一种对象化并行计算框架①

唐云善[1], 缪巍巍[2]

[1](国网电力科学研究院 信息通信技术分公司, 南京 210006)
[2](国网江苏省电力公司 信息通信分公司, 南京 210006)

摘 要: 分布式计算、并行计算、内存计算是目前提高计算性能的关键技术和热点研究领域. 在大数据环境下, 针对数据型统计分析系统性能劣化明显、不能满足用户使用需求的问题, 提出了一种轻量级高性能对象化并行计算架构, 研制了该架构的对象服务组件、对象管理服务组件和客户端代理组件, 并将该架构和组件在国家电网资产质量监督管理系统中进行了验证应用, 其效果表明该框架能大幅提升大数据处理效率.

关键词: 并行计算框架; 分布式计算; 内存计算

Objectification Parallel Computing Framework

TANG Yun-Shan[1], MIAO Wei-Wei[2]

[1](State Grid Electric Power Research Institute, Nanjing 210006, China)
[2](State Grid JIANGSU Electric Power Company, Nanjing 210006, China)

Abstract: At present, the distributed computing, parallel computing and memory computing are key technologies and research spots of improving performance. In Big Data scene, the computing performance of data statistics and analysis system degrades so noticeably that it cannot satisfy user demand. To deal with the problem, this paper provides the Objectification Parallel computing Architecture. Based on the architecture, we develop the system which includes the Object Server Component, Object Manager Component and Client Proxy component. The system is applied to the Electric Asset Quality Supervision Manage System (EAQSMS) of State Grid of China. The result shows that the statistics performance of system index is improved dramatically

Key words: parallel computing framework; distributed computing; memory computing

近年来, 大数据[1-3]引起了产业界、科技界和政府部门的高度关注, 大数据分析处理所需的计算机技术也在快速发展[4]. 当前, 随着企业信息系统应用的逐步深入, 业务数据的体量逐渐增大, 对大数据的处理效率要求越来越高, 现有的软件工具在提取、存储、搜索、分析和处理海量、复杂的数据已显得无能为力. 特别是, 现有系统的数据大都基于集中式磁盘关系数据库进行存储, 大量频繁的磁盘 I/O 操作使得系统运行效率和可靠性逐渐降低, 给用户的使用和体验带来不便, 更为严重的性能问题还致使系统不可用. 因此, 研究如何提高数据的分析处理效率是目前大数据领域的一个重要技术课题.

计算框架技术是提高数据分析处理效率的关键技术之一. 目前, 业界出现了一些数据处理框架, 如 Storm 一种实时数据流处理框架, MapReduce[5]一种大数据分布式处理框架, 以及内存计算平台等. 本文在研究分布式对象服务[6]、内存计算和分布式计算等技术基础上提出了一种轻量级高性能对象化并行计算(Objectification Parallel Computing, 以下简称 OPC)框架.

1 相关工作

1.1 MapReduce 计算框架

目前, MapReduce 分布式计算框架是比较成熟、可

① 收稿时间:2014-11-07;收到修改稿时间:2015-03-18

用于大数据分析处理的主流计算框架，最早由 google 公司提出并实现，用于解决互联网站系统的大数据分析处理. Hadoop[7,8]是 google 公司分布式存储、分布式计算体系的开源实现，Hadoop 实现了 MapReduce[9]大数据处理框架. 在 Hadoop 的 MapReduce 框架下，用户的任务被分成 Map 环节和 Reduce 环节，通过 Map 环节将任务拆分成子任务，通过 Reduce 环节执行子任务，最后汇总得到结果. 详细过程见图 1.

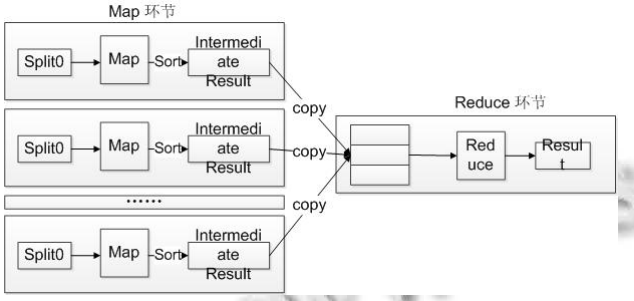

图 1　MapReduce 框架

特别的，Map 环节可根据输入数据的分块来进行任务拆分，任务执行环节可根据输入数据所在机器位置来执行任务. 为提高大数据处理效率，MapReduce 天然就是与分布式文件系统和分布式数据库进行集成的处理框架. MapReduce 也可与集中式关系数据库进行集成，但是难以与内存计算平台进行集成，大量磁盘 I/O 性能瓶颈问题无法解决，效率提高不明显，难以满足实时应用需求.

