王甫棣,姜立鹏,姚燕. 北京全球信息系统中心的数据缓存功能优化. 应用气象学报,2014,25(2):242-248.

北京全球信息系统中心的数据缓存功能优化

王甫棣* 姜立鹏 姚 燕
(国家气象信息中心,北京 100081)

摘 要

世界气象组织信息系统(WMO Information System,WIS)是一个支撑全球气象数据交换共享的通用信息服务平台,北京全球信息系统中心作为 WIS 的核心功能中心之一,必须缓存最近 24 h 内的 WMO 全球交换数据以提供高效的数据访问服务。为了检查收集到的全球交换数据的合法性,需要校验每条数据是否存在与之匹配的元数据,这些元数据信息以关系型数据库方式存储在北京全球信息系统中心中。由于每日接收到的 WMO 全球交换数据文件个数多且收集时间分布不均,大量频繁的数据库查询操作导致处理性能下降,特别在数据密集收集的情况下容易产生较大延迟,直接影响业务的实时性。设计并实现一种基于内存对象缓存的应用优化现有通过数据库查询的校验方式,实现在内存中一次性载入元数据信息,并在内存中完成缓存数据校验的操作,以此来减少磁盘的读写访问,提升处理效率。此外,还通过多线程方法实现与缓存数据相关的功能,使该方案具有良好的扩展性。实际应用表明:数据缓存功能优化后能满足实时业务性能要求。

关键词: 数据缓存; 内存对象; 北京全球信息系统中心

引 言

世界气象组织信息系统(WMO Information System, WIS)是世界气象组织(WMO)正在组织开发的综合、通用的信息服务平台,用以支撑 WMO 各项计划以及相关国际组织和计划的数据交换和共享[1]。全球信息系统中心(GISC)是 WIS 的核心功能中心[2],承担全球交换资料的收集和分发,提供对 WIS 全部数据的发现和访问服务。中国气象局国家气象信息中心(NMIC)目前是全球电信系统(GTS)主干通信网的亚洲区域通信枢纽(RTH),建成北京 GISC 是中国气象局的既定目标,也是巩固和提升中国气象局在 WMO 通信网络及信息系统中的地位和影响力的重要举措[3]。

北京 GISC 是全球首批业务运行的 GISC 之一,其服务系统可收集责任区内提供全球交换的数据和产品,与其他 GISC 交换全球数据,提供责任区内的数据收集或产品中心(DCPC)和国家中心(NC)对 WIS 全部数据的发现和访问服务。包括国家气象中心、国家卫星气象中心、国家气候中心在内的中国气象局内部 DCPC 于 2012 年获得第 64 届 WMO 执行理事会批准投入业务试运行,所有提供的数据和产品都可以通过北京 GISC 进行发现、访问和检索服务。

按照 WIS/GISC 要求,每个 GISC 须至少缓存 24 h 内的全球交换数据,若授权用户通过元数据浏览或者检索发现到某些关心的数据条目,可以链接形式显示出这些缓存的本地数据。此外,所有这些数据资料均需要进行文件名和传输格式的规范化处理。

1 GISC 数据缓存服务

1.1 数据发现、访问和检索服务

当前,WIS 分两部分并行实施[4]:一部分将继续发展 GTS,基于实时推送机制,进一步改进高时效和关键业务的资料、产品的服务提供,包括警报;另一部分则是通过 Internet 提供数据发现、访

2012-11-05 收到, 2013-11-22 收到再改稿。
资助项目:公益性行业(气象)科研专项(GYHY200906057)
* email: wangfd@cma.gov.cn

问和检索(DAR)服务,并基于请求/应答的拉取机制提供灵活及时的数据获取。GISC 中的缓存数据下载是 DAR 服务的一部分内容,它是 DAR 服务对于 24 h 全球交换数据访问的具体实施。

每个 GISC 需要维护包括全球交换资料在内的 WIS 服务数据的元数据目录并提供访问,以支持 DAR 功能,包括上传、修改和删除元数据,用户发现和访问元数据,元数据同步[5]等功能。

