

# 分布式系统中的控制成本研究综述

方思桐

(华中科技大学 计算机科学与技术学院, 武汉市 430070)

**摘 要** 分布式系统可以是同质的(集群),也可以是异质的,如网格、云和 P2P。在这些类型的系统中会出现一些问题,如服务质量、资源选择、任务分配、负载平衡和容错。在过去的几十年里,人们对分布式系统的上述问题都给予了总结调查,现有的调查都只是对一种单一类型的分布式系统的技术现状做了非常初步的回顾。不难发现,以上技术的发展创新,也离不开一个目的,即成本节省。因此,本调查从成本的角度出发,调研了三篇文献。Leis 与 Kuschewski 提出了一个简单而直观的模型,该模型将工作负载、硬件和成本考虑在内,以确定最佳实例配置。Durner 等人提供了一种新的智能缓存存储架构 Crystal,与计算机同地协作,能够大大改善查询延迟,同时节省了远程存储的带宽。Microsoft Azure 团队提出的 Seagull 是一个采用数据科学来预测用户负载、优化资源分配、控制成本的基础架构。

**关键词** 分布式系统, 数据中心, 成本, 资源分配

**中图法分类号** TP338.8

## A Survey of Task Allocation and Load Balancing in Distributed Systems

Sitong Fang

(Department of Computing, Huazhong University of Science and Technology, Wuhan 430070, China)

**Abstract** Distributed systems can be homogeneous (cluster), or heterogeneous such as Grid, Cloud and P2P. Several problems can occur in these types of systems, such as quality of service (QoS), resource selection, load balancing and fault tolerance. In past decades, although there have been some related surveys about this subject, each of which only made a very preliminary review on the state of art of one single type of distributed systems. We can find that the above technology development innovation cannot be separated from the common purpose of cost control. Therefore, this survey investigates three papers from a cost perspective. leis and Kuschewski propose a simple and intuitive model that takes the workload, hardware, and cost into account to determine the optimal instance configuration. Durner et al. advocate a new architecture of a smart cache storage system called Crystal, that is co-located with compute. It can significantly improve query latencies while also saving on bandwidth from remote storage. Seagull, proposed by the Microsoft Azure team, is an infrastructure that uses Data Science to predict user load, optimize resource allocation, and control costs.

**Key words** Distributed systems, data centers, controlling costs, resource allocation

## 1 引言

不同的文献中给出了分布式系统的各种定义,归纳总结为分布式系统是一个独立的计算机集合,在其用户看来是一个单一的连贯系统。其分布式计算单元通过网络连接和组织,以满足大规模和高性能计算的需求,在过去的几十年里受到了相当大的关注。分布式系统有很多类型,如网格、点对点(P2P)系统、特设网络、云计算系统、普适计算系统和在线社交网络系统。归纳总结,真正的分布式系统具有以下一般特征:

**缺少一个全局控制单元。**由于分布式系统的大规模、动态和异质性,很难为整个系统实现一个全局控制单元。例如,在临时网络中,无线节点可以在没有固定的基础设施和没有中央控制的情况下相互通信;在传感器网络中,许多传感器节点是密集部署的,由于大量的开销和大量的传感器,可能没有全局识别;在 P2P 系统中,采用了分散的架构,网络中的各个节点既是资源的提供者,又是使用者,与集中的客户-服务器模式相反,客户节点请求访问中央服务器提供的资源。

**资源的分配和共享。**系统中存在各种资源,如数据、流行网络对象的副本、处理器、计算能力和磁盘存储。资源分布在各节点之间,各节点经常同意相互分享其本地资源以执行任务。为了实现资源的有效共享,资源管理是非常关键的,它通常包括资源放置、资源提供、资源发现和资源协商。此外,资源缓存经常被用来提高资源访问的性能,一些节点缓存资源以服务未来的请求,而不是从原始节点获取资源。

**系统的开放性和不可靠性。**分布式系统通常是开放的,一些节点和网络结构可能是不可靠的。例如,没有有效的方法来防止恶意的对等人加入开放系统,因此分布式系统非常容易被自私和恶意的用户滥用。此外,由于节点的自主性,一些节点可能无法独立运行。因此,可靠性和容错性对于真正的开放分布式系统是至关重要的。

**异质性节点。**在一些分布式系统中,节点是异质的。例如,在网格中,节点可能归属于不同的组织;在传感器网络中,节点可能有不同的能力和感应范围;在社会网络系统中,节点可能有不同的响应时间和扩散的阈值。当任务执行时,异质节点应相互协调,以使总效用最大化。此外,异构节点在

