On Predictive Modeling for Optimizing Transaction Execution in Parallel OLTP Systems

问题背景

马尔科夫模型

本文使用到的马尔科夫模型是对存储过程进行建模，在概念上类似于状态图，包含顶点（执行状态）和边，其中执行状态指的是单个查询的某个具体的调用过程，包含4个方面的信息：查询语句（名称）、该查询语句之前在存储过程中被执行的次数、该语句涉及到的数据所在的分区、该顶点的上层顶点已经获取到的分区。同时顶点还包含一些概率（读磁盘/写磁盘概率，Abort概率等）（右图的概率表）示例如下：

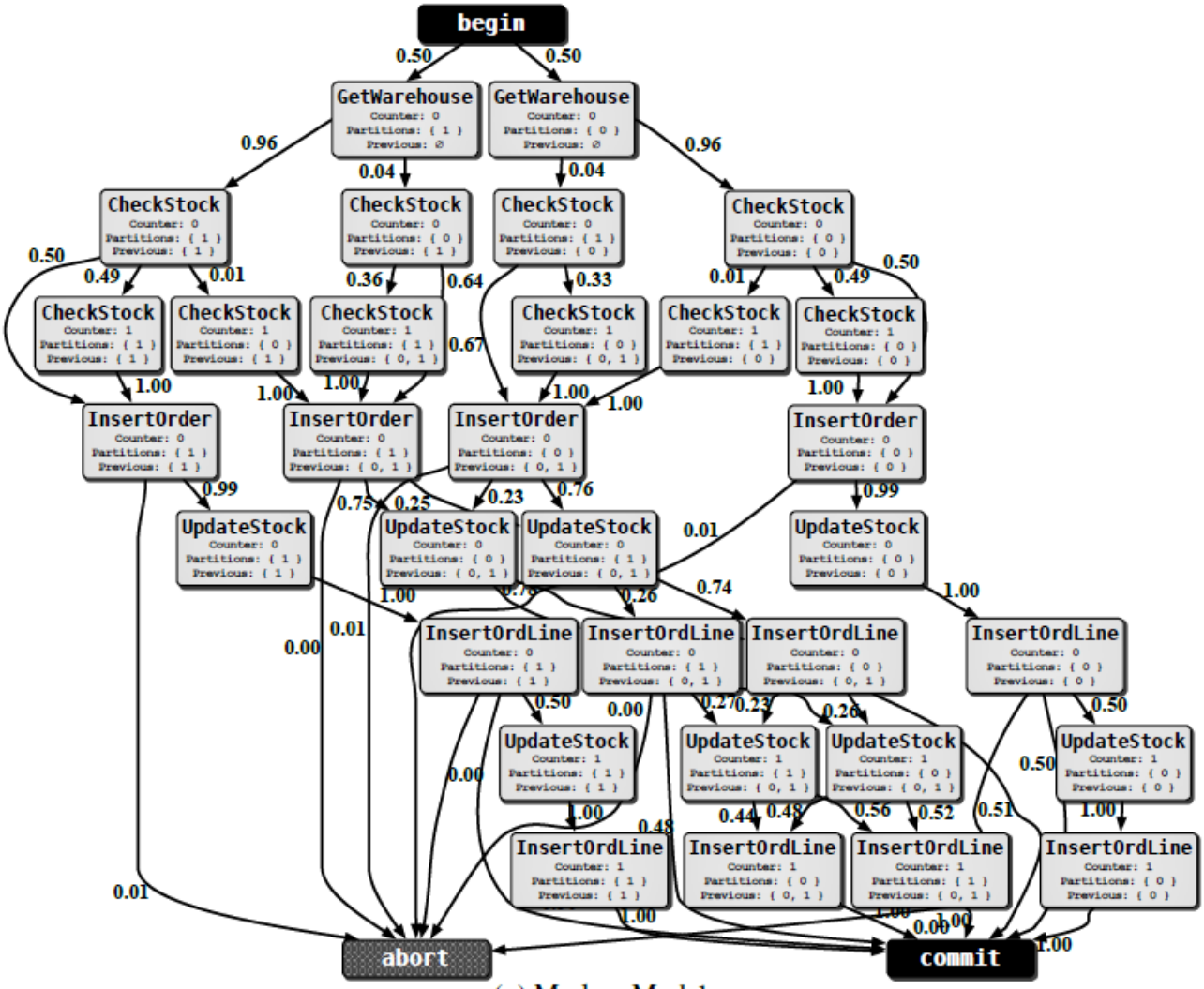
GLOBAL

Single-Partitioned: 0.95
Abort: 0.01

PARTITIONS

	Read	Write	Finish
Partition 0	1.00	1.00	0.00
Partition 1	0.05	0.05	0.95

边代表执行状态的转移概率，该概率是在sample中边被遍历的次数除以顶点被遍历的次数（某个顶点转移到其他顶点的概率和为1）。整个的马尔科夫模型的示例如下：



(a) Markov Model

建模的目的