1.2　DAR 元数据

参考国际所公认的空间信息元数据标准[6]引入气象元数据[7-8],使用元数据来描述气象数据是解决气象数据共享的理想办法[9]。目前 WMO 各 GISC 发布的气象元数据均须遵循 WMO 核心元数据标准,但由于不同 WMO 成员对元数据描述内容的需求和利用元数据提供应用服务方面存在差异,各成员各自创建满足自身需求的气象元数据,但所有在 WIS 中注明来源的资料均将根据 ISO 标准由相关元数据定义。北京 GISC 系统中的元数据采用 XML 格式记录元数据,目前已遵循 WMO 最新的核心元数据标准(1.2 版本),主要由文件标识、语言、字符集等 15 个主要元素组成,可以描述产品和数据的名称、时间、地理位置等属性以及数据格式和数据获取方式、地址等数据服务信息,是北京 GISC 系统提供数据发现和访问服务的基础。对于 WMO 全球交换数据的元数据描述中,使用固定的文件标识"urn:x-wmo:md:int.wmo.wis:T1T2A1A2iiCCCC",其中,T1T2A1A2iiCCCC 是简式报头项,其余部分为固定字符串。通过文件标识中简式报头项可以匹配到一个时间序列的某类 WMO 全球交换数据。

根据不同使用目的,北京 GISC 中元数据以 3 种形式进行存储:①将 GISC 数据应用相关的元数据抽取其中部分信息存储在关系型数据库中,为 GISC 数据应用提供便利,如缓存数据是否存在对应元数据的检查;②为支持元数据文件的访问,提供元数据的 XML 文件存储;③为支持 GISC 全文检索功能,以 BLOB(二进制大对象)数据类型将元数据存储于数据库中。

1.3　数据缓存

北京 GISC 缓存数据有两部分数据来源:一是直接通过 GTS 接收到的全球交换数据,二是通过网页数据收集收集的数据[10]。二者虽然数据来源不同,但通过统一的文件名格式保证数据缓存功能可以进行统一处理。对于 GTS 接收到的全球交换数据,为了保证用户下载缓存数据的效率,由 GTS 直接将 WMO 全球交换数据送至北京 GISC 系统指定目录结构进行缓存,当用户请求这些数据下载时,系统可直接将数据文件返回给用户,减少因通过从中国气象局本地实时数据库进行数据检索的耗时。

根据 WIS/GISC 的要求,对于进入 GISC 数据缓存中但不存在元数据文件的数据文件,系统能够进行识别和告警提示,并允许 120 s 的非法校验时间延迟[11-12]。如果 120 s 内收到数据文件的元数据文件,系统将在 GISC 数据缓存中提供该数据文件的访问服务,否则标识其为非法文件不提供服务。由于检查缓存数据合法性是通过判定系统中是否存在与之对应的元数据,需要进行数据库检索操作。

GISC 数据缓存的物理存储结构设计如图 1 所示。所有缓存数据根据处理逻辑结果的不同被存储在不同目录中,如未知格式文件(unknown)、重复文件(repeat)、非法文件(error)、过期文件(outdate)、120 s 时间延迟待校验文件(wait)等。所有通过检查的合法数据将根据文件名中简式报头 T1T2A1A2iiCCCC 项存储在指定的子目录(entry)中,子目录分为 CCCC(指 WMO 成员)和 TT(指资料类型)两级。

