

面向期限感知分布式矩阵相乘的高效存储方案

赵永柱¹, 黎卫东^{2*}, 唐 斌², 梅 峰³, 卢文达³

(1. 国网陕西省电力公司信息通信公司, 西安 710065; 2. 计算机软件新技术国家重点实验室(南京大学), 南京 210023;

3. 国网浙江省电力有限公司, 杭州 310007)

(* 通信作者电子邮箱 shanzhonghai@foxmail.com)

摘要: 分布式矩阵相乘是众多分布式机器学习、科学计算等应用中的关键操作,但其性能会受到系统中常见的落后节点的严重影响。最近研究者提出了基于喷泉码的编码矩阵相乘方法,能够充分利用落后节点的部分计算结果,从而大幅度减轻落后节点问题,但忽略了工作节点的存储开销。在考虑存储开销与计算完成时间之间的权衡关系的基础上,首先提出了面向异构工作节点的计算期限感知的存储优化问题;然后进一步通过理论分析,提出了基于期望近似的解决思路,并通过松弛将问题转化为凸优化问题以方便高效求解。仿真实验表明,在保证较大的任务成功率的情况下,所提方案的存储开销会随着任务期限的放宽迅速下降,并且该方案能够更大幅度降低编码带来的存储开销。也就是说,所提方案能够在保障整体计算在期限内大概率完成的前提下,大幅度降低总体的额外存储负载。

关键词: 分布式计算; 矩阵相乘; 落后节点; 凸优化; 期限感知

中图分类号: TP311 **文献标志码:** A

Efficient storage scheme for deadline aware distributed matrix multiplication

ZHAO Yongzhu¹, LI Weidong^{2*}, TANG Bin², MEI Feng³, LU Wenda³

(1. State Grid Shaanxi Electric Power Information & Telecommunication Company Limited, Xi'an Shaanxi 710065, China;

2. State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing Jiangsu 210023, China;

3. State Grid Zhejiang Electric Power Company Limited, Hangzhou Zhejiang 310007, China)

Abstract: Distributed matrix multiplication is a fundamental operation in many distributed machine learning and scientific computing applications, but its performance is greatly influenced by the stragglers commonly existed in the systems. Recently, researchers have proposed a fountain code based coded matrix multiplication method, which can effectively mitigate the effect of stragglers by fully exploiting the partial results of stragglers. However, it lacks the consideration of the storage cost of worker nodes. By considering the tradeoff relationship between the storage cost and the finish time of computation, the computational deadline-aware storage optimization problem for heterogeneous worker nodes was proposed firstly. Then, through the theoretical analysis, the solution based on expectation approximation was presented, and the problem was transformed into a convex optimization problem by relaxation for efficient solution. Simulation results show that in the case of ensuring a large task success rate, the storage overhead of the proposed scheme will rapidly decrease as the task duration is relaxed, and the scheme can greatly reduce the storage overhead brought by encoding. In other words, the proposed scheme can significantly reduce the extra storage overhead while guaranteeing that the whole computation can be finished before the deadline with high probability.

Key words: distributed computation; matrix multiplication; straggler; convex optimization; deadline aware

0 引言

随着互联网、云计算、物联网的迅猛发展,无处不在的移动设备、传感器持续产生数据,数以亿计用户的互联网服务时时刻刻产生巨量的交互,使得数据呈现出爆炸式增长趋势。大规模的数据资源为大规模的机器学习和数据挖掘应用提供了物质基础,因此近年来涌现了很多大规模的机器学习模型。这些模型在语音处理、图像处理、自然语言处理、人机对弈等

领域表现出非凡的问题处理能力;但是大规模机器学习的训练对计算能力和存储容量都提出了较高的要求,单机训练和存储开销太高,利用分布式计算技术来完成训练任务已成常态。

矩阵-向量相乘,简称矩阵相乘,即计算 $y = Ax$ (其中, A 为矩阵, x 为向量),是很多数值计算、机器学习算法中的核心计算,比如求解偏微分方程^[1]、神经网络的前向后向推进^[2]、计算 PageRank^[3]等。通过多台机器来分布式计算矩阵相乘时,通