建模之后，当新的存储过程到达之后，可以将事务包含的查询映射到马尔科夫链的某个实例顶点上，就是说这样可以预知一个查询会执行的动作（需要获取哪个分区的数据，多大概率abort等）。在文中作者列举出了4类优化：

> Execute the transaction at the node with the partition that it will access the most.

由于文中针对的是分布式环境，当一个事务到达之后，最优的选择是在分区包含的数据是该查询涉及到的数据占绝大多数的分区上进行执行，这样可以减少数据的移动（如单分区事务）。这样的分区被成为“基分区”（base partition）。

> Lock only the partitions that the transaction accesses.

为了保证数据的一致性，并行事务需要对数据进行加锁。如果可以“预知”那些数据会被一次查询使用，就可以保证更细粒度的锁级别（分区锁）。

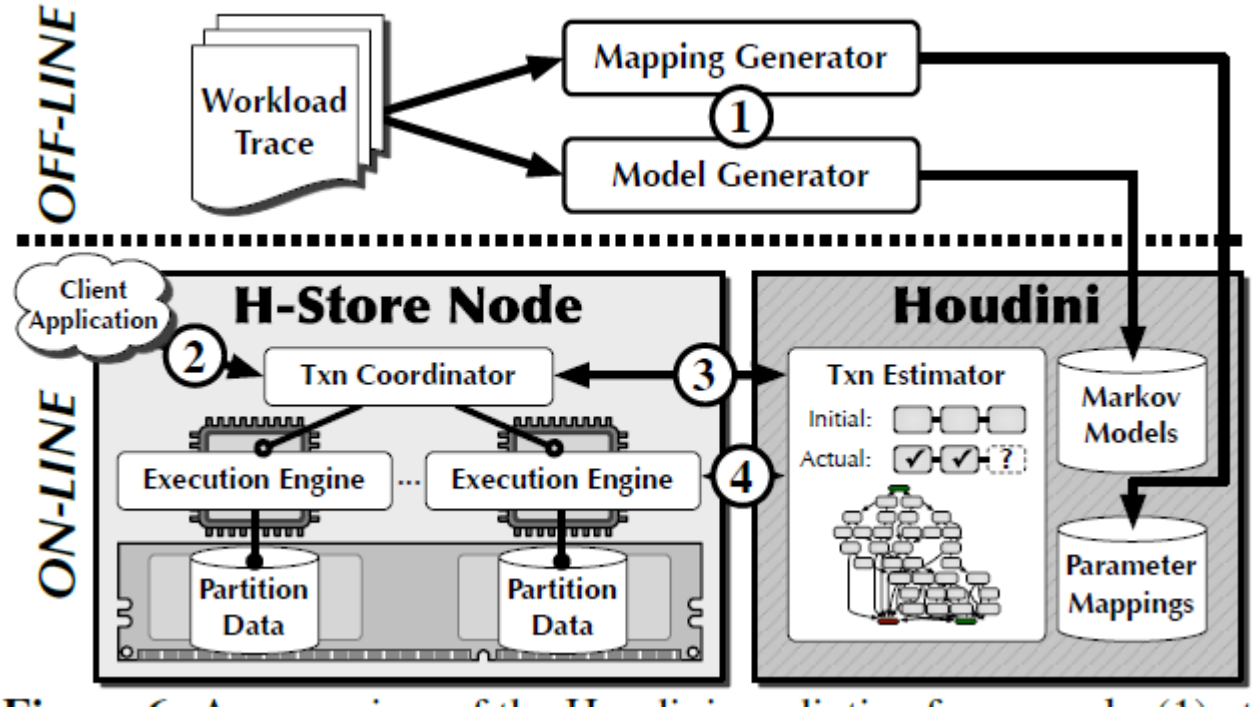
> Disable undo logging for non-aborting transactions.

如果事务不会abort，也就没有了写undo日志的必要

> Speculatively commit the transaction at partitions that it no longer needs to access.