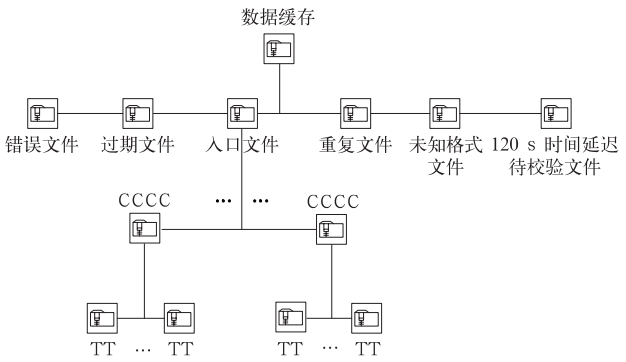

图 1　数据缓存的物理存储结构

Fig. 1　Physical storage structure of data cache

2　基于数据库检索的数据缓存处理现状

2.1　处理性能低

为了满足缓存数据检查需求,必须循环扫描缓存入口文件目录,在处理掉重复文件和非法格式文件后,提取正确文件名中简式报头信息进行元数据

匹配检索操作，如果判定数据合法则将数据拷贝到entry的指定子目录，否则进入wait目录等待120 s的时间延迟校验。随着北京GISC系统投入业务试运行，系统中以关系型数据库表形式存储的元数据记录已超过100000条，并且不断增多，而每天WMO全球交换数据文件总数超过50000个。图2记录了2012年2月29日00:00—05:59(世界时，下同)时间段内缓存数据入口目录每秒接收文件数统计趋势图，可以明显看出数据进入目录时间分布不均匀，如在00:36:20，有504个文件进入入口目录；大约在02:40—03:50，国外数值预报产品资料会非常密集地进入入口目录。由于GTS每日例行进行数据交换，特定某日某时段的数据接收情况具备普遍性，在其他时间段里，也存在时间分布不均的情况。

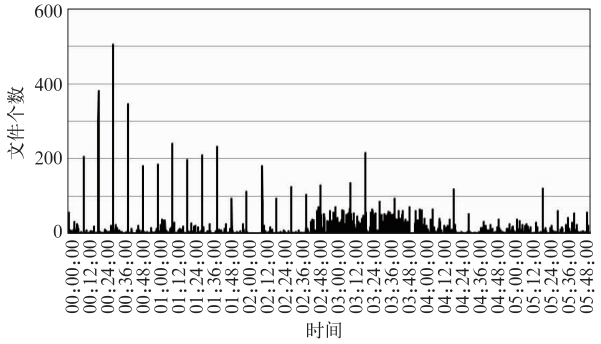

图2 2012年2月29日00:00—05:59缓存入口目录文件接收个数统计

Fig. 2 The number of receiving files from data cache entry directory from 0000 UTC to 0559 UTC on 29 Feb 2012

系统为了判断该条数据是否有效，现有的技术方案要求每条数据都需要进行数据库表查找操作，选择2012年2月29日00:25—00:35 10 min内59个文件的通过数据库进行查找的时间延迟统计，如图3所示。为提高效率，系统开启5个多线程并行处理，元数据信息表也添加索引进行优化，但处理性能仍然不理想，并且性能也可能随时因系统负载增大而降低。尽管不在数据密集到达时间段，但检索耗时平均时间大约是500 ms。通过在多个其他不同时间段的反复测试结果显示，检索耗时平均超过500 ms。在同时开启5个处理线程情况下，500份文件仅数据检索操作就需要50 s，这对于120 s时间延迟校验会导致较大误差，若遇到数据密集到达时甚至会出现1条数据等待120 s还未进行校验就已标记为非法的情况。

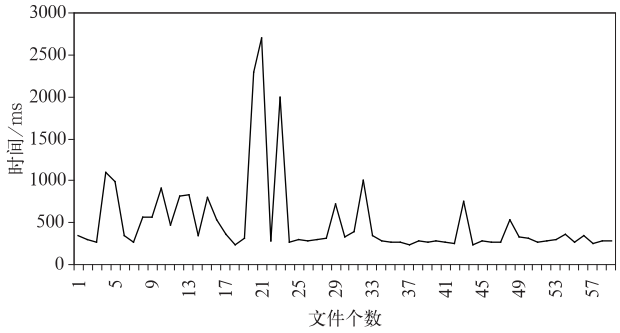

图3 2012年2月29日00:25—00:35原有技术方案中元数据文件检索耗时

Fig. 3 The metadata files retrieval consuming time in the original technical solutions from 0025 UTC to 0035 UTC on 29 Feb 2012