执行任务时可能会竞争关键资源。

因此,分布式系统中的应用是多种多样的,如网络服务、科学计算和文件存储等。一般来说,分布式系统中的应用可以分为若干任务,并在不同的节点上执行。根据分布式系统的一般特征,在这些类型的系统中会出现一些问题,如服务质量、资源选择、任务分配、负载平衡和容错。例如,分布式系统中的应用的性能取决于构成应用的任务在可用节点上的分配,被称为任务分配问题。如果太多任务挤在某些节点上,可以将任务从负担重的节点切换到负担轻的节点,以减少任务在节点上的等待时间,这被称为负载平衡。

若从**经济学**的角度来考虑问题,将所有问题抽象出一个共同点,在云中,资源(如 CPU 内核、DRAM、SSD、网络)是可替代的,可以相互交易,货币作为交换媒介。任务的优化问题成为了成本优化问题。

三篇被调研的论文,便可被视为从不同的方面解决了成本优化问题。Leis 与 Kuschewski 提出了一个简单而直观的模型,该模型将工作负载、硬件和**成本**考虑在内,以确定最佳实例配置。Durner 等人提供了一种新的智能缓存存储架构 Crystal,与计算机同地协作,能够大大改善查询延迟,同时节省了远程存储的带宽,降低**成本**。Microsoft Azure 团队致力于保证为客户提供高质量的服务,特别是在客户活动频繁的时期,同时控制**成本**。他们提出的 Seagull 能够采用数据科学来预测用户负载、优化资源分配,实现了控制成本。

## 2 原理和优势

### 1.1 Cost-Optimal Query

公共云提供商提供数百个异构硬件实例。对于分析查询处理系统,这提出了一个主要的挑战:根据硬件配置,性能和成本可能会有数量级的不同。Leis 提出了一个简单而直观的模型,该模型将工作负载、硬件和成本考虑在内,以确定最佳实例配置。基于模型的方法能够显著降低成本,并指导云本地数据库系统的发展,以实现成本最优查询处理的愿景。该模型的关键特征:

1) 假设一个能够充分利用所有可用硬件资源的假想系统,而不是模拟任何具体的、现有的数据库系统及其所有的性能瓶颈,即该模型为工作负载成本提供了一个下限。

2) 该模型是在云中构建成本最优的 OLAP (联机分析处理) 系统的基础。

3) 基于成本的方法支持严格的、数据驱动的策略制定, 这将有助于使系统更接近可用硬件的极限, 特别是在随着时间的推移演进系统时。

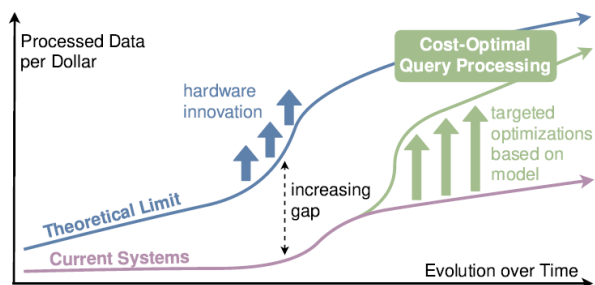


图 1 基于模型的硬件/软件联合进化以消除 gap

如图 1 所示, 硬件的发展与现有系统之间, 存在着发展差距。间距产生原因为引擎效率低下, 尤其是引擎没有跟上硬件创新所导致的效率低下。有了这个模型, 用户可以预测好的工作负载的正确硬件选择, 也可以直接应用在现在的系统, 但从长远来看, 可以使用该模型来发展现有的系统, 以找出在哪些领域, 例如网络, 他们必须改进, 以赶上当前硬件的可能。总的来说该基于模型的方法通过模拟能够更好的选择硬件与实例, 显著降低成本。此外, 该模型能够指导云原生数据库系统的发展, 以实现成本最优的查询处理的愿景。

## 1.2 Crystal

在云的灵活性和随用随付功能的驱动下, 分析数据库系统向云的范式转变。此类数据库采用分级或分类存储模型, 弹性计算层访问存储在独立可伸缩的远程云存储(如 Amazon S3 和 Azure Blobs)上的数据。目前, 包括 Apache Spark、Greenplum 等在内的, 几乎所有大数据系统都支持直接查询云存储。云供应商也提供云服务, 如 AWS Athena、Azure Synapse 和谷歌 BigQuery, 从而满足用户日益增长的需求。