收稿日期: 2019-07-31; 修回日期: 2019-10-14; 录用日期: 2019-10-16。

基金项目: 国家自然科学基金资助项目(61832005, 61872171); 国家电网公司科技项目(52110418001M)。

作者简介: 赵永柱(1974—),男,陕西乾县人,高级工程师,硕士,主要研究方向: 电力信息系统、数据中心; 黎卫东(1996—),男,江西赣州人,硕士研究生,主要研究方向: 分布式计算; 唐斌(1986—),男,江苏东台人,助理研究员,博士,CCF会员,主要研究方向: 分布式计算、编码理论; 梅峰(1977—),男,浙江湖州人,高级工程师,硕士,主要研究方向: 电力信息系统、大数据; 卢文达(1989—),男,吉林松原人,助理工程师,硕士,主要研究方向: 数据挖掘、云计算。

常将矩阵 A 进行分割后分配到系统中的各个工作节点 (worker node), 每个节点将子矩阵与 x 进行相乘后将结果交给主节点 (master node), 由主节点整合出最终的计算结果。在这种方式下, 主节点需要等到所有工作节点均返回计算结果后方可完成整体计算任务。

然而, 大规模分布式系统由于资源竞争、负载均衡、网络变化等诸多原因, 工作节点在执行计算任务时, 其完成时间经常难以预测且表现出较大波动, 从而会有部分计算节点计算严重滞缓, 其计算完成时间能超出平均水平 5~10 倍, 从而严重拖累整体计算。这样的节点通常被称为落后节点 (straggler), 产生的问题被称为落后者问题^[4,5]。

近年来, 研究者提出采用编码的方法能够有效应对上述落后者问题。Lee 等首先提出基于 MDS (Maximum Distance Separable) 码的编码方案^[6-7], 即将矩阵划分成 k 份, 并应用码率为 k/n 的 MDS 码, 将其编码成 n 份 (n 为工作节点个数), 每个工作节点存储其中一份并完成计算。这种方法可以容忍 $n - k$ 个落后节点。这种方法将落后节点等价于传统通信中的擦除 (erasure), 无法利用落后节点上的部分计算结果, 从而导致性能欠佳。为了能够利用落后节点, 研究者从不同的角度提出了一些可行方案^[7-10]。其中, Mallick 等提出了基于喷泉码的编码方案^[10-12], 利用喷泉码的无码率特性, 可以将 k 份子矩阵编码成任意多的编码矩阵, 并将这些编码矩阵预先存储到工作节点上去, 而工作节点则可以逐步计算并返回每个编码矩阵与向量 x 相乘的结果; 与此同时, 主节点只需要接收到的部分计算结果数略多于 k 即可恢复出完整计算结果。与其他方法相比, 这种基于喷泉码的方式能够充分利用每个工作节点的计算能力, 且具有线性的编译码复杂度。

Mallick 等的工作假设每个工作节点存储的分片数充分多, 从而保障在整体计算完成前, 每个工作节点均持续进行计算^[10], 但这种方式将消耗过多的存储资源, 从而导致计算成本过高。譬如, 用户以租用云计算平台的方式进行分布式计算, 成本开销与存储开销密切相关。另一方面, 如果工作节点存储的编码矩阵数目过小, 则可能导致工作节点因过早完成分配给它的计算任务而处于空闲状态, 从而延长整体计算完成时间。因此, 存储开销与整体计算完成时间存在一定的权衡关系。

基于上述分析, 本文首先对如何在保障给定期限内大概率完成计算任务的前提下优化工作节点的存储开销问题进行了建模。该问题求解面临的难点有两方面: 一是工作节点的耗时具有随机性; 二是工作节点的异构性, 即其耗时相关的参数不同。通过理论分析发现, 在通常情况下, 一定时间内主节点接收到的计算结果个数以较大概率集中在其期望值左右, 从而提出了基于期望近似的解决思路, 将原问题约束条件进行了转化。由于转化后的问题仍是整数规划问题, 进一步地考虑了松弛, 并证明松弛后的优化问题为凸优化问题, 从而可以高效求解, 并将其最优解通过简单的取整转化为未松弛问题的可行解。仿真实验结果表明, 所提方法能够在保障整体计算在期限内大概率完成的前提下, 大幅度降低总体的额外存储开销。

