

基于非关系数据库的分布式负载均衡技术^①

李永魁, 谢文阁

(辽宁工业大学 电信学院, 锦州 121001)

摘 要: 在非关系数据库下的动态负载均衡算法中引入查询成本指标, 通过分析历史查询日志来估计查询代价, 并应用到负载均衡中的负载决策部分, 使得负载排序更准确. 同时对当前任务的查询代价进行估计, 以找到合适的接受此任务的节点, 从而使负载均衡变被动为主动, 有效地缩短平均查询响应时间, 提升系统的整体性能.

关键词: 非关系型数据库; 负载均衡; 异步计算

Distributed Load Balances Technology Based on the Nosql Database

LI Yong-Kui, XIE Wen-Ge

(Liaoning University of Technology, Jinzhou 121001, China)

Abstract: This paper is based on the non-relational database of load balancing algorithm by introducing the query cost index. The load is more accurate through the analysis of the historical inquires the log to estimate inquires the price, and applied to the decision-making part of the load in the load balancing. At the same time figure out query cost of the current task to find the right to accept the task node, thus the load balancing change from passive to active, effectively shortening the average query response time, and improve the overall performance of the system.

Key words: NOSQL; load balancing; asynchronous computation

1 相关介绍

1.1 负载均衡

负载均衡问题是经典的组合优化问题. 近年来关于负载均衡的相关算法层出不穷, 且越来越朝着智能化方向发展^[1]. 比如将负载均衡算法与粗糙集算法、蚁群算法等结合, 可参见^[2,3]. 研究者往往通过更多已知负载信息和未知预测信息作为负载均衡的评判标准^[3]. 集群负载均衡算法主要分为静态和动态两种算法. 静态负载均衡算法不考虑服务节点的实际负载情况, 而动态负载均衡算法则要考虑服务节点的当前实际负载^[4].

1.2 非关系型数据库

非关系数据库是在关系数据库基础上的一个补充. 本文选择 Redis 数据库存储聚合结果及缓存结果. Redis 本质上是一个 key-value 类型的内存数据库, 很像 Memcached. Redis 的性能非常出色, 每秒可以处理超过 10 万次读写操作, 是目前最快的 key-value DB. Redis 最大的魅力是支持保存 List 链表和 Set 集合的数

据结构, 而且还支持对 List 进行各种操作, 此外单个 value 最大的限度是 1GB, 而 Memcached 只能保存 1MB 的数据^[5].

1.3 分布式中的异步计算

异步计算是分布式计算实现的一种方式. Gearman 是一个开源产品, 其初衷是用来实现远程函数调用. 它提供的 API 可以方便地将计算转移到其他服务器上, 能够很好的实现分布式异步计算. 本文利用 Gearman 构建分布式的集群(cluster)不但能够很好的实现任务的分发, 而且 Gearman 构架设计本身就是异步计算的构架, 拥有自动任务的重发等一系列的安全机制^[6].

2 基于非关系数据库的OALP分布式查询技术

2.1 系统整体模型

该模型(如图 1 所示)利用内存数据库 Redis 读写速度快和 Gearman 擅长任务分发管理等优点, 根据 95%

① 收稿时间:2012-09-27;收到修改稿时间:2012-11-04

的 OLAP 查询中聚集函数的可分布计算的特点^[7], 将客户端提交的批量请求均匀地分配到各节点上并独立完成将客户端提交的批量请求均匀地分配到各节点上并独立完成查询, 最终将结果返回给中心服务器, 生成最终的查询结果, 从而提高了系统的综合处理性能。

分发, 通过 Gearman 可以把任务分发给不同的 Worker 去做计算. 计算, 在一个 Worker 内进行一定数量的小范围维度聚合计算. 结果合并存储, 将各个 Worker 的计算结果合并然后存储到中心 Redis. 中心 Redis 存储的才是最终的聚合结果. 如图 2 所示。

