華中科技大學

本科生毕业设计（论文）开题报告

题　　目：面向大规模并行计算系统和网络的动态自适应编码传输方案研究

院　　系　计算机科学与技术

专业班级　本硕博 2001 班

姓　　名　李茗畦

学　　号　U202015630

指导教师　胡燏翀

2024 年 2 月 26 日

开题报告填写要求

一、 开题报告主要内容：

1. 课题来源、目的及意义；

2. 国内外研究现况及发展趋势；

3. 课题研究的内容和技术方案；

4. 可行性与风险分析；

5. 课题研究进度安排；

6. 主要参考文献。

二、 报告内容用小四号宋体字编辑，采用 A4 号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。

三、 指导教师评语、教研室（系、所）或开题报告答辩小组审核意见用蓝、黑钢笔手写或小四号宋体字编辑，签名必须手写。

四、 理、工、医类要求字数在 3000 字左右，文、管类要求字数在 2000 字左右。

五、 开题报告应在第八学期第二周之前完成。

1 课题来源、目的及意义

1.1 课题来源

本课题来自国家重点研发计划重点专项项目《大规模并行计算系统的可靠性编码理论和技术研究》。

1.2 课题目的及意义

大规模并行计算系统由许多计算节点组成，这些节点同时进行计算任务以实现高性能和高吞吐量的计算。系统中的计算节点通过网络互连，彼此之间可以进行通信和协作，并行地执行计算任务，每个节点负责处理一部分数据。常见的并行计算框架有 MapReduce 等。

大规模并行计算系统会出现故障频发现象，导致系统的网络联通度降低，出现链路拥塞，降低系统的网络传输性能，从而导致系统的计算效率下降。并且这一现象是动态变化的。因此，我们需要设计一个方案，可以最大化地利用系统中现有的网络资源和计算资源，加快分布式并行计算的效率，同时可以自适应地应对动态变化的网络和节点负载情况。

2 国内外研究现况及发展趋势

2.1 并行计算系统中关于编码的研究现状

CDC（coded distributed computing）[1]利用 MapReduce[2]中计算开销和通信开销的权衡，通过在 Map 阶段将计算开销增加 r 倍（即在 r 个精心选择的节点上执行每个 Map 函数）可以创造出编码机会实现对 MapReduce 中间结果的多播，将 Shuffle 阶段的通信开销也减少 r 倍。

CDC 方案和 MDS 编码结合以解决节点滞后的问题，实现了计算延迟和通信负载的权衡[3]。对于多阶段的计算任务，将其组织为一个 DAG 图，并根据 DAG 中每个节点的计算负载各自应用 CDC 方案，以最小化通信开销[3]。

编码计算(coded computation)以类似纠删码的方式在分布式计算中引入冗余任务，通过解码得到计算结果，避免少数滞后节点的影响，使矩阵乘法快 O(logn)倍(n 为节点数)[4]。编码洗牌(coded shuffling)在多轮洗牌的间隔中，选择性地保留

节点缓存中的冗余数据，使得节点在下一轮洗牌时可以通过解码的方式获得本轮的全部数据[4]。这种洗牌方式使通信开销减少 O(γ (n))倍(γ 为将 n 个消息单独发送与广播给 n 个节点的开销之比)。

对计算数据本身引入冗余数据，可以避免滞后节点的影响[5]。这种方法可以应用在梯度下降，L-BFGS 等批量算法中，无需修改具体的计算过程，可实现对原始问题近似解的样本路径的线性收敛。

Heterogeneous Coded Matrix Multiplication 算法（HCMM）同样通过引入冗余数据来加快矩阵乘法的运算[6]。并且，根据节点运行时间的概率分布情况进行负载均衡。

2.2 并行计算系统中关于负载均衡的研究现状

自适应，分层次的任务调度方案 AHS（An Adaptive And Hierarchical Task Scheduling Scheme）使用任务窃取和任务共享来实现负载均衡，将任务调度分为节点内的调度和节点间的调度两层来减少任务迁移带来的通信开销[8]。

大规模的数据集往往具有倾斜性，导致 MapReduce 系统的节点的负载不均衡。估计节点的运行成本是实现负载均衡算法的基础，这需要从 Mappers 中收集信息，较多的信息会增加成本估计的复杂度。TopCluster 在常数时间内构建全局直方图用于估计分区成本，并将估计误差限制在一个阈值之内[9]。