1.2 内存计算与并行存取

在大数据内存计算方面，目前出现的一些内存计算平台(如 SAP 公司的 HANA[10])通过内存计算、一体机技术、共用内存数据模型等技术，实现了高效的数据查询统计和分析,极大提高了应用数据存取效率，避免了数据处理过程中频繁的磁盘 I/O 操作. 但是，内存计算平台只是解决了数据高效存取问题，对于如何提高数据处理效率还需进一步研究或考虑与其他分布式计算框架集成.

为解决数据库的磁盘 I/O 瓶颈问题，目前也出现了数据服务与数据存储分离、数据并行存取技术(如 Oracle 的 Exadata 数据库一体机). 通过高性能硬件、高速网络接口，采用智能扫描、智能存储、智能索引、混合列压缩等技术，提高数据库系统在多并发应场景下的数据存取效率，其核心思想是将数据存储与数据服务分离，数据服务器将数据存取需求进行分解后，分配到多台存储服务器并行工作，各存储服务器将结果返回到数据服务器，存储服务器上的数据存取通过磁盘 I/O 来进行. 此技术能够极大缓解目前集中式磁盘关系数据库的数据存取效率问题，但性能提升的效果任然受到磁盘 I/O 限制，难以满足实时性要求较高的应用需求.

2 对象化并行计算原理

2.1 内存计算与并行存取

依据数据与业务逻辑分离的思想，本文提出数据对象和任务对象. 数据对象是对磁盘关系数据库中数据的封装，形成具备属性和操作的内存对象，通过数据同步机制将磁盘关系数据库中对应数据单向同步缓存到对象属性中. 任务对象是对用户提交任务请求的抽象和封装，形成具备属性和操作的内存对象，属性由任务参数和执行结果组成，操作则执行任务的具体工作. 任务对象使用数据对象缓存的数据.

将内存计算封装在于内存对象，将任务执行演变成对象接口调用，将数据和计算任务通过分布式计算[11]紧密集成，提出一种对象化并行计算框架 OPC.

2.2 对象化并行计算框架结构

2.2.1 结构组成

OPC 框架由对象服务器、对象管理服务器和客户端代理组成，体系结构见图 2.

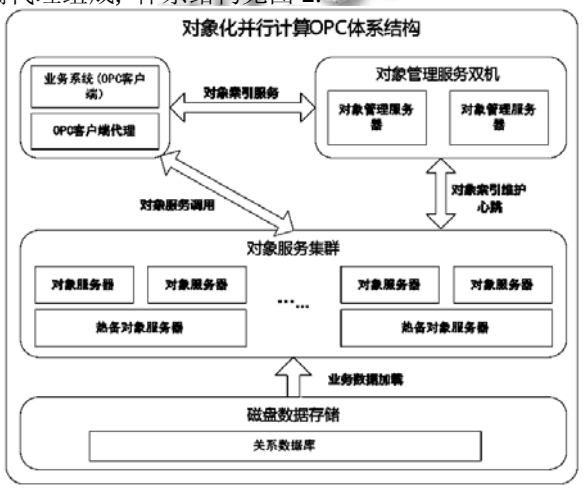

图 2　OPC 体系结构

(1)对象服务器

对象服务器是 OPC 框架的核心组件，提供任务对象服务远程接口；负责创建和维护数据对象，加载缓

存数据并维护磁盘关系数据库数据与数据对象的一致性; 屏蔽底层数据存储方式的差异性, 负责内存中任务对象和数据对象的持久化等. 对象服务器组件由对象池、对象管理、数据同步、数据访问、服务接口、并发控制等功能组成.

(2)对象管理服务器

对象管理服务器是 OPC 的管理中心, 接受对象服务器的对象注册信息, 负责对象服务检索、分配和管理, 具备对象索引管理和对象服务故障恢复管理等功能.

(3)客户端代理

客户端代理是给业务应用系统提供对象服务接口调用的本地代理组件, 主要包括会话管理、对象服务代理和对象管理服务代理.