鉴于远程存储的延迟相对较高, 带宽较低, 在计算节点缓存数据变得十分重要。因此, 我们看到分析的缓存技术再次飙升, 热数据被保存在计算层的快速本地存储(如 SSD)中, 但其大小有限。

基于此些问题, Durner 等人提出了一种新的智能存储中间件 Crystal, 它与数据库解耦, 位于 DB 和原始存储之间。Crystal 可以被看作是一个用于存储的微型 dbms, 或者缓存管理系统(CMS)。

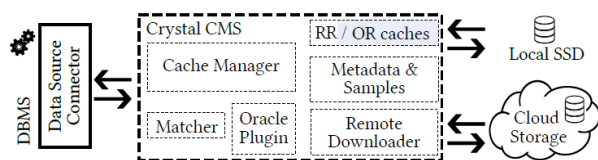


图 2 Crystal component

图 2 显示了 Crystal 内容管理系统 (CMS) 的详细情况。它维护着两个本地缓存, 一个是小的请求区域 (RR) 缓存, 一个是大的数据库区域 (OR) 缓存, 分别对应短期和长期知识。这两个缓冲区都以高效的列式开放格式 (如 Parquet) 存储数据。Crystal 通过 Crystal API 接收来自连接器的查询。一个查询包括一个对文件 (远程路径) 的请求, 并带有下推谓词。Crystal 首先与匹配器检查, 看它是否可以使用一个或多个缓存区域来覆盖查询。如果是 (缓存命中), 它从本地存储中返回一组文件路径。如果是 (缓存缺失), 将有两个选项:

(1) 它响应远程路径, 以便连接器可以像往常一样处理它。Crystal 可以选择请求连接器将下载和过滤的区域存储在其 RR 缓存中。

(2) 它从远程下载数据, 应用判断式, 将结果存储在 RR 缓存中, 并将此路径返回给连接器。

Crystal 作为一种新的智能缓存存储系统的架构, 与计算系统共处一地协同操作。Crystal 的客户是针对 DBMS 的数据源, 具有向下推的判断式。与 DBMS 的作用相似, Crystal 包含了查询处理和优化组件, 重点是高效地缓存和提供称为区域的单表超矩形。结果显示, Crystal 与一个小型的 DBMS 特定数据源连接器, 可以大大改善未修改的 Spark 和 Greenplum 的查询延迟, 同时还可以节省远程存储的带宽。

## 1.3 Seagull

主流的云服务提供商目标是保证为客户提供高质量的服务, 同时控制运营成本。但手动实现这些冲突的目标是劳动密集型的, 耗时的, 容易出错的, 既不能扩展, 也不能持久。因此, 这些供应商转向了自动管理服务。为此, 数据科学 (DS) 技术用来预测资源需求, 并利用这些预测来优化资源分配。

一个能够分析每个系统组件的历史负载、预测其未来负载并利用这些预测来优化资源分配的基础设施, 对于许多产品和用例来说是有价值的。在过去的两年里, Microsoft Azure 团队已经建立了这

样的基础设施,称为 SEAGULL,并将其应用于两个场景:

- (1) PostgreSQL 和 MySQL 服务器的备份调度
- (2) SQL 数据库的抢先自动伸缩。

这些场景要求团队在所有 Azure 区域对基础设施进行实战测试,使其对 SEAGULL 方法的高影响力和通用性有信心。

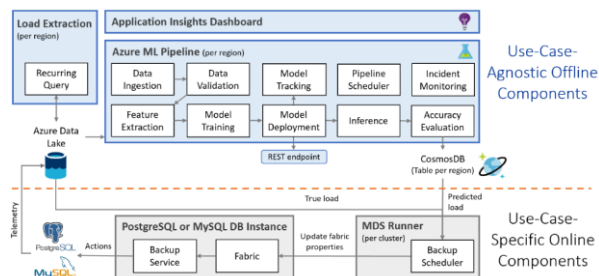


图 3 SEAGULL 的基础结构

如图 3 所示,为了实现所定目标,SEAGULL 将用例区分的组件和特定于用例的组件组合在一起。且该方法具有以下关键创新特点:

(1) 设计并实现了一个端到端的 SEAGULL 基础设施,并将其部署在所有 Azure 区域,以优化 PostgreSQL 和 MySQL 服务器的备份调度。

(2) 团队进行了全面的分析,将 PostgreSQL 和 MySQL 服务器根据其典型的客户活动模式划分为相同的组。大多数服务器要么是稳定的,要么遵循每天或每周的模式。