Map-Balance-Reduce（MBR）编程模型在 MapReduce 编程模型的 Map 和 Reduce 阶段添加了 Balance 阶段，对于负载较重的 Reduce 节点，根据 K-hash 将这些任务平均分为多个 Reduce 节点，从而避免 Reduce 阶段的负载不均匀[10]。

Mapper 通过分布式元数据存储(DMDS)进行通信发布各自的状态信息，了解全局的工作状态，并基于此技术设计了自适应的 Mapper，Combiner，取样以及分区函数[11]。

Network-Aware Shuffle Scheduling Algorithm (NAS)利用数据中心可预测的周期性网络状态，在网络空闲的时间段内安排 shuffle 任务，减少 shuffle 阶段持续的时间[12]。

ShuffleWatcher 根据网络负载情况延迟作业的 Shuffle 阶段，在网络较拥塞时降低作业内部的 Map-Shuffle 的并行性缓解网络拥塞[13]。

2.3 关于网络拥塞的研究现状

面向性能的拥塞控制 PCC(Performance-oriented Congestion Control)基于在线学习算法，通过观察每个发送方的不同行为与实际性能之间的关系动态调整每个发送方的行为，提供了相比 TCP 适应性和灵活性更好的拥塞控制策略[14]。

PCC Vivace 对 PCC 的效用函数和学习效率控制算法进行了调整，采用了严格凸的效用函数，并根据计算出来的效用值动态调整发送速率的步长，保证多个发送方都可以更快的收敛最优发送速率[15]。

一种基于强化学习的拥塞控制框架 Custard(CUSTomized And Robust Decision)进一步扩展了 PCC[14]，采用深度强化学习来生成策略，将观测到的指标(丢包率，平均延迟等)映射为发送速率的选择[16]。Custard 采用了一个三个隐藏层，每层 32 个神经元的神经网络来完成这个生成策略。可以通过设置不同的奖励函数来让网路达到需要的性能要求。

3 研究内容与技术方案

3.1 课题研究内容

本课题主要面向大规模并行计算系统。由于大规模并行计算系统的网络中共常常出现故障，导致链路拥塞，进而降低整个系统的计算效率，因此我们需要一个方案应对动态变化的网络负载情况。本课题旨在为大规模并行计算系统设计一个高效的网络传输方案，以自适应的应对动态变化的网络拥塞情况，并且可以处理节点负载不均衡的情况。

3.2 课题预期目标

本课题的主要目标是：

(1) 了解大规模并行计算系统的基础知识，以及基于网络编码的高效网络传输技术；

(2) 实现一套并行计算的原型系统；

(3) 设计并实现一个面向大规模并行计算系统的动态自适应网络传输算法；

(4) 在实现的原型系统上对方案的性能进行测试。

3.3 技术路线与方案

3.3.1 关键技术

（1）网络编码技术。CDC 方案通过网络编码实现中间结果的多播，从而减少通信开销，这一技术可应用在避免并行计算系统中的链路拥塞中；

（2）基于在线学习（或强化学习），动态调整 CDC 的参数避免链路拥塞

（3）基于 CDC 方案以及网络和节点负载感知的任务迁移

3.3.2 总体方案

一个基本的 MapReduce 并行计算框架如图 1 所示。一个 MapReduce 任务的执行被划分为 Map 和 Reduce 两个阶段，由 master 节点进行调度。每个 worker 节点根据输入文件执行 Map 任务生成中间结果保存在节点本地。当所有 worker 节点的 Map 任务执行完毕后，worker 节点彼此交换 Reduce 任务需要的中间结果。最后 worker 节点根据其本地的中间结果和 Shuffle 阶段收到的中间结果，执行 Reduce 任务生成最终的输出结果。在整个 MapReduce 计算期间，主要的计算开销集中在 Map 和 Reduce 阶段，主要的通信开销集中在 Shuffle 阶段。当由于网络故障出现链路拥塞时，导致部分中间结果的传输时间增加，从而降低 Shuffle 阶段的整体执行效率。