2.2 扩展能力弱

除数据缓存服务外，GISC系统还有其他功能需要依托缓存处理实现，例如：①数据缓存日志信息记录，为了提供缓存数据监视功能，所有数据缓存的处理结果需要实时记录；②缓存数据订阅处理，在判断缓存数据合法后检查是否有用户进行订阅，一旦是用户订阅的数据则将数据链接到订阅用户分发目录。由于原有缓存处理性能低，为了尽可能保证效率，系统不能在该功能上进行扩展以支持相关功能的实现，否则会因为额外的处理逻辑加剧缓存处理效率下降。

3 基于内存对象缓存应用的优化处理方案

3.1 问题分析

传统的建立数据库索引，启用多线程方法能一定程度提高处理效率，但无法从根本上避免频繁读写操作带来的耗时，而其中大量读写操作来自对数据库的访问，因此需要对处理方式进行优化，减少数据库访问。

近年来，内存容量不断提高，价格不断下跌，操作系统已经可以支持更大的地址空间，充分利用内存技术提升系统性能成为一个热点[13-14]。较为普遍的做法是使用内存数据库技术(也叫主存数据库技术)，对查询处理、并发控制和恢复的算法和数据结构进行重新设计，以便更有效地使用内存。相对于磁盘访问，内存数据读写速度要高出几个数量级。这种做

法实现复杂度较高,即使利用已有的中间件产品,由于不了解其中的实现算法,其运行稳定性和可靠性存在风险。常见的开源内存数据库产品包括 Memcached 和 Redis[15]。

Memcached 是一款基于事件处理的分布式内存数据库产品。Memcached 没有过多考虑数据的持久化问题,一旦重新启动操作系统会导致全部数据消失,如果使用 Memcached 来解决数据缓存的校验功能,需要通过二次开发解决处理日志信息的持久化问题。Memcached 按照固定的块大小分割内存来解决内存碎片问题,若存储长度不一的元数据信息表时无法充分利用内存。

Redis 在很多方面与 Memcached 具有相同的特征,不同在于 Redis 增加了持久化的功能,Redis 定期通过异步操作将数据库内容拷贝到硬盘。但 Redis 需要通过额外的数据裁剪功能来保证合理的内存使用空间,使得应用复杂度增加。

数据缓存处理功能仅针对独立的元数据数据表进行操作,考虑一种简单方案是使用内存对象缓存技术[16-17],将某些量小、使用次数多的数据以键(key)/值(value)对的方式保存在内存对象缓存系统中,合理增大内存缓冲,优化数据结构,改进查询访问的命中率,提升整体效率。这种方法实现复杂度小,通过自主开发相应的程序,有针对性地解决 GISC 缓存处理的业务逻辑。

3.2　技术实现

如图 4 所示,基于内存缓存技术的应用定期更新抽取元数据信息表中的信息,封装成元数据信息对象后载入内存,进行数据的合法性检查、延迟处理以及订阅等业务逻辑。

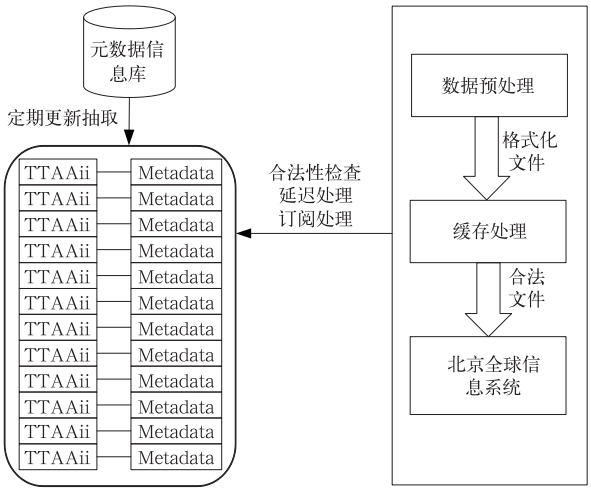

图 4　数据缓存功能处理流程图

Fig. 4　Workflow of data cache functionality

由于元数据列表不是链表或者树形结构的序列,不能直接使用任何排序方法来提高查找效率。采用哈希表存储这种离散的对象数据[18]后,可通过建立哈希索引来提高检索效率。哈希表(Hash Map)是将键通过固定的哈希函数转换成一个整型数字,然后将该数字对数组长度进行哈希散列运算,运算的结果被当作数组的下标,然后将值存储在以该数字为下标的数组空间里。当使用哈希表进行查询的时候,再次使用哈希函数将键(key)转换为对应的数组下标,并定位到该空间获取值(value)。