为了保证数据库的一致性，很多分布式数据库使用两阶段提交协议，产生额外的通信开销。如果一个查询是该次事务的最后一个操作，则可以将查询结果和提交请求（”I am prepared to commit“）一起发送给master。

事务建模



事务建模分成两步：

产生顶点和边

对于负载中的事务，调用数据库API决定事务会获取的分区，生成一个执行状态的四方面信息（名称、次数、要获取和已经获取的分区）。然后根据执行的顺序遍历路径，将顶点和边添加到模型中。模型中同时包含begin、abort、commit三个定点。

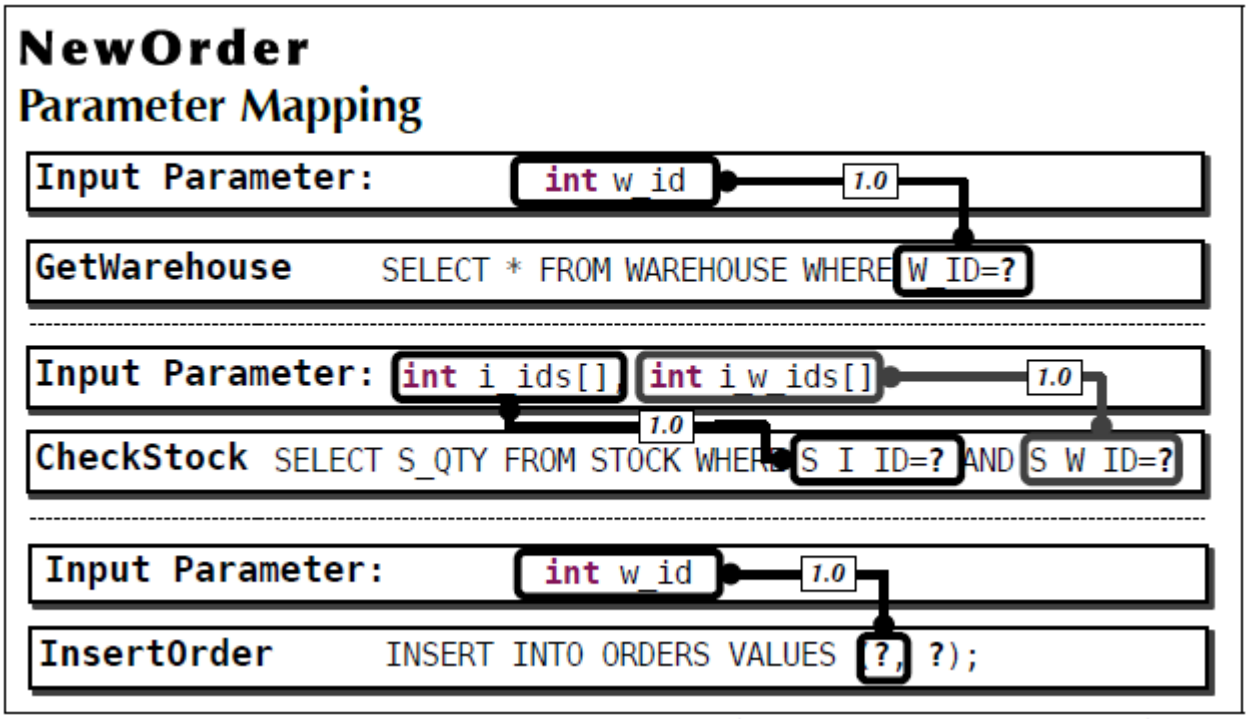
计算转移概率

转移概率的计算方法在上文已经提到。然后计算顶点的概率表，计算过程自下而上：commit和abort的相关概率都为1，然后根据边转移概率计算上层顶点的对应概率（最长路径优先）。

预测框架

参数映射

该步骤将存储过程的参数信息映射到查询语句的参量上，在sample中计算两个变量的相关系数。示例：



初始路径计算

对于一个新的事务，该步骤将事务中的查询映射成模型中的一条路径中的顶点。从begin顶点出发，对模型中所有可能的下层定点进行遍历，从中过滤出"valid"顶点，如果有超过一个以上的"valid"顶点，选择模型中边转移概率最大的顶点，添加到路径中。一个顶点"valid"需要满足的条件：

> （1）we can determine all the query parameters needed for calculating the partitions accessed by that state's query
>
> （2）the next state's set of previously accessed partitions contains all the partitions accessed by the transaction up to this point.