(3) 将每个服务器在备份日的低负载窗口预测的准确性定义为两个指标的组合。

(4) 应用了几个通常用于时间序列预测的 ML 模型(NimbusML、GluonTS 和 Prophet)来预测不稳定服务器的低负载,这些服务器不遵循可被持久预测识别的模式。

### 3 研究进展

分布式系统要求存得下不够,还要存得好。存得好,主要是为了节省成本,因为海量数据随之带来的,就是海量成本。同样,算得快也不够,还要尽可能压榨机器计算性能,节省成本。有别于分布式存储引擎在存储方式上产生创新,由于计算逻辑优化的空间有限,并且相对属于应用层需要考虑的问题,分布式计算框架更多尝试对资源的调度做优化。也就促成了一系列所谓 Resource Manager 的诞生。本篇文章中提到的三个工具,虽不是典型的

Resource Manager,但可以从资源管理,成本节省的角度去看待。

首先,我们先来看一下:什么是资源管理器。以往,公司不同业务都会有自己专享的各种服务器和集群。每个业务都希望能独立运行,不受其他程序的干扰,也不留下安全隐患。但业务负载是动态变化的,规律性的波峰波谷,不规律的业务增长和下降,随之而来的就是频繁的扩缩容。为了降低扩缩容带来的工作量和对业务的影响,一般都会留有资源余量。所谓余量,换一个说法,就是浪费。怎么灵活又及时地把这些浪费掉的资源利用起来,顺便减轻扩缩容带来的维护成本,就是各种资源管理器要解决的主要问题。

混部,也就是多租户,可以解决这一问题。对多租户的支持,好处自然是整体资源率的提升,于整体有利;坏处就是牺牲了个体,失去独立性就意味着相互影响。

总的来说,分布式系统用来节省成本的办法思路为:分布式计算框架更多尝试对资源的调度做优化,以此节省成本;提升资源整体利用率的方式是混部,错峰使用资源;混部就会相互影响,所以需要资源隔离;隔离又会导致资源借用受限,所以要软性隔离;资源借用需要有时间限制,所以要支持强杀;资源借用需要有容量限制,所以要能设置硬顶;没有绝对的公平,带权重和 DRF 都是折中和改进;平台部门是管理员,不掌握分配资源的权利,而应当提供足够的指标协助决策。

大致了解了分布式系统资源管理与节省成本后,接下来介绍一下被调查的三篇文章的研究进展与从成本的角度看待其进展。

首先是面向最优查询的模型,该模型的出发点即为节省成本。由于硬件配置,性能与成本之间可能会有数量级的差异。举例来看,比较 AWS EC2 实例上运行一个假想的查询引擎和 AWS Athena 的成本,查询扫描 S3 中的 1tb 数据。如果在 c5n 上执行这个查询,有 72vcpu 和 100gbit 的峰值网络带宽。在 80gbit/s 的实际速率下,扫描来自 S3 的 1tb 数据需要 100 秒。在这 100 秒内,查询引擎必须使用 72 个 vCPU 处理 10 GB/s 的内存,相当于每个 vCPU 处理 142MB/s。现代查询引擎甚至可以以这种速度处理中等复杂的查询。这意味着我们的 1 TB 查询的执行可能是网络绑定的,耗时 100 秒。如果利用每秒计费,我们在 100 秒后关闭实例,即每小时按需收费 c5n.18xlarge \$3.89 的价格导致总查询成本



仅为\$0.11。另一方面，对于同一个查询，Athena的成本是5美元，也就是说，在这个场景中，Athena几乎要贵50美元。使用现货或预留实例而不是按需实例将使差距进一步扩大到100-200倍。

因此，该模型考虑分布式数据仓库系统(eg. Redshift)，在云中运行大型分析查询产生的问题：

(1) 是使用平衡的配置更好，还是专门关注一个资源(CPU、网络、SSD或DRAM)?

(2) 单个大实例是否优于多个较小的实例?

(3) 哪个公共云(AWS, Azure, GCP)最便宜?

(4) 应该在实例上缓存数据，还是总是根据需求从分布式文件系统读取数据?

(5) 将工程精力投入到优化网络或存储I/O堆栈上会更好吗?