如图 5 所示,元数据通过简式报头哈希索引来管理:抽取元数据标识中简式报头 T1T2A1A2iiCCCC 项作为哈希表的 key 项,对应的元数据信息作为 value 项,每个对象包含元数据校验、更新必须的属性信息。

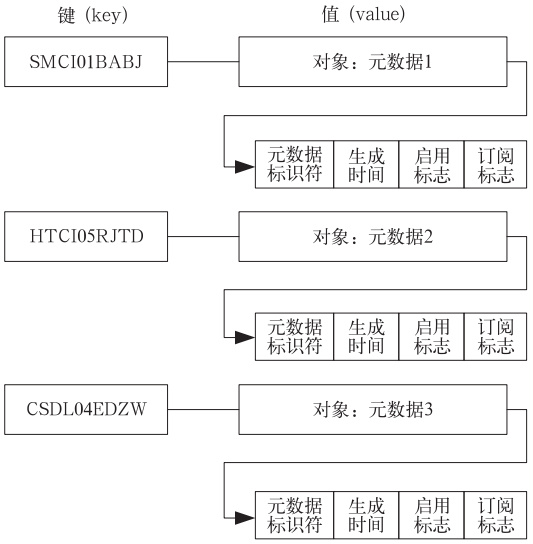

图 5　基于哈希表的内存对象组织

Fig. 5　Structure of the memory objects based on Hash Map

为了满足数据订阅、延迟处理等功能,采用多线程机制,封装其他的功能对象进行并发处理(如图 6 所示)。

①守护进程对象:监视和控制各个线程的运行状态。

②入口文件检查对象:当检测到缓存入口目录数据到达时,通过查找内存对象缓存是否有对应的报头 T1T2A1A2iiCCCC 来检查入口数据有效性,未找到匹配信息的数据放入延迟检查目录。