1 问题建模

1.1 系统模型

考虑在一典型分布式计算环境中计算 $y = Ax$, 其中, $A \in \mathbb{R}^{r \times m}$ 是大小为 $r \times m$ 的矩阵, $x \in \mathbb{R}^m$ 为 m 维的向量。该计算环境由一个主节点 (master node) 及 n 个工作节点构成。考虑采用喷泉码的方法有效应对落后者问题。具体而言, 如图 1 所示, 首先将矩阵 A 按行分割成 k 个大小相同的子矩阵 ($a_1, a_2, \dots, a_k$) (如果 r 不能被 k 整除, 可以通过补 0 完成), 每个子矩阵包含 r/k 行, 并利用喷泉码 (如 LT 码^[13]、Raptor 码^[14]) 对这 k 个子矩阵进行编码, 可以生成一系列 (数量可以无穷多) 的编码子矩阵, 其大小与这 k 个子矩阵保持一致。

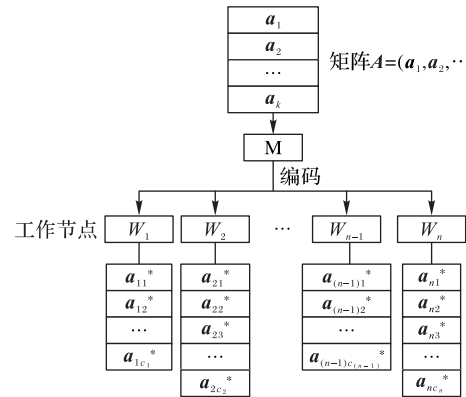


图 1 矩阵预编码和部署

Fig. 1 Matrix pre-coding and allocation

令 c_i 为第 i 个工作节点放置的编码矩阵个数, 并记这些编码矩阵为 $(a_{1i}^*, a_{2i}^*, \dots, a_{c_i i}^*)$, 则第 i 个工作节点按序计算 $(a_{1i}^* x, a_{2i}^* x, \dots, a_{c_i i}^* x)$, 每算完一个就将结果发送给主节点。当主节点从所有工作节点处接收到 $(1 + \varepsilon)k$ 个这样的结果时, 可以通过喷泉码的 Peeling 算法^[10]进行解码, 从而能够以非常接近 1 的概率恢复出 $(a_1 x, a_2 x, \dots, a_k x)$, 其中 $\varepsilon > 0$ 是一个非常小的常数, 称之为解码条件。整个工作流程如图 2 所示。

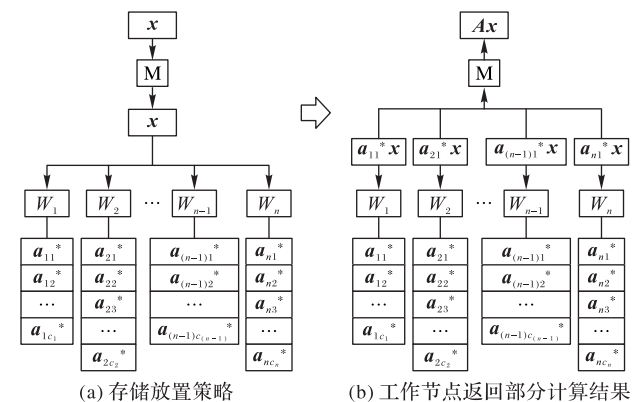


图 2 编码矩阵相乘的流程

Fig. 2 Flowchart of coded matrix multiplication

1.2 耗时模型

为刻画任务完成时间, 本文采用与文献^[10]相同的耗时模型。每个工作节点 i 的耗时包含两部分: 一是启动时延 X_i , 它服从参数为 μ_i 的指数分布, 即 $X_i \sim \exp(\mu_i)$; 二是每计算并返

回一个结果给主节点所需要的时间为常数 τ_i 。因此,工作节点返回 R_i 个计算结果所花费的时间为:

$$T_i = X_i + \tau_i R_i; i = 1, 2, \dots, n \quad (1)$$

其中 $R_i < c_i$ 。此外, X_i 之间是独立的。本文考虑了工作节点之间的异构性,即对于不同的 i, μ_i 和 τ_i 的取值可能不同,如图3所示。

