

各位老师同学下午好，我是王国畅。我要介绍的工作是通过深度强化学习 对数据库系统进行 query 敏感的配置调整。从题目可以看出这份工作的核心贡献是为数据库配置 调整 提供了一种新方法。我先来解释一下数据库配置调整问题，也就是Knob tuning问题。

数据库通常有配置文件用于调整数据库的一些参数，例如等待请求的最大时延等。我们把这些参数称为knob，knob意思是旋钮，这些参数就像旋钮一样可以调节数据库的行为，进而影响它的性能。Knob tuning的目标就是让数据库对于某项任务能用最佳配置从而取得最佳性能。例如在批处理时我们希望数据库有高吞吐量低延时，而在一些混合需求的场景下，比如HTAP场景下，我们希望数据库能够通过切换配置使数据库达到最佳性能。

现在的数据库配置参数量一般都在100以上，这里列举了MySQL，POSTGRESQL，Mongodb作为参考。显而易见如果对每个参数的取值进行枚举再搜索最佳方案，会直接组合爆炸。knob tuning问题也在早先就被定义为NP-Hard问题。因此先导工作通常采用巧妙的方式解决优化问题，我下面就简单介绍三个先导工作的基本思想和他们的不足。

三个先导工作分别是BestConfig，OtterTune和CDBTune

先说这个BestConfig。他的基本思路有些像一些编程软件的代码自动补全，他们会从你敲代码时输入的函数名变量名历史搜索几个出来作为你补全的建议，BestConfig从数据仓库管理员为数据库做的历史配置里，搜索出一个最适合于当前情况的配置。那么显而易见这样的做法非常依赖于历史记录，如果没有近似场景的配置文件，就无法得到一个适应于当前场景的配置，如果训练历史中不存在高质量配置，也往往无法做到性能上的优化。

第二个尝试是Ottertune，他的基本思想还是依赖于数据管理员的配置文件，但他试图学到管理员应对不同场景进行配置的经验，来让模型写出一个配置。机器学习里非常重要的模块就是数据集，这种训练过程的瓶颈就在于难以获得数量足够大且质量足够高的数据集，也就是数据库管理员的配置经验，来进行模型的训练，因此这个模型训练的数据获取很困难，同时由于配置文件各个数据库都有差别，一个数据集无法迁移到另一个数据库进行使用，即使一个数据库上实现了这样的工具，也难以迁移另一个数据库，仍需要在另一个数据库上进行数据筹备。

第三个工作是CDBTUNE，他的基本思路是通过深度学习模型，用一个试错的策略去训练，也就是我先给出一个配置，然后用这个配置去处理一些对数据的请求，再去评估这个配置，从而得到训练数据。这样的做法虽然解决了Ottertune中的数据难以获得的问题，但是第一点它的试错过程非常麻烦，甚至可能需要人工的参与，使得整个训练过程很耗时，第二点由于它试错的过程很耗时，很难对一条一条请求进行试错评估，所以CDBTUNE只保证做到在workload及以上的粒度上进行配置优化的训练。第三点就是它使用的是已经预训练的模型进行调整，所以它同样依赖于高质量的配置历史数据也同样面对迁移困难的问题。

三个先前的工作介绍完了，我们来说现在的state of art，叫做QTUNE，也就是我们报告的这份工作，发表在VLDB19，是清华和华为联合的一份工作，感兴趣的同学可以自己去找查看一下。

QTUNE做到了三个提升：第一它支持更加细粒度的配置优化，最小支持query粒度，即它最小粒度下可以针对query来调整配置。第二它训练过程是易于构建的，QTUNE同时在PostgreSQL，MySQL和MongoDB下实现了工具，第三就是它对比之前的调参工作，能够使数据库取得更好的性能。

QTUNE对比Ottertune和CDBTune的一个重大改变是它不依赖于管理员的配置历史，而是依赖于Query数据，通过将SQL语句进行特征处理，得到一个向量，并通过Query的cost来打标签，完成一个训练数据集的构建，这样的做法可以通过很低的成本的得到用于训练的数据，而且基本可以自动化进行数据集构建。

另外，QTUNE在模型构建方面提出了一个双状态决策策略梯度模型，简而言之，它的模型是由两个模型复合而成，结合了两个模型的特点。

我们结合这张架构图说一下整个过程。workload中的query进来以后，Qtune抽取其中的特征，形成一个Vector也就是向量，这时如果用户需要的是query粒度的配置调整，Qtune就可以直接通过第一状态的模型输出优化的配置。如果是更粗粒度的，例如workload或cluster粒度，Qtune就通过第二状态模型来把query分组，每组得到一个优化配置，通过这个优化配置对这个组里的query进行处理。

最后就是这个工作的性能评估，总结一下就是在粗粒度上，与之前的工作对比QTune总是可以做到更大的吞吐量和更低的延时。之前的工作没有细粒度上的配置优化，所以细粒度上没有横向的对比，Qtune自身的细粒度和粗粒度相比，细粒度有着明显低于粗粒度的延时，但吞吐量也很小，不过我认为这个特征也更符合流式处理的场景需求。

根据老师的要求我本来应该评价一下他的缺点，因为我不是专门做数据库的，所以难以从数据库的角度去找他的缺陷，我觉得从机器学习和软工的角度来讲这个工作的训练设计和模型架构做的很干净，步骤和思路上都没有什么可以指摘的地方，所以我没有评价他的缺点，报告就到这里，谢谢各位。