2.2.2 运行机理

可动态横向扩展的对象服务集群将磁盘关系数据库数据以数据对象形式缓存到内存, 并接受对象管理服务器的统一调度和管理. 客户端代理通过对象管理服务器寻址得到对象服务器, 然后向对象服务器发送远程任务对象调用, 对象服务器接受调用后, 利用缓存的数据对象进行计算, 最后将各对象服务器的计算结果汇总返回给客户端. 各组件之间的详细协作关系见 2.4 节.

2.3 对象化计算解决方案

在应用 OPC 进行计算分析时, 需经对象建模、对象分布式缓存、对象索引、计算任务拆分及分布式计算等步骤.

2.3.1 对象建模

OPC 按照业务逻辑关系创建两种对象模型, 一种是缓存数据库数据的数据对象模型, 另一种是执行任务工作的任务对象模型.

数据对象模型对象化后缓存参与分析计算的关系数据库中的业务数据, 一个数据对象模型对应关系数据库中一张表或者多张表. 数据对象模型必须继承 OPC 框架的 PersistantableObject 基类, 并实现用于加载数据的 load 虚函数, 数据对象模型的属性由业务数据字段构成.

任务对象模型是对任务工作的封装, 按逻辑关系, 任务对象分为单对象和对象集, 单对象之间耦合度低, 对象之间没有逻辑关系, 可以完成某一类任务; 对象集之间耦合度高, 对象之间有逻辑关系, 多个对象一起完成某一类任务. 任务对象模型属性由任务条件和任务结果构成, 方法包括任务执行的函数 syncStatistic, syncStatistic 对任务进行拆分、执行和结果汇总.

2.3.2 分布式对象缓存

OPC 分布式对象缓存包括数据对象缓存和任务对象缓存, 缓存机制见图 3.

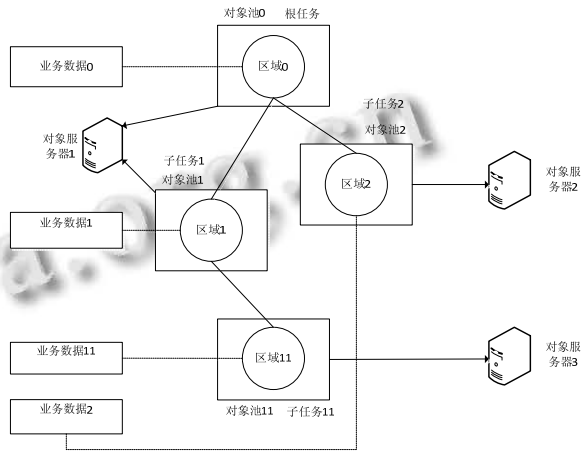

图 3 OPC 数据缓存机制

(1)数据对象缓存. 对象服务器将数据源中参与计算的数据以数据对象缓存到集群中多个对象服务器的数据对象池中. OPC 目前支持树形结构的数据, 即认为业务数据具有“所属区域”属性, 而具体区域之间形成一颗或多颗树, OPC 一个数据对象池缓存一个树节点对应的数据. 如: 图 3 中, 业务数据 0 的“所属区域”属性值为“区域 0”, 业务数据 1 的“所属区域”属性值为“区域 1”, 业务数据 11 的“所属区域”属性值为“区域 11”, 业务数据 2 的“所属区域”属性值为“区域 2”. “区域 0”、“区域 1”、“区域 11”、“区域 2”根据隶属关系形成一颗树. OPC 按照“区域 X”定义对象池 0、对象池 1、对象池 11、对象池 2, 并以对象池为单位将数据缓存在对象服务器上. 对象池和对象服务器之间可以是 1 对 1 或多对 1 的关系.

(2)任务对象缓存. 任务对象缓存遵循任务与其对应的对象池位置相同原则. 图 3 中, 根任务对象对应对象池 0, 缓存在对象服务器 1 上; 将根任务拆分出子任务 1 和子任务 2, 子任务 1 对应对象池 1, 子任务 2 对应对象池 2, 子任务 1 和子任务 2 分别缓存在对象服务器 1 和对象服务器 2 上; 将子任务 1 拆分出子任务 11, 对应对象池 11, 子任务 11 缓存在对象服务器 3 上.

2.3.2 对象索引创建

对象服务器缓存对象后，将服务器 IP 地址、对象池信息、对象总数、对象名称、占用空间、缓存所用时间等以心跳方式发送到对象管理服务器，对象管理服务器收集信息后根据逻辑关系创建并维护对象索引表. 对象管理服务器通过对象服务接口对外提供对象索引服务，通过该服务可根据用户任务定位对象服务器和任务对象.