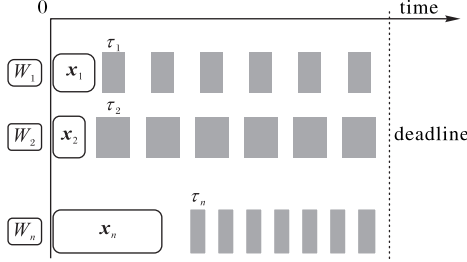


图3 耗时模型与任务执行示意图

Fig. 3 Schematic diagram of time consumption model and task execution

1.3 期限感知的存储优化

考虑上述计算任务存在期限 t^* , 即上述计算任务能够以较大概率在 t^* 时间内完成。一种方式是每个工作节点上放置充分多的编码子矩阵, 保证每个工作节点在 t^* 时间内均保持计算状态, 这能使得计算完成时间最短; 但这种方法带来的存储开销将非常大。另一方面, 如果在工作节点上部署过少的编码子矩阵, 则可能导致工作节点过早进入闲置状态, 从而延长计算完成时间。基于上述考虑, 本文提出期限感知的存储优化问题, 致力于优化工作节点的存储开销, 同时保障计算能以较大概率在期限 t^* 内完成。

令 $R_i(t^*)$ 表示工作节点 W_i 在 t^* 时刻返回的计算结果数, 它取决于启动时延 X_i , 同时受限于编码子矩阵的存储个数 c_i , 因此为取值不超过 c_i 的随机变量。因此, 根据解码条件, 计算任务能在期限 t^* 完成的充分必要条件为:

$$\sum_{i=1}^n R_i(t^*) \geq (1 + \varepsilon)k \quad (2)$$

基于此, 本文将期限感知的存储优化问题(记为问题P)建模成:

$$\begin{aligned} \min \quad & \sum_{i=1}^n c_i \\ \text{s.t.} \quad & \Pr\left(\sum_{i=1}^n R_i(t^*) \geq k(1 + \varepsilon)\right) \geq 1 - O(1/n) \\ & c_i > 0, c_i \text{ 为整数} \end{aligned} \quad (3)$$

其中, 限制条件表示计算任务在 t^* 时间内无法完成的概率不超过 $O(1/n)$, 即是 $1/n$ 的无穷小量。

2 存储优化方案

在上述优化建模的限制条件中, 由于 X_i 之间的独立性, $R_i(t^*)$ 之间也是独立的, 因此 $\sum_{i=1}^n R_i(t^*)$ 的分布可以通过卷积公式进行计算, 但其结果会相当复杂, 且包含待优化变量, 难以帮助问题求解。针对此问题, 本文提出了基于期望近似的解决思路。

为方便求解, 首先引入如下一些自然的假设:

1) 异构性受限假设: 存在某常数 C , 使得对于任意的 i 和 j 均有 $\tau_i/\tau_j \leq C$, 即不同工作节点的计算速度之间的比例受限。

2) 期限假设: 由于 $R_i(t^*) \leq \frac{t^*}{\tau_i}$, $\sum_{i=1}^n \frac{t^*}{\tau_i} \geq \sum_{i=1}^n R_i(t^*) \geq k(1 + \varepsilon)$,

假设 $\sum_{i=1}^n \frac{t^*}{\tau_i} = \Theta(k)$, 这是因为考虑过大的 t^* 并没有太多实际意义。

基于上述假设可以发现, 存在常数 C' 使得对于充分大的 k ,

$$0 \leq R_i(t^*) \leq \frac{t^*}{\tau_i} \leq C' \cdot \frac{k}{n}$$

此外, 由于 $R_i(t^*)$ 之间是相互独立的, 因此, 可以应用 Hoeffding 不等式^[15]得到:

$$\Pr\left(\left|\sum_{i=1}^n R_i(t^*) - \mathbb{E}\left[\sum_{i=1}^n R_i(t^*)\right]\right| \geq \varepsilon'k\right) \leq 2e^{-\frac{2(\varepsilon'k)^2}{n \cdot (C' \cdot \frac{k}{n})^2}} = 2e^{-\frac{2\varepsilon'^2 n}{C'^2}} = O\left(\frac{1}{n}\right) \quad (4)$$