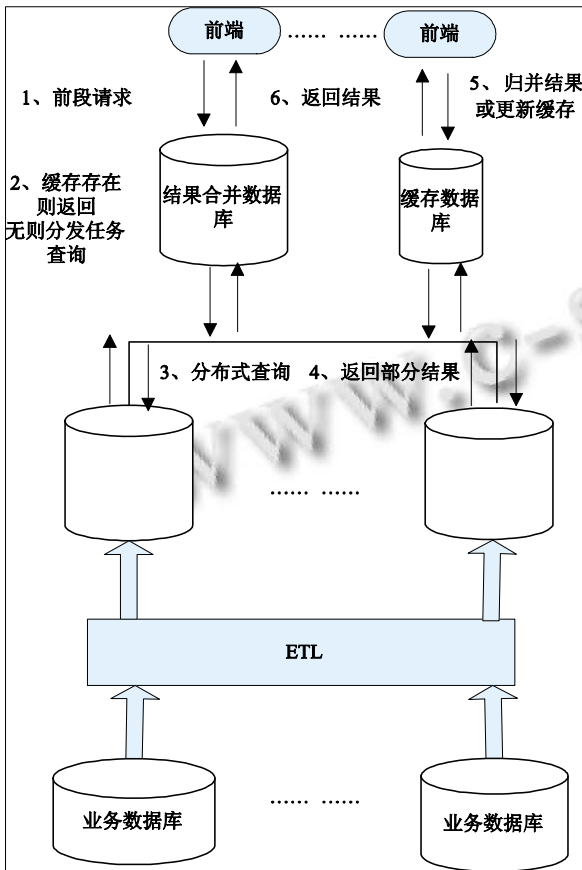


图 1 分布式查询系统构架图

2.2 具体流程

当客户端节点并行地提交查询处理请求时, 首先中心服务器会将一系列的查询请求均衡的分配到各个负载局部 KPI 计算的节点上, 该节点会迅速计算并把结果返回给中心服务器同时保存到本地的内存数据库 Redis 中以便下次使用. 中心服务器会把汇集来的结果整合后返回给客户端。

整个过程主要完成数据收集, 工作分发, 计算, 结果合并存储四部分功能. 数据收集, 主要负责收集数据源通过 E T L 传过来的数据并发送给负责计算工作的模块. 数据的分配采用事实表水平分片, 维度表全复制的方式. 并且把 ETL 改造成一个一直运行轮询新数据的 Worker, 保证各节点中的数据实时更新; 工作

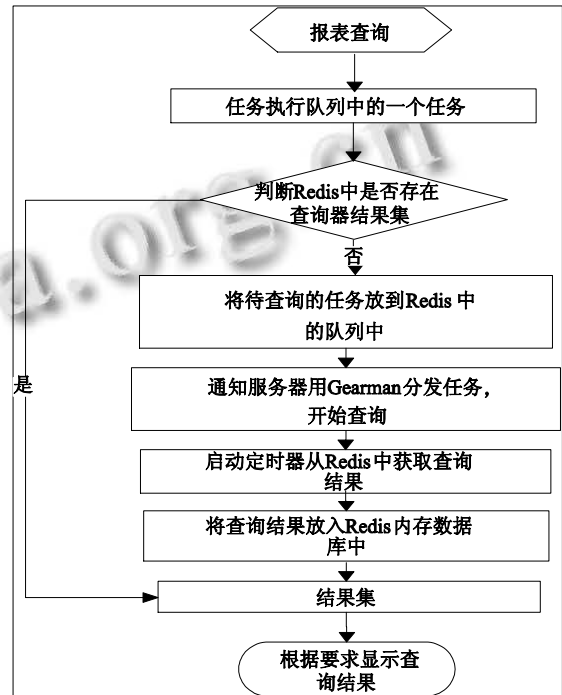


图 2 分布式计算流程图

3 负载均衡策略的优化

Gearman 本身擅长任务的分发但均衡性不够. 动态负载均衡算法能够适应异构和环境的变化动态的调节权重. 负载均衡算法大多数是选择几个计算机性能指标, 比如 CPU 占用率, 内存占用率等作为负载因子用于负载决策. 数据库集群中主要任务是完成查询统计分析工作, 对于每个查询可以实现预测估计. 常用的数据库厂商在数据库查询优化时采用根据查询代价对查询进行优化. 比如 ORACLE 的 CBO(cost-based sql optimizer 基于成本的查询器)所以借鉴这种方法引入查询成本指标, 对负载均衡算法进行了改进, 更好地找到适合接受此任务的节点. 集群中的各个数据库分在集群内的不同节点上, 为了要及时、准确掌握节点上数据库的负载情况, 在中心节点上建立负载信息表, 有中心节点管理维护。

定义 1 超载阈值: 节点 i 的最大数据处理能力, 记为 $Ability_{Data}(i)$.