③等待文件检查对象:等待 120 s 非法校验延

迟目录中的数据文件的元数据文件是否存在，仍然没有则判定该数据非法。

④元数据更新对象：初始运行时元数据库中元数据列表一次性导入内存，此后内存对象缓存将定期增量更新，同步频率可以进行配置。

⑤订单更新对象：判断合法的缓存数据是否为用户订阅的数据，如果是，则将数据文件链接到用户订阅分发目录[19]。

⑥日志记录对象：为提供缓存日志信息功能，系统采取日志信息异步更新方法，当内存中保存的日志信息超过配置限额时，一次性将日志持久化写入磁盘，减少磁盘操作。

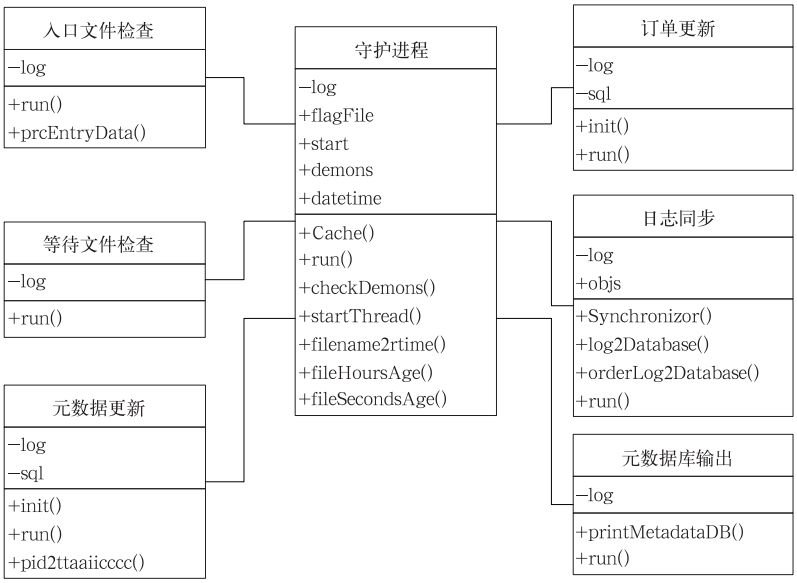

图 6 多线程缓存处理类图

Fig. 6 Class diagram of multi-threaded cache processing

4 优化效果

优化的数据缓存应用处理效率高，还可以扩展进行订阅处理操作，使得数据处理耗时平均小于 5 ms，能保证缓存数据高效率的处理需求，并提供缓存日志记录以及订阅数据处理等扩展功能。图7是通过内存对象缓存应用进行缓存数据检查的优化效果，选择 2012 年 2 月 29 日 13:00—13:40 缓存数据文件处理延迟情况。

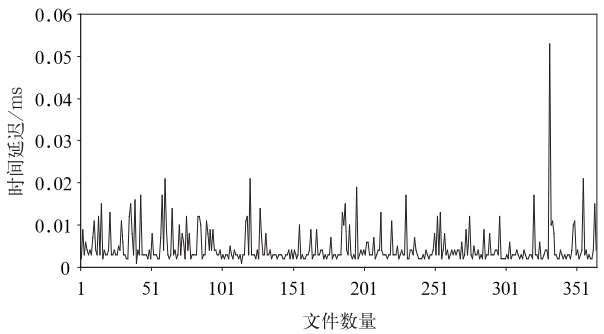

图 7 2012 年 2 月 29 日 13:00—13:40 优化后缓存数据文件处理时间延迟

Fig. 7 The data cache file processing delay in optimized technical solution from 1300 UTC to 1340 UTC on 29 Feb 2012

5 小 结

分析现有数据缓存处理的性能瓶颈主要在于频繁访问数据库的磁盘读写耗时，通过建立一个内存对象缓存应用优化传统数据库检索处理方式，将需要使用的数据一次性组织在内存中，有效减少了数据库查询访问带来的性能下降。该方案也为其他海量数据处理应用提供了一种新的解决思路，特别是针对数据具备动态变化特性以及高时效要求的场景。传统的关系型数据库在应对高时效的数据处理时受制于磁盘读写性能，而广泛采用的内存数据库中间件产品则复杂度太高，很难满足特殊的气象资料复杂的处理逻辑。因此，采用内存对象缓存应用是介于二者的一种方案，既能与关系性数据库较好衔接，又能减少关键的核心处理的磁盘读写操作，可

以有效降低数据库的负载,提高处理性能。

随着 WIS 进一步发展,GISC 缓存数据的范围可能不限于 WMO 全球交换数据,为了保证 7×24 h 业务的高效、可靠运行,提供分布式的缓存应用,需要研究多机内存缓存机制并构建一种分布式的缓存应用。

参 考 文 献

[1] 刘华,周峥嵘. WIS-WMO 未来信息系统. 气象软科学,2007,83:143-150.

[2] Geoff L. The birth of WMO Information System. *Bulletin of WMO*,2003,55(4):232-238.

[3] 李湘,王甫棣,姜立鹏,等. WIS 的实现技术研究及应用. 气象,2011,37(10):1301-1308.

[4] 祝婷,李湘. WMO 信息系统中气象元数据的设计与实现. 应用气象学报,2012,23(2):238-244.

[5] 姜立鹏,李湘. 基于 OAI-PMH 协议的 WMO 信息系统元数据同步功能设计与实现. 气象科技,2012,40(2):185-188.

[6] 曹卫. 基于 XML 的空间元数据系统的思考. 计算机技术与发展,2010,20(7):32-35.

[7] 周峥嵘,王琤,何文春. 分布式气象元数据同步系统的探索研究. 应用气象学报,2010,21(1):121-128.

[8] 高峰,王国复,喻雯,等. 气象数据文件快速下载服务系统的设计与实现. 应用气象学报,2010,21(2):243-249.