其中 $\varepsilon' > 0$ 可以是任意小的常数。这意味着, $\sum_{i=1}^n R_i(t^*)$ 的实际值以大概率集中在其期望值 $\mathbb{E}\left[\sum_{i=1}^n R_i(t^*)\right]$ 左右。因此, 问题建模中的限制条件可以转化为 $\mathbb{E}\left[\sum_{i=1}^n R_i(t^*)\right] \geq k(1 + \varepsilon')$, 其中

$$\varepsilon' = \varepsilon + \varepsilon'.$$

进一步, 计算 $\mathbb{E}\left[\sum_{i=1}^n R_i(t^*)\right]$ 如下:

$$\begin{aligned} \mathbb{E}[R_i(t^*)] &= \sum_{j=1}^{c_i} \Pr(R_i(t^*) = j) \cdot j = \\ &= \sum_{j=1}^{c_i-1} \Pr\left(\left\lfloor \frac{t^* - X_i}{\tau_i} \right\rfloor = j\right) \cdot j + \Pr\left(\left\lfloor \frac{t^* - X_i}{\tau_i} \right\rfloor \geq c_i\right) \cdot c_i = \\ &= \sum_{j=1}^{c_i-1} \Pr\left(j \leq \frac{t^* - X_i}{\tau_i} < j+1\right) \cdot j + \Pr\left(\frac{t^* - X_i}{\tau_i} \geq c_i\right) \cdot c_i = \\ &= \sum_{j=1}^{c_i-1} \Pr(t^* - (j+1)\tau_i \leq X_i < t^* - j\tau_i) \cdot j + \Pr(X_i \geq t^* - c_i\tau_i) \cdot c_i = \\ &= \sum_{j=1}^{c_i-1} (F(t^* - j\tau_i) - F(t^* - (j+1)\tau_i)) \cdot j + (1 - F(t^* - c_i\tau_i)) \cdot c_i = \\ &= \sum_{j=1}^{c_i-1} \left(1 - e^{-\mu_i(t^* - j\tau_i)} - \left(1 - e^{-\mu_i(t^* - (j+1)\tau_i)}\right)\right) \cdot j + e^{-\mu_i(t^* - c_i\tau_i)} \cdot c_i = \\ &= c_i - \frac{e^{\mu_i\tau_i} - \mu_i t^*}{e^{\mu_i\tau_i} - 1} \left(e^{\mu_i\tau_i c_i} - 1\right) \end{aligned}$$

其中, 函数 $F(t) = 1 - e^{-\mu_i t}$ 为 X_i 的分布函数。根据期望的线性性质, 有

$$\mathbb{E}\left[\sum_{i=1}^n R_i(t^*)\right] = \sum_{i=1}^n \left(c_i - \frac{e^{\mu_i\tau_i} - \mu_i t^*}{e^{\mu_i\tau_i} - 1} (e^{\mu_i\tau_i c_i} - 1)\right) \quad (5)$$

至此, 可以将期限感知的存储优化问题转化为如下优化问题(记为问题P2):

$$\min \sum_{i=1}^n c_i \quad (6)$$

$$\text{s.t. } \sum_{i=1}^n \left(c_i - \frac{e^{\mu_i \tau_i - \mu_i t^*}}{e^{\mu_i \tau_i - 1}} (e^{\mu_i \tau_i c_i} - 1) \right) \geq k(1 + \varepsilon')$$

$c_i > 0, c_i$ 为整数

由于上式是整数规划问题,难以直接求解。对 c_i 为整数的条件进行松弛,即考虑 c_i 为非负实数的情形(记为问题 P3):

$$\min g(c_1, c_2, \dots, c_n) \quad (7)$$

$$\text{s.t. } \sum_{i=1}^n \left(c_i - \frac{e^{\mu_i \tau_i - \mu_i t^*}}{e^{\mu_i \tau_i - 1}} (e^{\mu_i \tau_i c_i} - 1) \right) \geq k(1 + \varepsilon')$$

$c_i > 0$

$$\text{令 } f_i(c) = c - \frac{e^{\mu_i \tau_i - \mu_i t^*}}{e^{\mu_i \tau_i - 1}} (e^{\mu_i \tau_i c} - 1), \text{ 有}$$

$$f_i'(c) = 1 - \frac{e^{\mu_i \tau_i - \mu_i t^*}}{e^{\mu_i \tau_i - 1}} \mu_i \tau_i e^{\mu_i \tau_i c}$$

$$f_i''(c) = -\frac{e^{\mu_i \tau_i - \mu_i t^*}}{e^{\mu_i \tau_i - 1}} (\mu_i \tau_i)^2 e^{\mu_i \tau_i c} \leq 0$$