2.3.3 任务拆分

执行一次任务可能需要使用多个对象池的数据，因此需要对任务按照对象池进行拆分. 接受任务处理的入口对象服务器首先根据任务参数条件从对象池中选择一个根任务对象与本次任务对应，根任务根据完成此任务所需本级和直接下级对象池信息对根任务进行拆分，得出下级子任务，并将子任务与任务对象进行对应，子任务还可根据是否需要下级对象池继续进行拆分，直到拆分出的子任务所需对象池就在本子任务对象所在对象服务器为止. 这样，就形成了一颗以根任务为根、各级子任务为节点的任务对象树.

2.3.4 任务执行

OPC 由客户端代理、对象服务器、对象管理服务器各组件相互协作完成工作任务，协作关系见图 4，OPC 任务执行步骤如下.

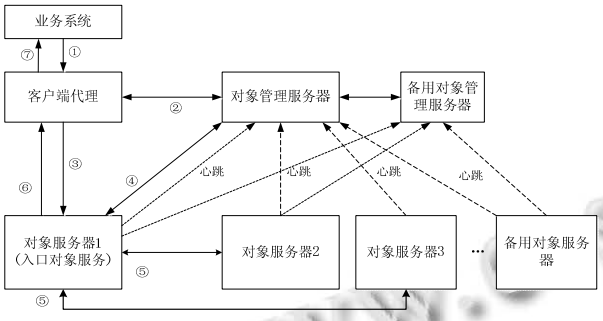

图 4　OPC 组件协作关系

(1) 业务系统将分析处理相关工作组装成任务，将任务及执行任务的条件和参数传递给 OPC 客户端代理组件，然后等待(异步或同步)OPC 客户端代理返回结果.

(2) 客户端代理组件根据任务条件和参数向对象管理服务器询问可完成此任务的对象服务器及任务对象，对象管理服务器依据缓存的对象索引，通过比较、计算，将可完成此任务的入口对象服务器(比如图 7 中对象服务器 1)地址及任务对象返回给客户端代理.

(3) 客户端代理向入口对象服务器发起对象接口调用，入口对象服务器上的根任务对象接受调用并分析此任务所需数据对象.

(4) 如果入口对象服务器本机数据对象不足以支撑此任务，则向对象管理服务器发起其他对象服务器的寻址请求，对象管理服务器经过比较、计算后，返回其他对象服务器地址及服务对象.

(5) 入口对象服务器同时向本机对象服务和其他对象服务器进行任务(子任务)调用，对象服务器执行任务(子任务). 特别的，子任务在执行过程中还可自动拆分执行.

(6) 各子任务对象将执行结果返回给直接父任务对象，入口对象服务器的根任务对象最终汇总得到任务执行结果并返回给客户端代理.

(7) 客户端代理将任务执行结果返回给业务系统.

3　验证与分析

OPC 解决方案在电网资产质量监督管理系统上进行了验证和应用. 不改变现有电网资产质量监督管理系统的部署和应用，增加 OPC 对象服务器、对象管理服务器和 OPC 适配器(OPC 适配器是将电网资产监督管理系统与 OPC 集成的纽带). 电网资产质量监督管理系统将电网可靠性指标统计分析任务写入数据库，适配器读取统计分析任务并解析，将统计条件和统计数据范围发送给 OPC 客户端代理组件，客户端代理组件与 OPC 集群进行交互执行统计分析任务，适配器将统计分析结果写回资产质量监督管理系统数据库统计分析界面中展现统计分析结果，系统集成方案见图 5.

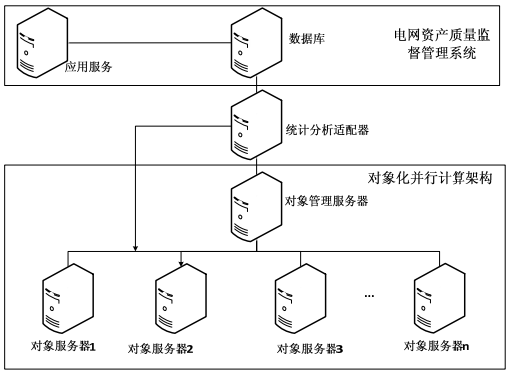

图 5　OPC 应用集成与测试

采用普通 4 台 PC 机组成集群作为 OPC 应用的硬件环境. 以中压供电可靠性最近 1 年的基础数据和运行

数据作为指标计算的数据范围, 并分别以 100 万、500 万、1000 万、2000 万条记录数据进行了统计分析的测试, 对 1 亿、10 亿、100 亿条记录数据进行了推演.

