Performance and Resource Modeling in Highly-Concurrent OLTP Workloads

问题背景

数据库中的某些查询可能会占用大量的系统资源（CPU、RAM、IO）；数据库的并行执行有时会拖慢数据库的查询速度，例如：对于共享资源（shared resources）的竞争。因此，需要对数据库的负载进行建模，了解负载对资源的占用以及预测某时某刻数据库的工作负载。为了解决这些挑战，本文的作者提出了DBSeer系统。

方法

收集SQL查询、数据库系统和操作系统的日志信息，从中提取出一些有用的统计信息；然后对负载建模。总体的架构如下：

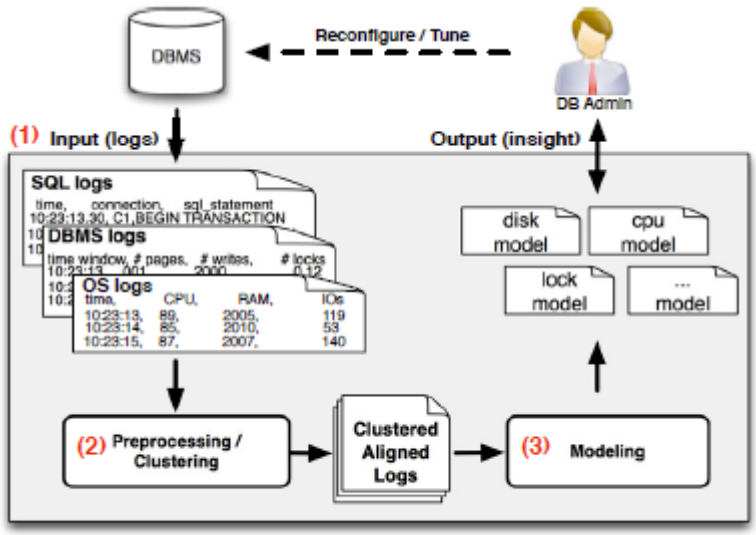

Figure 1: Workflow in DBSeer.

包括三步：

1. 收集日志
2. 预处理
3. 建模

预处理阶段

预处理阶段处理的信息包括：

- 执行的数据库语句和事务；
- 事务的延迟；
- OS信息：

> per-core CPU usage, number of I/Oreads and writes, number of outstanding asynchronous I/Os, total network packets and bytes transferred, number of page faults,number of context switches, CPU and I/O usage

- MySQL全局状态变量：

> the number of SELECT, UPDATE, DELETE, and INSERT commands executed, number of flushed and dirty pages, and the total lock wait-time.

事务聚类

聚类的目的是为了划分事务的类别（i.e.they access the same tables in the same order and perform similar operations on each table）。首先遍历日志将每个事务表示如下：

$$[t0(mode1, table1, n1, t1), \cdots, (modek, tablek, nk, tk)]$$

具体含义参照3.2节。

使用的聚类方法是DBSCAN

确定事务（每类）的分布

建模硬盘IO和RAM

影响IO的三个因素：

- 写日志
- 写脏页
- Cache Miss产生的IO

脏页写回

当日志中的redo日志写满之后，需要刷新内存中的数据到磁盘中，保证数据的一致性。这时就产生了脏页IO。作者对IO进行预测。具体来说，作者用p表示每类事务写数据的概率分布，然后将page划分成三类：脏 页但脏数据在较老的日志中；脏页但脏数据在当前日志中；干净的page。然后推导出page落到每个类别中的概率，接着统计第一类page的数目。由于脏页的刷新率和第一类page相关，从而推到出IO次数。

锁竞争模型

Thomasian模型

托马森模型是计算资源冲突概率的一类模型，假设事务的类别是固定的，同时数据库包含固定数目的区域（数据表，行）。每类事务以固定的概率访问特定的区域，申请同样数量的锁。在计算中，托马森模型综合产生冲突的概率，事务的数量以及平均等待时间，通过迭代的方式计算直到算法收敛。