令 $g(c_1, c_2, \dots, c_n) = k(1 + \varepsilon') - \sum_{i=1}^n f_i(c_i)$ 不难发现,函数 $g(c_1, c_2, \dots, c_n)$ 的二阶偏导构成的黑塞矩阵(Hessian Matrix)为对角矩阵,且第 i 个对角元素值为 $-f_i''(c_i) \geq 0$; 因此,该黑塞矩阵为半正定矩阵,即函数 $g(c_1, c_2, \dots, c_n)$ 为下凸函数^[16]。此外, $\sum_{i=1}^n c_i$ 为线性函数; 因此上述松弛后的优化问题 P3 为凸优化问题,是多项式时间内可解的。可通过 Matlab CVX 软件包快速求解。记 P3 的最优解为 $c_1', c_2', \dots, c_n'$ 。

令 $c_i^* = \lceil c_i' \rceil$, 即对 c_i' 向上取整。下面证明 $c_1^*, c_2^*, \dots, c_n^*$ 为 P2 的可行解。注意到函数 $f_i(c)$ 表示当工作节点 i 存储 c 个编码子矩阵的情况下, t^* 时刻期望完成的计算结果数。显然,放置更多的编码子矩阵只会提升完成的计算结果数,因此, $f_i(c)$ 为单调递增函数,这意味着 $f_i(c_i') \leq f_i(c_i^*)$ 。从而,

$$g(c_1^*, c_2^*, \dots, c_n^*) - g(c_1', c_2', \dots, c_n') =$$

$$\sum_{i=1}^n f_i(c_i') - \sum_{i=1}^n f_i(c_i^*) \leq 0$$

故 $g(c_1^*, c_2^*, \dots, c_n^*) \leq g(c_1', c_2', \dots, c_n') \leq 0$, 其中第二个不等式是因为 $c_1', c_2', \dots, c_n'$ 为 P3 的可行解, 因此, $c_1^*, c_2^*, \dots, c_n^*$ 为 P2 的可行解。

注意到 P2 的最优目标值不会低于 P3 的最优目标值 $\sum_{i=1}^n c_i'$, 因此, P2 的可行解 $c_1^*, c_2^*, \dots, c_n^*$ 对应的目标函数值与最优值相比, 不会超过 n 。这说明, 找到的可行解 $c_1^*, c_2^*, \dots, c_n^*$ 是接近最优的。

需要补充说明的是, 在小规模分布式计算环境中, 当 n 较小, 且 ε' 取值较小时, 基于 Hoeffding 不等式建立的计算失败概率上限可能会较大。因此在实际应用时, 可以通过增加 P3 的限制条件中 ε'' 的取值去降低计算失败的概率。

3 仿真验证

对于本文提出的存储优化方案, 将从两个角度来分析验证: 一方面, 本文存储方案是期限感知的, 所以希望知道随着任务期限放宽, 在保证任务完成概率的前提下, 本文方案存储

开销的变化情况; 另一方面, 与其他方案进行对比, 以验证本文方案在存储优化方面的效果。

3.1 仿真环境

仿真实验在 Matlab R2018b 工具平台中进行, 凸优化求解工具是 CVX 工具包下的 SDPT3 求解器。仿真实验中, 有 20 个工作节点和 1 个主节点。工作节点的参数, 如任务的单位处理时间 τ_i 和异构与启动时延参数 μ_i , 通过均匀分布采样随机生成, 并满足 1.2 节中的假设。其中 τ_i 从 $[0.1, 1]$ 中随机生成, μ_i 从 $[0.0125, 0.125]$ 中随机生成。这些设定构成仿真实验的拓扑信息, 在一组实验过程中不会发生改变。此外, 假设数据矩阵被分成 1000 份, 即 k 取值为 1000。