该模型的关键思想为：在异构的云硬件环境中，目标必须是最小化工作负载成本，而不是运行时间成本。有一个性能模型对于能够利用云中的所有可用硬件资源的假设的、理想化的系统是有效的。即可以实现：成本优化作为新系统的基准；基于证据的性能特征优先化；关注市场定价；使用按需价格。硬件/软件共同发展；

inst.	cores		DRAM		SSD		network		cost ↑
name	#	/s	GB	/s	TB	/s	Gbit/s	/s	\$/h
c5n.18	36	9.3	192	49.4	-	-	100	25.7	3.89
c5.24	48	11.8	192	47.1	-	-	25	6.1	4.08
c5d.24	48	10.4	192	41.7	4×0.9	0.78	25	5.4	4.61
m5.24	48	10.4	384	83.3	-	-	25	5.4	4.61
i3.16	32	6.4	488	97.8	8×1.9	3.04	25	5.0	4.99
m5d.24	48	8.8	384	70.8	4×0.9	0.66	25	4.6	5.42
m5n.24	48	8.4	384	67.2	-	-	100	17.5	5.71
r5.24	48	7.9	768	127.0	-	-	25	4.1	6.05
m5dn.24	48	7.4	384	58.8	4×0.9	0.55	100	15.3	6.53
r5d.24	48	6.9	768	111.1	4×0.9	0.52	25	3.6	6.91
r5n.24	48	6.7	768	107.4	-	-	100	14.0	7.15
r5dn.24	48	6.0	768	95.8	4×0.9	0.45	100	12.5	8.02
i3en.24	48	4.4	768	70.8	8×7.5	5.53	100	9.2	10.85
x1e.32	64	2.4	3,904	146.3	2×1.9	0.14	25	0.9	26.69

图4 根据2020.6市场定价得到的花费排序

如图4所示，不同的实例在不同的变量下有着不同的花费。根据比较可以看到，c5n.18的总体花费最少。在实际选择中，便可模拟出最少花费实例，进行选择与购买使用。

但该模型也存在一些异议。由于只有6个变量，模型非常简单，实际上并不能准确预测任何现有系统的执行时间。试图对真实系统的性能行为进行建模会使模型及其预测几乎与被建模的系统一样难以理解。该方法的目标是拥有一个大致正确且易于推理的健壮模型，而不是一个试图完全准确的复杂

模型，这可能是无法实现的。模型的简单性也使评估变得快速：为了确定最佳实例，该方法详尽地列举给定工作负载的所有实例配置。

这个模型是基于一些想要证明的隐含假设。文中假设CPU工作可以在单个节点上完美地并行化(尽管忽略了超线程，从而留出了一些空间)。这是基于经验，这种可伸缩性确实可以在低于100个核心的系统上实现。另一方面，在分布式环境中，实现完美的可伸缩性更困难，节点的数量是无界的，这就是为什么选择引入可伸缩性分数变量。文中进一步假设网络、存储和CPU阶段是连续的而不是重叠的(使用总和而不是阶段持续时间的最大值)。虽然实际系统可能会尝试重叠这些阶段，但是具有重要网络或I/O阶段的查询也可能导致更多的CPU负载，这就是该方法在这种情况下选择更保守的原因。

总的来说，该模型考虑不同实例下的最小化运行时间。但最小化运行时本身并不是优化目标，而是在给定一些运行时约束的情况下，以美元计算的工作负载成本。对工作负载成本的关注直接导致了成本最优的概念，这是在某些云硬件环境中执行特定工作负载的最低金钱成本。

第二个方法为Seagull。该方法来自Microsoft Azure团队，Seagull是一个采用Data Science来预测用户负载、优化资源分配、控制成本的基础架构。SEAGULL的影响有两方面，即它改善了客户体验并减少了工程工作。可以看出，该方法依旧有控制成本的作用。Seagull的设计原则主要有模块化和可扩展性。文章对Seagull的架构，服务器分类进行了介绍，并定义了低负荷预测精度与比较了ML模型。

总的来说，文中提出的SEAGULL基础架构，用于云上的负载预测和优化资源分配。虽然该方法适用于广泛的用例，但本章仍通过备份调度场景来说明它。

第三篇文章中介绍的Crystal，是针对数据库的架构方法。他被设计为：对DBMS不需要修改就可以集成，可插拔的缓存。；关注于存储层，可以实现最大程度的DBMS间复用(Spark和Greenplum可以用一套缓存)；可以根据策略(策略可定制)自主选择cache的方法。