3.1 需求描述

电网可靠性指标统计是电网资产监督管理系统最为重要的功能, 各类指标计算都需要对电网基础数据、运行数据进行逻辑计算, 涉及数据量达到数十亿记录数据. 下面以架空线路停电平均持续时间 ICATI 指标计算为例进行说明. ICATI 计算公式如下:

$$ICATI = (G8 + Y8) / L341 \quad (1)$$

其中 G8 为故障停电时户数, Y8 为预安排停电时户数, L341 为停电用户总数, 子指标计算公式如下:

$$G8 = sum(BL \times CXSL) \quad (2)$$

其中 BL 为责任部门分摊比例, CXSJ(持续时间)=停电终止时间(ZZSJ)-停电起始时间(QSSJ), sum 是对某个单位下所有满足条件的停电时间数据求和, 求和条件是“停电责任部门 like 51%".

$$Y8 = sum(BL \times CXSL) \quad (3)$$

其中 BL 为责任部门分摊比例, 求和条件是“停电责任部门 like "50%".

$$L341 = count(*) \quad (4)$$

在计算 L341 指标时,count 表示统计时段内的所有记录数据, 需要合并相同停电用户的记录数.

传统基于磁盘关系数据的统计分析方法由于需要在数据读取时进行大量磁盘 I/O 的操作, 使得指标统计性能较低, 本文依托该项目, 将中压供电可靠性指标统计功能采用OPC实现, 验证了OPC的效率和可靠性.

3.2 验证过程

1) 对象建模

遵循 OPC 框架接口规范, 根据电网可靠性指标统计业务统计需求, 对中压供电可靠性指标进行分类, 抽象成统计类, 统计类实现 OPC 框架定义的统计算法接口, 定义统计任务模型类同时, 对中压供电可靠性指标统计分析所需数据需进行数据对象模型设计.

本文验证实例 ICATI 指标计算用到的数据对象类有: gd_yx_zytdyh、gd_yx_zytdyh_ky. 任务对象类有: StatisticTDYH、StatisticZHFX.

2) 分布式对象数据缓存

根据集群中计算机内存配置情况, 为对象存放分配 JVM 空间, 把 6 种对象按区域划分成数据块, 缓存到不同计算机内存中. 其中每个地区内的 4 种对象共同完成指标计算, 缓存在同一个对象池中.

3) 对象索引创建

对象服务器缓存对象后, 对象管理服务器创建对象索引, 数据对象分为三级, 1 级代表国网公司, 2 级代表分部(共有 5 个分部分别是 0130 华东分部)、0140 华中分部、0160 西北分部、0120 东北分部、0110 华北分部), 3 级代表省公司(如 013032 代表江苏省电力公司). 索引表存储了不同对象池所在的服务器地址, 如: 01 代表国网总部数据对象.

4) 任务对象执行

任务按照国网总部、国网分部、省公司三级进行划分, 省公司子任务完成统计任务后将结果提交给分部子任务, 分部子任务完成统计任务后将结果返回给国网根任务, 国网根任务汇总后将结果返回给客户端代理. 如, 指标 ICATI 计算任务, 首先根据对象管理服务器的索引表, 对根任务 ICATI 进行拆分, 见图 6.

图 6 ICATI 任务拆分

任务完成拆分后, 在 OPC 框架下执行, 具体执行过程将按照 2.4 节描述的过程执行, 不再赘述.

3.3 验证结果

1) 功能测试结果

中压供电可靠性统计 127 个指标分为 5 类统计, 采用 OPC 架构的统计指标结果与原系统(采用存储过程进行统计)的结果进行对比, 正确率为 100%.

2) 扩展性测试结果

在数据量逐渐增大, 通过增加集群中对象服务器数量, 测试集群扩展性能, 结果见图 7.