在本文的工作中，作者对托马森模型的假设进行了改进，并对计算过程进行调整，使之更符合显示情况。详见5.2节。

黑盒模型

线性模型

由于CPU的使用率，网络包的数目等和TPS（每秒到达的事务数量）呈线性相关，因此训练出线性模型进行预测。

非线性模型

对于非线性相关，如锁冲突、Disk IO和TPS的关系，使用下面的模型进行训练

- Polynomial fitting
- Kernel Canonical Correlation Analysis
- Decision Trees
- Neural Networks

实验

数据集

TPC-C

Wikipedia transactions

图表

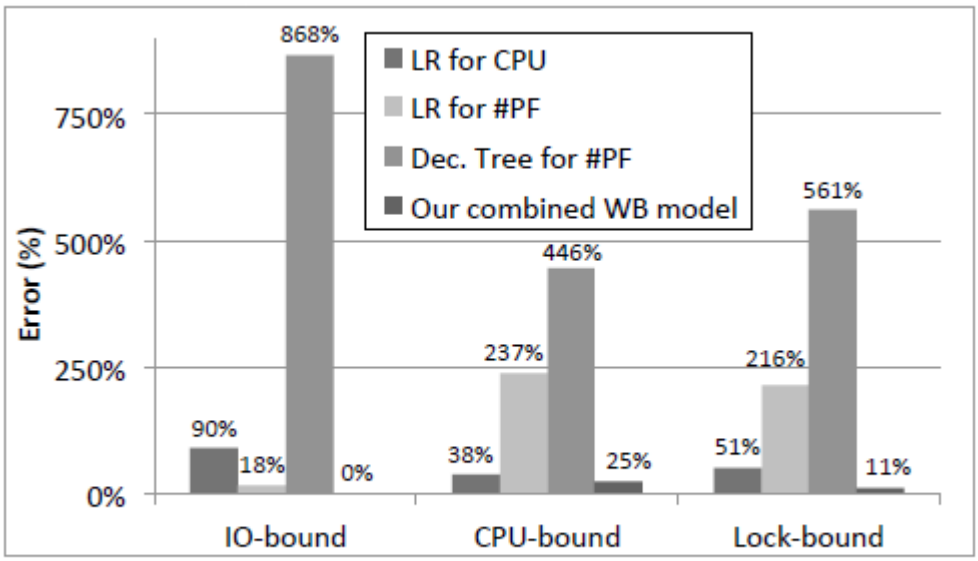

Figure 3: Max throughput prediction.

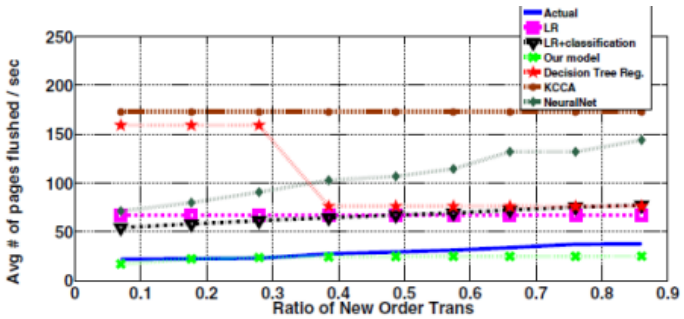

Figure 6: Predicting average page flush rate.

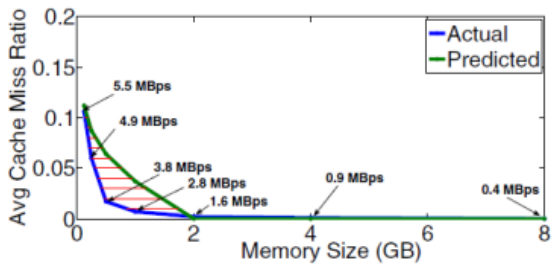

Figure 5: Cache-miss rate for memory provisioning and physical read prediction using the original TPC-C benchmark.

从图表可以看出，使用黑盒模型（机器学习相关方法），准确率较难保证。

需要进一步了解

MySQL的redo日记机制

托马森模型