Spark因为支持S3 Select，所以不会只简单的把文件从远端下载下来使用，Crystal作为中间层会从DBMS接收带谓词下推的请求，类似SELECT

name, age FROM test WHERE age > 20 AND age < 35, 然后 Crystal 将请求转化为 AST (语法树) 并序列化到字符串里面, 类似 `and(gt(age, 20), lt(age, 35))`。

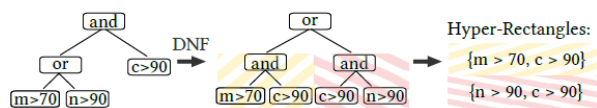


图 5 Crystal 实现将任何谓词树转换为超矩形的并集。

如果只是以上面序列化后的 AST 为 Key, 返回的文件为 Value 进行缓存, 意义不大, 因为只有相同的查询才会复用这块缓存, 因此要进行正则化。上图就是对一个请求进行等价变换 (否定下推、德摩根定律、否定推入谓词、否定范式转换为析取范式) 后的形态。这时一个请求会被拆分成若干个超矩形 (如图 5 的黄色与粉色两个限定就可以组成两个超矩形), 超矩形是合取 (and), 若干个超矩形的析取被称为 Region (超矩形之间是 or)。Cache 的粒度仍然是 Region, 论文也讨论了为什么不用超矩形作为 cache 粒度, 主要原因是粒度太低, 超矩形太多, 对 Cache 策略等额外代价也不友好。

总的来说, Crystal 是一个智能缓存存储系统, 在语义数据区域上运行, 并不断调整本地缓存的内容, 以获得最大利益。结果显示, Crystal 可以显著改善未修改的 Spark 和 Greenplum 的查询延迟, 同时还可以节省远程存储的带宽, 从而降低成本。

现有的分布式计算架构技术的实现是通过软件并行计算的支持, 服务器平台可以通过大量单机成本比大机器低得多的起步价, 快速实现应用上线, 并在用户量爆发时, 通过服务器叠加方式实现扩容。由于纵向拓展可优化空间太小 (单台服务器的性能上限很明显), 分布式系统强调横向扩展、横向优化, 当分布式集群计算资源不足时, 就要往集群里面添加服务器, 来不停地提升分布式集群的计算能力。因此, 分布式系统往往会产生各种意料之外的软硬件成本。

现有的技术关注分布式系统的服务质量、资源选择、任务分配、负载均衡和容错等问题, 技术的进步除了使分布式系统运行速度变快与变好, 降低运行成本与系统经济化也成为重要考量标准。因为我们必须要小心货币带来的风险。

## 4 总结与展望

对于原生云数据库系统, 效率和成本效益是密切相关的。Crystal 是一个智能缓存存储系统, 与计算并行, 可以通过数据源连接器客户端被任何未修改的数据库使用。Crystal 在语义数据区域上操作, 并不断调整本地缓存的内容以获得**最大收益**。用于云中分析查询处理的直观模型也应运而生, 该模型可以估计特定硬件实例上工作负载的运行时间和成本。该模型最重要的应用程序是制定数据驱动的架构设计决策、发现性能瓶颈, 以及朝着云中**成本最优**的查询处理目标发展数据管理系统。SEAGULL 虽不针对数据库进行操作, 但在用户数量大量增加时, 采用 Data Science 来预测用户负载、优化资源分配, 从而控制**成本**。

并且, 针对最优查询的经济方法可以应用于分析数据库查询处理工作负载之外的其他领域。例如, 机器学习模型的训练可以使用类似的方法进行优化, 该方法使用的是一个专注于浮点计算的模型。在其他领域探索基于成本的方法将是未来工作的一个有趣的途径。

**致谢** 感谢施展老师与童薇老师为我们讲授数据中心技术这门课程, 感谢课程提供的学术会议模拟机会, 让自己得到了锻炼。

## 参考文献

- [1] Leis V, Kuschewski M. Towards cost-optimal query processing in the cloud[J]. Proceedings of the VLDB Endowment, 2021, 14(9): 1606-1612.
- [2] Durner D, Chandramouli B, Li Y. Crystal: a unified cache storage system for analytical databases[J]. Proceedings of the VLDB Endowment, 2021, 14(11): 2432-2444.
- [3] Poppe O, Amunike T, Banda D, et al. Seagull: An infrastructure for load prediction and optimized resource allocation[J]. arXiv preprint arXiv:2009.12922, 202