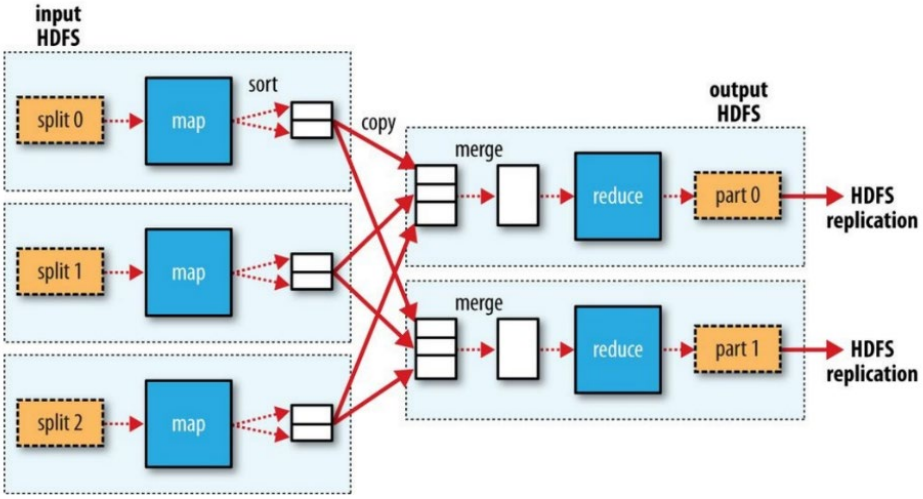

图 1 MapReduce 并行计算框架示意图

本课题的总体架构的设计思路如图 2 所示，是基于 MapReduce 框架设计的。节点负载和网络负载感知模块负责收集 worker 节点的负载情况(正在执行以及等

待执行的任务数)以及网络的负载情况(网络带宽，丢包率，传输延迟等)，并将这些信息反馈给任务调度模块，任务调度模块将根据这些信息分析系统的节点负载均衡情况以及网络拥塞程度，以此调整任务调度策略。

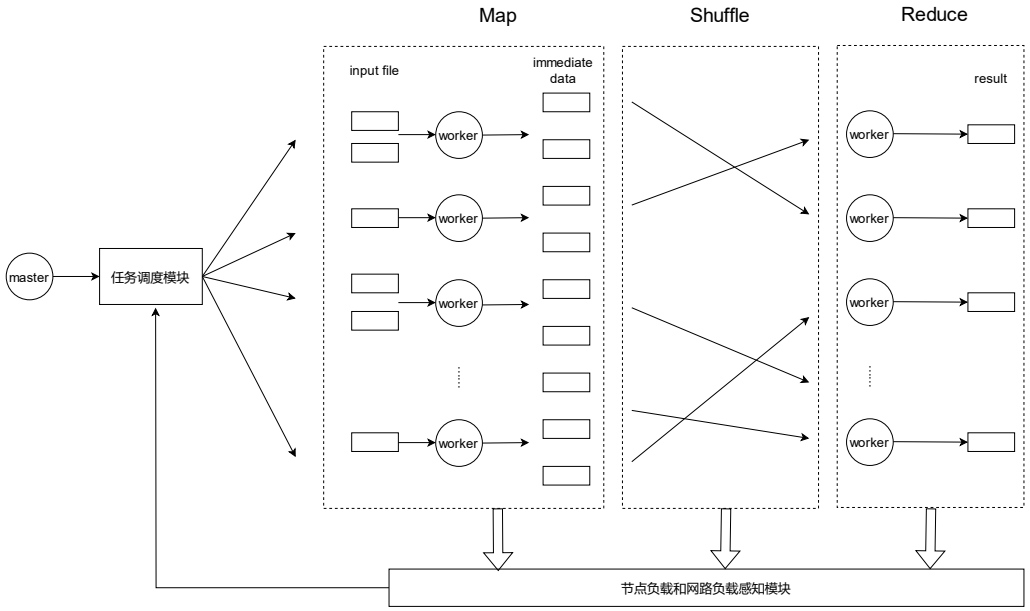

图 2 总体架构图

任务调度模块负责根据节点负载和网络负载，调整自己的任务调度策略。调正分为以下两个方面。第一，根据网络负载和节点负载情况动态调整 CDC 方案的参数。通过在系统中执行一些微实验在线观察不同行为如适当增加 CDC 的 r 值或降低 r 值对一个 MapReduce 作业的执行效率和系统节点以及网络负载情况的影响，向性能更优的方向上对系统进行调整。具体的流程如图 3 所示。