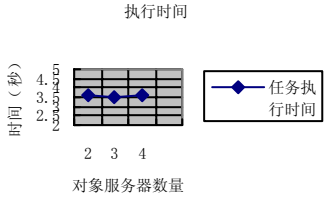

图 7 扩展性测试

纵坐标表示执行时间, 横坐标表示集群计算机数

量, 集群计算机数量从 2、3 扩展到 4, 缓存的数据从 500 万、1000 万到 2000 万, 任务执行时间在基本保持在 3.5-3.7 秒左右. 这表明, 随着数据量增加, 只要扩展对象服务器数量, 除去少量网络开销后, 任务执行时间不会明显增长.

3)性能测试结果

通过逐步加大数据量进行性能测试, 并与已有系统统计进行对比, 结果见表 1.

表 1 性能对比

序号	数量量 (万条记录)	耗时(s)	
		OPC	传统存储过程
1	100	2.1	2
2	500	3.1	4
3	1000	3.5	120
4	2000	3.8	270

从表 1 可以看出, 在数据量较小的情况下, OPC 和传统存储过程的统计耗时相当. 随着数据量加大, 传统存储过程统计耗时增加比较明显; 通过扩展 OPC 对象服务器数量, 其性能提升 70 倍左右.

3.4 分析

电网资产质量监督管理系统是国家电网公司一级部署系统, 系统数据涵盖整个国家电网公司, 数据量大, 高中低压全部数据约有 60-100 亿条, 如果全部缓存到内存中需要 6000-10000GB 的内存容量. 因测试集群有限, 无法完全加载, 故选取部分数据进行了加载并测试.

通过实验和现场部署测试, 可以看出, OPC 系统正确率为 100%, 同时具备故障自动恢复能力, 具有良好的可用性; 随着数据量的增加通过扩展对象服务器数量, OPC 计算效率是传统计算方法的数十倍到数百倍.

4 结语

在大数据环境下, 针对数据型统计分析系统性能随数据量增大而劣化、不能满足用户使用需求的问题, 本文提出了一种轻量级高性能对象化并行计算框架, 研制了该框架的对象服务组件、对象管理服务组件和客户端代理组件. OPC 具有良好的架构设计, 系统自身消耗资源少, 是一种轻量级计算框架; 通过数据库和对象的同步, 实现内存对象更新, 系统移植时, 原有的数据存储不需做改动; 基于对象的任务拆分和调度, 能够利用对象提供的封装和复用机制, 以不同粒度的对象优化任务拆解和分配, 提高任务执行效率. OPC 在国家电网资产质量监督管理系统项目进行了应用验证, 其效果表明, 可大幅提升系统指标统计效率.

OPC 目前已提供了一种大数据的高效解决方案, 但在内存优化管理、数据高效写入及事务管理、性能提升理论公式研究等方面需进一步深入研究.

参考文献

1 李国杰,程学旗.大数据研究:未来科技及经济社会发展的重大战略领域—大数据的研究现状与科学思考.中国科学院院刊,2012: 647–657.

2 邬贺铨.大数据时代的机遇与挑战.求是,2013:47–49.

3 王元卓,靳小龙,程学旗.网络大数据:现状与展望.计算机学报,2013:1125–1138.

4 覃雄派,王会举,杜小勇,王珊.大数据分析—RDBMS 与 MapReduce 的竞争与共生.软件学报,2011,23(1):32–45.

5 Dean J, Ghemawat S. MapReduce: Simplified data proeessing on large clusters. Communications of the ACM, 2008, 51(l): 107–113.

6 Ma YT, He KQ, Li B, Liu J, Zhou XY. A hybrid set of complexity metrics for large-scale object-oriented software systems. Journal of Computer Science & Technology, 2010, 25(6): 1184–1201.

7 White T. Hadoop the definitive guide. Beijing: Tsinghua University Press, 2010.

8 Zhao J, Zhang RS, Zhao ZL, Chen DW, Hou LJ. Hadoop mapreduce framework to implement molecular docking of large-scale virtual screening. Services Computing Conference (APSCC). 2012. 350–353.

9 马辉.基于 MapReduce 的分布式地震射线追踪方法研究[博士学位论文].北京:中国地质大学,2012.

10 Lee J, Kwon YS, Farber F, Muehle M, Lee C, Bensberg C, Lee JY, Lee AH, Lehner W. SAP HANA distributed in-memory database system: Transaction, session, and metadata management. IEEE 29th International Conference on Data Engineering (ICDE). 2013. 1165–1173.

11 李国东,张德富.基于分布式对象的并行计算框架.软件学报,2002,13(3):342–353.