017185645	A1	29 June 2017	None			

国际检索报告

国际申请号

PCT/CN2019/118426

A. 主题的分类

G06F 12/0802 (2016.01) 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; H04L

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; CNKI; WPI; EPDOC; IEEE: 日志, 问题, 故障, 服务链, 多个, 若干个, log+, problem, trouble, service, link, multi+, several

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
PX	CN 110018969 A (平安科技深圳有限公司) 2019年 7月 16日 (2019 - 07 - 16) 权利要求1-10	1-20
A	CN 103544119 A (广东电网公司电力科学研究院) 2014年 1月 29日 (2014 - 01 - 29) 说明书第[0030]-[0063]段	1-20
A	CN 102223681 A (中兴通讯股份有限公司) 2011年 10月 19日 (2011 - 10 - 19) 全文	1-20
A	CN 106888262 A (北京邮电大学) 2017年 6月 23日 (2017 - 06 - 23) 全文	1-20
A	US 2017185645 A1 (SYBASE, INC.) 2017年 6月 29日 (2017 - 06 - 29) 全文	1-20

☐ 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

“&” 同族专利的文件

国际检索实际完成的日期

2020年 1月 14日

国际检索报告邮寄日期

2020年 2月 1日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (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

授权官员

张彩霞

电话号码 86-(10)-53961804

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/118426

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	110018969	A	2019年 7月 16日	无	
CN	103544119	A	2014年 1月 29日	无	
CN	102223681	A	2011年 10月 19日	W0 2011131066 A1	2011年 10月 27日
CN	106888262	A	2017年 6月 23日	无	
US	2017185645	A1	2017年 6月 29日	无	