示例参考4.2节

初始优化选择

按照顶点的概率，计算会应用到的优化（OP1-OP4）

更新优化&模型维护

按照产生的优化路径和优化动作，事务在队列中进行执行。当优化器在执行过程中发现更优的执行时，重定向执行过程， 此时Houdini（本文的模型）会记录这条路径并添加到模型中。当模型不再精确时，Houdini**重新**计算边和顶点的概率信息。

分布式模型

本文提到的模型在事务很多的情况下占用较大的内存。为了减少内存开销，使用聚类的方法对模型进行聚类，分布式存储在不同的顶点上，然后使用决策树找到事务评估需要使用到的模型所在的节点。

实验

Benchmarks

TATP:The Telecom Application Transaction Processing

TPC-C

AuctionMark:developed by Brown University and a well-known Internet auction company

模型精确度

		TATP	TPC-C	AuctionMark
OP1	Global	95.0%	94.8%	94.9%
	Partitioned	94.9%	99.9%	94.7%
OP2	Global	98.9%	90.9%	90.7%
	Partitioned	100%	99.0%	95.4%
OP3	Global	100%	100%	100%
	Partitioned	100%	100%	100%
OP4	Global	99.5%	100%	100%
	Partitioned	99.5%	95.8%	99.9%
Total	Global	**94.9%**	**93.8%**	**85.6%**
	Partitioned	**94.9%**	**95.0%**	**90.2%**

Table 3: Measurements of the global and partitioned Markov models' accuracy in predicting the execution properties of transactions.

模型开销

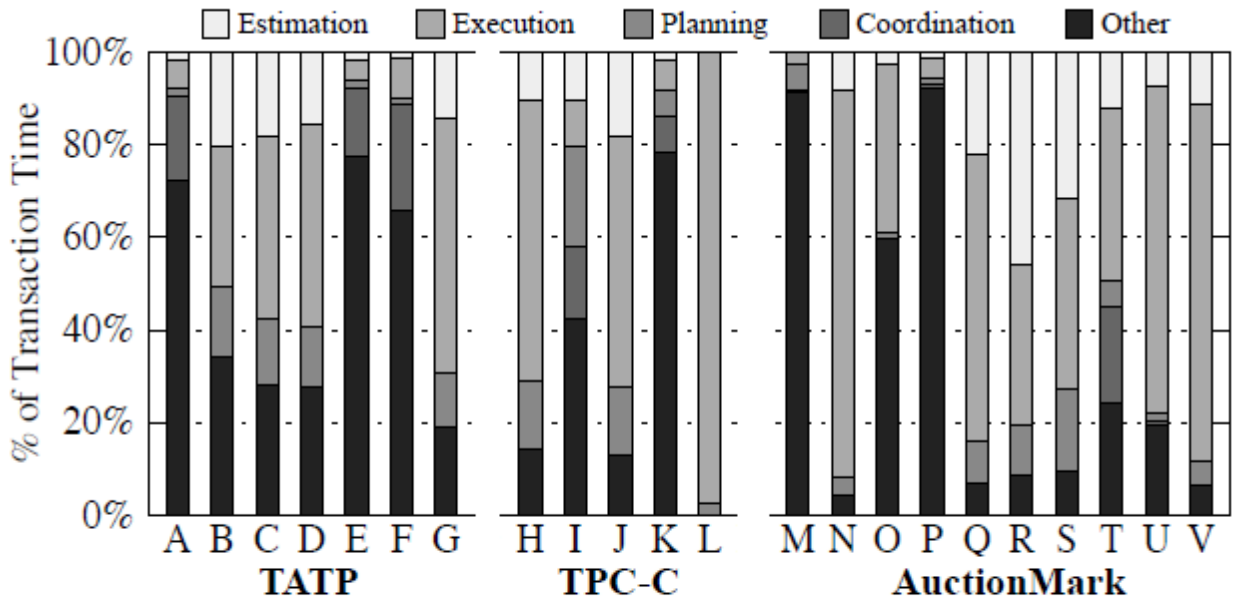


Figure 11: Relative measurements of the time spent for each transaction (1) estimating optimizations, (2) executing, (3) planning, (4) coordinating its execution, and (5) other setup operations.

总结

本文的想法使用马尔科夫方法从sample负载中训练出模型，然后对一个事务进行预测。借鉴了机器学习（觉得和图片识别类似）领域的方法。但在具体的过程中，由于数据库的执行过程远比图片识别简单，需要考虑到很多优化信息，应该还存在进一步优化的地方。