3.2 存储开销变化趋势

根据系统的拓扑信息, 给定一个任务完成期限, 同时设定一个存储阈值 ε'' , 就可以通过所提方法求得一个存储分配方案, 并依据这个方案分配计算任务。计算任务分配成功后, 便能进行大量的仿真测试, 可以求得本文方案的任务完成成功率。与此同时, 可以仿真测得在每个节点上存储无限量的计算任务假设下任务完成的概率, 作为任务成功率的上限。可以调节存储阈值 ε'' , 使得本文的存储方案成功率接近任务成功率上限, 此时获得的存储方就是最终的优化方案。随机生成两组不同的系统参数, 针对一组不同的任务期限, 每个任务期进行 10^4 次仿真测试, 统计其任务成功率, 以及优化方案的存储开销(用存储的子矩阵数目表示)。

在两组随机生成的环境下进行了实验, 实验结果如图 4。可以看出, 在保证较大任务成功率(接近任务成功率上限, 即每个工作节点均存放充分多的编码子矩阵)的前提下, 随着任务期限的逐渐放宽, 所提方案引入的存储开销在迅速下降, 并逐渐趋向于原始子矩阵个数。根据存储开销的变化趋势, 结合实际情形, 本文的存储优化方案都可以有较好的收益; 对于对任务完成时间要求高的场景, 本文方案可以以不超过原始数据 2 倍存储的开销, 就达到和无限存储接近的任务完成时间和成功率; 面对对任务时间要求不高的场景, 本文方案只引入了输入矩阵数据量 0.2~0.3 倍的额外存储开销, 即可以大概率保证计算任务成功完成。同时, 对比图 4(a) 和 (b) 可以发现, 对于不同的系统配置, 本文方案的存储开销变化趋势都是一致的。

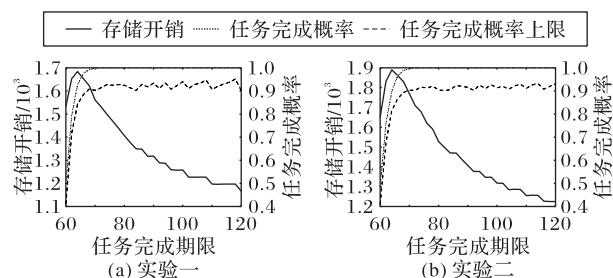


图 4 任务完全期限延长下存储开销变化趋势

Fig. 4 Trend of storage overhead with deadline increasing

3.3 对比实验

该部分实验中, 选取了另外两种预先放置策略, 并在给定任务期限前提下, 比较三种方案的存储开销变化对任务完成概率的影响。三种方案只在每个节点预先存储子矩阵数目方面有所不同, 其他方面则保持一致, 即均是采用无码率的喷泉

码编码,逐步返回计算内容。另外两种对比方案如下:

1) 均匀放置方案:每个工作节点放置等量的计算任务。这种方案忽视了工作节点之间的异构性。

2) 比例放置方案:按照参数 τ_i 即处理速度来按比例分配总的存储开销。这种方案只考虑了不同工作节点计算速度上面的区别,并未考虑启动时延方面的区别。

在一组随机生成的环境下进行了实验,并考虑两个不同的任务期限100及80。对于不同的存储开销,每个方案均仿真 10^4 次求得任务成功的概率,实验结果如图5所示。从图5中可以看到,随着存储开销的变大,三种方案的任务完成概率均是逐步上升的,其中所提存储优化方案任务完成概率上升速度最快,而比例放置策略则优于均匀放置策略。当任务大概率成功,比如概率取0.9或者0.95时,如图5所示,本文方案带来的冗余存储只有按比例放置方案的一半左右。此外,通过对比图5(a)和(b)可以发现,在任务完成期限较小时,本文存储优化方案与对比方案相比,性能优势更为明显。