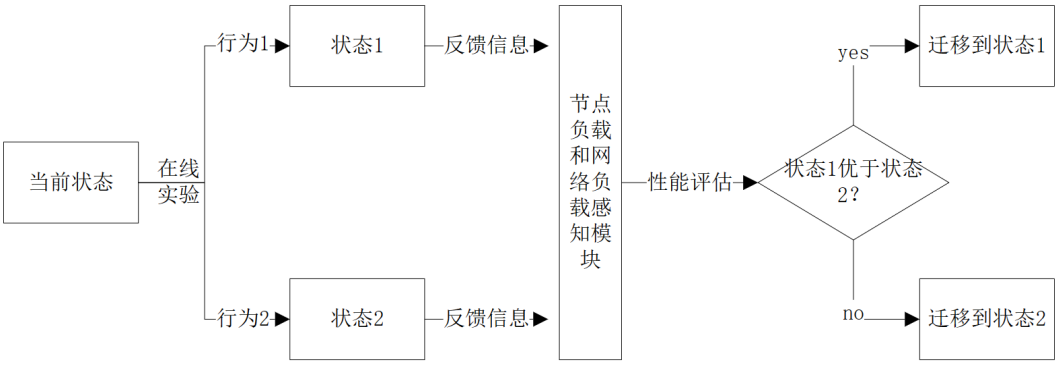

图 3 动态调整 CDC 方案参数的流程

第二，当节点负载和网路负载感知模块观察到某个节点的计算负载或者网络负载比较重时，便将该节点上的一些任务迁移到比较空闲的节点上执行。空闲的

节点同样由负载感知模块进行选择。这一部分会引入节点之间的通信开销。我们由于 CDC 方案在节点中生成了一些冗余的中间结果，因此我们在选择迁移到的节点时便可以优先选择具有要迁移的任务要用到的较多的中间结果的节点。这一部分的示意图如图 4 所示。

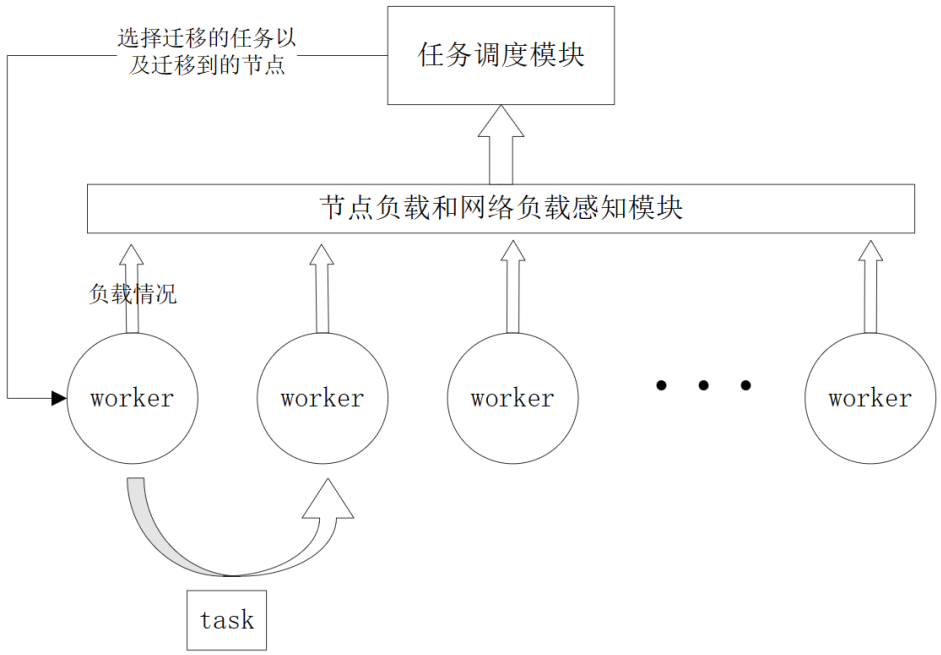

图 4 任务迁移示意图

4 课题研究进度安排

本课题的大致的进度安排如表 1 所示。

表 1 课题研究进度安排表

月份	工作任务
2024 年 1 月	搜索并阅读国内外并行计算领域的相关文献
~2024 年 2 月	了解该领域目前研究现状和现有的优化手段
2024 年 3 月	完成文献翻译和开题报告
~2024 年 4 月	实现课题的设计方案
2024 年 5 月	编写测试部分代码对系统性能进行测试，分析性能瓶颈并调优
2024 年 6 月	完成毕业论文，准备最终答辩