定义 2 节点处理能力: 节点 i 的处理能力 $Ability(i)$ 由节点的数据处理能力 $Ability_{Data}(i)$ 和节点的网络传输速度 $Ability_{Net}(i)$ 两部分组成, 即节点的数据处理能力

$$Ability(i) = Ability_{Data}(i) + Ability_{Net}(i)$$

定义 3 平均负载率:

$$L_{avg} = \frac{\sum_{i=1}^m L_i * a_i}{\sum_{i=1}^m a_i}$$

定义 4 m 个查询的平均查询代价:

$$C_{avg} = \sum_{i=1}^m D_i * Q_i$$

对于单个节点的查询代价, 不用考虑通信的代价. 设 q_1, q_2, \dots, q_m , 为某节点的 m 个查询, D_i 是 $C_{avg} = \sum_{i=1}^m D_i * Q_i$ 需要磁盘处理的时间, Q_i 是 q_i 发生的概率. 从工作负载来看, 该节点上 m 个查询的平均查询代价:

通过 sql 语句命令得到当前查询代价, 并对查询代价进行归一处理, 预估出当前任务的查询代价 C_i .

负载均衡算法流程如下:

(1) 选取 CPU 使用率、内存使用率、磁盘 IO 延迟、查询代价 C_{avg} 等形成决策矩阵 $\{a_{ij}\}$. 按照“基于多指标决策的负载均衡方法”将决策指标矩阵转化成负载轻重决策排序 $\{L_1, L_2, \dots, L_n\}$, 其中小于 L_{avg} 的一组序列为: $\{L_1, L_2, \dots, L_i\}$.

(2) 从序列 $\{L_1, L_2, \dots, L_i\}$ 找到最适合接受此任务的节点记为 J .

J 节点满足:

$$\text{Min} \left\{ \sqrt{\{L_{avg} - (L_1 + C_1)\}^2}, \sqrt{\{L_{avg} - (L_2 + C_2)\}^2}, \dots, \sqrt{\{L_{avg} - (L_i + C_i)\}^2} \right\}$$

(3) 若试分配后的节点处理能力超出了阈值 $Ability_{Data}(i)$ 则没有可以接受此任务的节点, 排队等候其他任务执行, 再次循环查找.

(4) 否则分配任务成功更改中心节点的全局负载信息表; 确定出其中的标题项;

4 实验与性能分析

实验环境为四台虚拟服务器: 需要软件 Redis;

Gearman; Mysql; 操作系统均为 Centos6.0. 联机客户端基础环境要求: Microsoft Internet Explorer 7.0 以上或者火狐浏览器. 通过仿真实验, 在分布式系统中分别使用两种算法进行对比, 通过火狐浏览器记录响应时间, 统计后得图 3: 改进的负载均衡算法可以提高一定的执行效率. 实验同时表明增加节点为 20 个为宜, 效果最好.

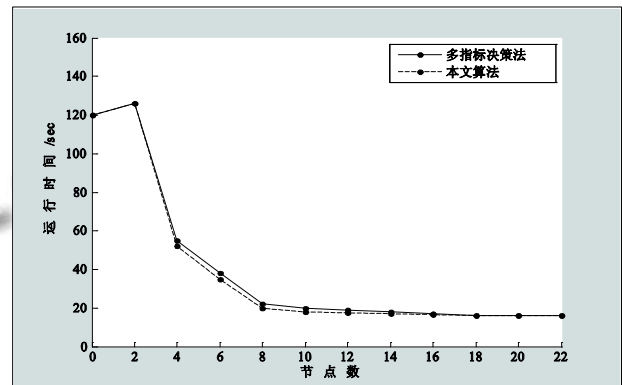


图 3 不同节点数下的程序执行时间

5 结语

本文充分发挥非关系数据的优势构建了分布式模型, 通过引入了查询成本指标改进了负载均衡算法. 经过仿真实验, 将所提出的负载均衡算法与目前较流行的多指标动态负载均衡算法进行了分析对比. 结果表明, 对于查询计算统计任务, 改进后的负载均衡算法可有效减少程序执行时间, 提高加速比和并行效率.

参考文献

- 董静宜, 王鹏, 陈磊. 云计算集群系统负载均衡算法的熵判定值. 成都信息工程学院学报, 2010, 25(6).
- 刘安丰, 陈志刚, 曾志文. 基于数据挖掘的 Web 集群负载均衡算法. 计算机工程与应用, 2003, 25(61).
- 范丽, 王燕燕. 基于改进蚁群算法的集群负载均衡研究. 计算机时代, 2007, 10.
- 徐群, 祝永志. 集群系统中的负载均衡问题的研究. 计算机技术与发展, 2009, 19(8).
- 颜开. NOSQL 笔谈. 程序员, 2010, 2:1-37.
- 郭欣. 异步计算的两把尖刀. 出版地: 程序员, 2009, (9):71.
- 戴文海, 陈红. 基于并行数据仓库的数据分布调整策略. 华中科技大学学报(自然科学版), 2005, (12):242.