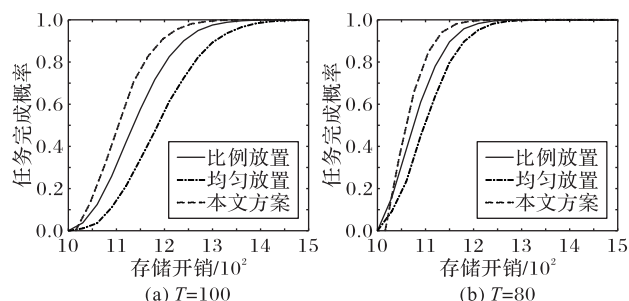


图5 三种存储方案下存储开销与任务完成概率对比

Fig. 5 Storage overheads and task success rates of three storage schemes

4 结语

在考虑存储开销与计算完成时间之间的权衡关系的基础上,本文对面向异构工作节点的计算期限感知的存储优化问题进行了建模,并基于理论分析,提出了基于期望近似的解决思路,进而通过松弛将问题转化为凸优化问题进行高效求解。仿真实验结果表明,本文存储方案与其他方案相比,能够大幅度降低编码带来的额外存储负载。由于很多实际应用中数据被建模成稀疏矩阵,因此未来还将进一步探讨面向稀疏矩阵-向量相乘的高效存储方法。

参考文献 (References)

- [1] MAZUMDER S. Numerical Methods for Partial Differential Equations[M]. Waltham, MA: Academic Press, 2014:145-151
- [2] DALLY W. High-performance hardware for machine learning [R/OL]. NIPS Tutorial, 2015 [2019-04-06]. <https://www.microsoft.com/en-us/research/video/tutorial-high-performance-hardware-for-machine-learning/?from=http%3A%2F%2Fresearch.microsoft.com%2Fapps%2Fvideo%2F%3D259571>.
- [3] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: bringing order to the Web, Technical Report 1999-66 [R]. Stanford: Stanford InfoLab, 1999.
- [4] ANANTHANARAYANAN G, KANDULA S, GREENBERG A G, et al. Reining in the outliers in map-reduce clusters using Mantri [C]// Proceedings of the 9th USENIX Symposium on Operating Sys-

tems Design and Implementation. Berkeley, CA: USENIX Association, 2010:265-278.

- [5] DEAN J, BARROSO L A. The tail at scale[J]. Communications of the ACM, 2013, 56(2):74-80.
- [6] 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.
- [7] KIANI S, FERDINAND N, DRAPER S C. Exploitation of stragglers in coded computation [C]// Proceedings of the 2018 IEEE International Symposium on Information Theory. Piscataway: IEEE, 2018: 1988-1992.
- [8] AMIRI M M, GUNDUZ D. Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air [EB/OL]. [2019-02-11]. <https://arxiv.org/pdf/1901.00844v1.pdf>.
- [9] DAS A B, TANG L, RAMAMOORTHY A. C³LES: codes for coded computation that leverage stragglers [C]// Proceedings of the 2018 IEEE Information Theory Workshop. Piscataway: IEEE, 2018: 1-5.
- [10] MALLICK A, CHAUDHARI M, JOSHI G. Fast and efficient distributed matrix-vector multiplication using rateless fountain codes [C]// Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2019: 8192-8196.
- [11] SEVERINSON A, AMAT A G I, ROSNES E. Block-diagonal and LT codes for distributed computing with straggling servers [J]. IEEE Transactions on Communications, 2019, 67 (3): 1739-1753.
- [12] WANG S, LIU J, SHROFF N. Coded sparse matrix multiplication [EB/OL]. [2019-02-11]. <https://arxiv.org/pdf/1802.03430.pdf>.
- [13] LUBY M. LT codes [C]// Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE, 2002: 271-280.
- [14] SHOKROLLAHI A, LUBY M. Raptor codes [J]. Foundations and Trends in Communications and Information Theory, 2011, 6 (3/4): 213-322.
- [15] MOTWANI R, RAGHAVAN P. Randomized Algorithms [M]. Cambridge: Cambridge University Press, 1995:76-78
- [16] BOYD S, VANDENBERGHE L. Convex Optimization [M]. Cambridge: Cambridge University Press, 2004:243-249

This work is partially supported by the National Natural Science Foundation of China (61832005, 61872171), the Science and Technology Project of State Grid Corporation of China (52110418001M).

ZHAO Yongzhu, born in 1974, M. S., senior engineer. His research interests include power information system, data center.

LI Weidong, born in 1996, M. S. candidate. His research interests include distributed computing.

TANG Bin, born in 1986, Ph. D., assistant research fellow. His research interests include distributed computing, coding theory.

MEI Feng, born in 1977, M. S., senior engineer. His research interests include power information system, big data.

LU Wenda, born in 1989, M. S., assistant engineer. His research interests include data mining, cloud computing.