[9] 王国复,徐枫,吴增祥. 气象元数据标准与信息发布技术研究. 应用气象学报,2005,16(1):114-121.

[10] Wang Fudi, Yao Yan, Li Xiang, et al. Establishment of WMO Information System in Beijing. *Lecture Notes in Electrical Engineering*, 2012, 100: 520-527.

[11] Robert H, David T, Eliot C. WMO Information System Functional Architecture. [2012-07-30]. http://www.wmo.int/pages/prog/www/TEM/ET-WISC-III/documents/WIS-FuncArch_current.doc.

[12] Thomas D, Christian E, Husband R. WMO Information System Compliance Specifications of GISC, DCPC, and NC. [2012-07-30]. http://www.wmo.int/pages/prog/www/WIS/documents/TechnicalSpecification1-2.doc.

[13] 刘云生,李国徽. 实时内存数据库的装入. 软件学报,2000,26(4):829-835.

[14] 杨润芝,马强,李德泉,等. 内存转发模型在 CIMISS 数据收发系统中的应用. 应用气象学报,2012,23(3):377-384.

[15] 杨艳,李炜,王纯. 内存数据库在高速缓存方面的应用. 现代电信科技,2011(12):59-64.

[16] 赵玉伟,赵小雨,乔木. 缓存技术在 B/S 架构信息系统中的应用. 计算机工程,2008,34(1):233-235.

[17] 张震波,杨鹤标,马振华. 基于 LRU 算法的 Web 系统缓存机制. 计算机工程,2006,32(19):68-70.

[18] 杨燕明,鲁志军,陈煜,等. 一种基于哈希索引的内存表模型. 计算机应用与软件,2012,29(1):215-216.

[19] 王甫棣,姚燕,李湘. 基于 XML 的气象数据订阅系统设计. 气象科技,2012,40(4):591-595.

Optimization of Data Cache Function in Beijing Global Information System Center

Wang Fudi Jiang Lipeng Yao Yan

(*National Meteorological Information Center*, *Beijing* 100081)

Abstract

WMO Information System (WIS) is a coordinated, distributed, global infrastructure for the collection and sharing of information for all WMO and related international programs. As the core center of WIS, each Global Information System Center (GISC) is responsible for the collection and distribution of global exchanging data, and providing data discovery and access service. As a GISC of WIS, a scalable and flexible system is designed and established to satisfy WIS/GISC functionalities in Beijing. Beijing Global Information System Center should hold at least 24-hour WMO global exchanging data files, which could be accessed by authorized users through DAR (data discovery, access and retrieval) services.

GISC Beijing has to do validation check for all the global exchanging data files, and only files matching the corresponding metadata could be brought into data cache. The existing approach for the validation is based on database retrieval operation. Currently, there are more than 100000 metadata records stored in the relational database GISC Beijing, while the system receives more than 50000 global exchanging data files in an uneven distribution of collection time. The disadvantage of this way is that frequent database I/O operation would lead to a sharp decline of the system performance, especially when a large number of data entering. Therefore, the approach could not satisfy the requirement of real-time data cache service. Although establishing the table index and multi-threaded mechanism could solve the efficiency of data processing to some extent, it is inevitable that frequent database I/O operation would bring about the performance bottleneck. Therefore, the operation treatment should be optimized.

It is a possible way making full use of memory technology to reduce disk I/O cache data and improve the efficiency significantly. Considering the complexity of the mature memory database, a more targeted approach is adopted which is suitable for the scenario of dynamic nature of data with the timeliness requirements. An application is designed and implemented based on the memory object caching technology. When initializing the application, the system loads metadata into memory as a hash table from the database based on stored key/value pairs, organized by the unique bulletin head information of global exchanging data. In this way, the metadata contents are encapsulated as memory objects, thereby providing fast data memory retrieval method. In addition, parallel processing is implemented to extend the functionalities, including data cache logging function and data subscription services.

As a result, effects of the optimized function can satisfy the real-time business requirements, reducing the data processing time to an average of less than 5 ms. It also provides an easier way to do extensions by adding memory object using parallel processing.

Key words: data cache; memory object; GISC Beijing