5 主要参考文献

[1] Li S, Maddah-Ali M A, Yu Q, et al. A fundamental tradeoff between computation

and communication in distributed computing[J]. IEEE Transactions on Information Theory, 2017, 64(1): 109-128.

[2] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[3] Li S, Maddah-Ali M A, Avestimehr A S. Coded distributed computing: Straggling servers and multistage dataflows[C]//2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2016: 164-171.

[4] Lee K, Lam M, Pedarsani R, et al. Speeding up distributed machine learning using codes[J]. IEEE Transactions on Information Theory, 2017, 64(3): 1514-1529.

[5] Karakus C, Sun Y, Diggavi S, et al. Straggler mitigation in distributed optimization through data encoding[J]. Advances in Neural Information Processing Systems, 2017, 30.

[6] Reisizadeh A, Prakash S, Pedarsani R, et al. Coded computation over heterogeneous clusters[J]. IEEE Transactions on Information Theory, 2019, 65(7): 4227-4242.

[7] Ezzeldin Y H, Karmoose M, Fragouli C. Communication vs distributed computation: An alternative trade-off curve[C]//2017 IEEE Information Theory Workshop (ITW). IEEE, 2017: 279-283.

[8] Wang Y, Zhang Y, Su Y, et al. An adaptive and hierarchical task scheduling scheme for multi-core clusters[J]. Parallel computing, 2014, 40(10): 611-627.

[9] Gufler B, Augsten N, Reiser A, et al. Load balancing in mapreduce based on scalable cardinality estimates[C]//2012 IEEE 28th International Conference on Data Engineering. IEEE, 2012: 522-533.

[10] Li J, Liu Y, Pan J, et al. Map-Balance-Reduce: An improved parallel programming model for load balancing of MapReduce[J]. Future Generation Computer Systems, 2020, 105: 993-1001.

[11] Vernica R, Balmin A, Beyer K S, et al. Adaptive MapReduce using situation-aware mappers[C]//Proceedings of the 15th International Conference on Extending

Database Technology. 2012: 420-431.

[12] Fan Y, Liu W, Guo D, et al. Shuffle Scheduling for MapReduce Jobs Based on Periodic Network Status[J]. IEEE/ACM Transactions on Networking, 2020, 28(4): 1832-1844.

[13] Ahmad F, Chakradhar S T, Raghunathan A, et al. {ShuffleWatcher}: Shuffle-aware scheduling in multi-tenant {MapReduce} clusters[C]//2014 USENIX Annual Technical Conference (USENIX ATC 14). 2014: 1-13.

[14] Dong M, Li Q, Zarchy D, et al. {PCC}: Re-architecting congestion control for consistent high performance[C]//12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). 2015: 395-408.

[15] Dong M, Meng T, Zarchy D, et al. {PCC} Vivace:{Online-Learning} Congestion Control[C]//15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). 2018: 343-356.

[16] Jay N, Rotman N H, Godfrey P, et al. Internet congestion control via deep reinforcement learning[J]. arXiv preprint arXiv:1810.03259, 2018.

华中科技大学本科生毕业设计（论文）开题报告评审表

姓名	李茗畦	学号	U202015630	指导教师	胡燏翀
院（系）专业	计算机科学与技术学院本硕博2001班				

指导教师评语

1. 学生前期表现情况。
2. 是否具备开始设计（论文）条件？是否同意开始设计（论文）？
3. 不足及建议。

表现良好，建议开始设计论文。

评　分：90　　指导教师（签名）：

2024年3月8日

教研室（系、所）或开题报告答辩小组审核意见

同意开题

教研室（系、所）或开题报告答辩小组负责人（签名）：

2024年3月8日