

基于编码技术的仿真平台分布式计算方法

吴巨爱¹, 严雨欣¹, 彭沛^{1*}, 谢东亮², 张卉琳²

¹南京邮电大学 江苏南京

²国网电力科学研究院有限公司 江苏南京

【摘要】分布式计算是解决大规模电力系统仿真算力需求的有效方法。然而分布式计算容易受到少数问题节点性能下降所带来的负面影响,从而导致算法的计算效率严重下降。本文提出基于编码技术的仿真平台分布式计算方法,从而规避上述负面影响。首先,对构建了大规模电力系统仿真的计算任务模型;然后,通过理论分析确定了其中最优编码率的取值范围;最后,仿真分析结果表明,应用编码技术并采用合适的编码率,可有效提高大规模电力系统仿真中分布式算法的运行效率。

【关键词】电力系统;分布式计算;仿真;编码技术;性能分析

【基金项目】江苏省自然科学基金(BK20230369, BK20232026)

【收稿日期】2025年2月20日 **【出刊日期】**2025年3月18日 **【DOI】**10.12208/j.jjea.20250078

Coding in distributed computing for the simulation platform

Juwei Wu¹, Yuxin Yan¹, Pei Peng^{1*}, Dongliang Xie², Huilin Zhang²

¹Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu

²State Grid Electric Power Research Institute, Nanjing, Jiangsu

【Abstract】 Distributed computing is an effective method to provide the computing sources for large-scale electric power system simulations. However, the performance degradation of some computing nodes will negatively impact distributed computing of the entire system simulations, which leads to a serious decline in the computational efficiency of algorithms. This paper investigates the simulation platforms based on coding techniques to avoid the negative effects in the distributed computing system. First, we model the computing tasks of the large-scale electric power system simulations. Then, we theoretically analyze the range of the optimal coding rate. Finally, from the simulation results, we conclude that coding with the appropriate coding rate can effectively improve the computational efficiency of the distributed computing of the large-scale electric power system simulations.

【Keywords】 Electric power system; Distributed computing; Simulation; Coding techniques; Performance analysis

引言

随着信息-物理-社会元素耦合下的新型电力系统不断发展,大规模电力系统仿真依赖庞大的算力支撑^[1-2]。传统使用单个计算节点对数据进行处理方案已经无法满足日益增长的算力需求。分布式并行计算系统,例如谷歌的 TensorFlow 系统、脸书的 PyTorch 系统和华为的 MindSpore 系统等可以执行需要数千个计算节点同时工作的大规模分布式计算^[3]。因此,采用分布式计算成为了解决大规模电力系

统仿真算力需求问题的有效方案^[4-5]。

目前,已有文章探讨了提高计算效率的方法,例如优化稀疏矩阵存储结构^[6]、使用开源数学求解器^[7]、使用图形处理器(GPU)进行加速^[8]等方法。然而,在分布式计算的运行过程中,一些计算节点不可避免的受到扰动因素的影响,例如,电源管理、软件或硬件的故障和维护、以及计算资源共享等。这些影响因素在分布式系统运行的过程中在所难免,并且会影响整个系统的性能,导致系统计算速度和

*通讯作者: 彭沛

可靠性的降低^[9]。由于大规模电力系统仿真具有计算数据量大和运算复杂度高等特点, 上述分布式系统固有问题所产生的影响更加明显。此外, 实时仿真往往具有较强的时效性, 因此, 需要有效避免有少数问题节点导致的计算效率下降问题。

目前国内外对解决上述问题的研究主要集中在两个方向。第一个方向是通过优化计算资源的分配来消除负面影响: Wang 等^[10]提出分布式计算系统可以提供一些多余的计算节点, 用于执行原计算节点相同的任务, 从而避免由少数计算节点导致的计算速度下降; Aktas 等^[11]提出当系统的运行过程中遇到一些计算节点的执行时间远超预期, 系统将取消当前节点的任务并重新执行。第二个方向是通过编码对任务本身进行优化从而避免负面影响。王艳等^[12]利用纠错码等方法对任务进行编码, 使得一个任务可以通过不同方式来完成。当系统使用编码时, 任务可通过执行任何固定大小的子任务的子集来完成, 此类任务执行为系统的任务计算带来了多样性, 从而减少了系统中计算速度过慢的节点产生的影响。

针对分布式计算系统中应用的理论建模研究方面, Gardner 等^[13]利用指数分布对计算时间进行建模分析; Joshi 等^[14]利用帕累托分布对计算时间进行建模分析; Gardner 等^[15]对分布函数进行了分析, 提出了一种更加普遍的计算时间分布函数; Peng 等^[16]考虑了任务的大小对分布式计算理论模型的影响。国内研究者也逐步将冗余编码思想应用于分布式计算系统和仿真平台中, 如张基^[17]等提出一种基于冗余和编码的容错算法, 该算法应用于 MapReduce 分布式计算框架; 刘怡等^[18]将 LT 编码应用于分布式矩阵运算, 从而降低系统时延提供鲁棒性。然而上述文献均未考虑到大规模电网状态估计中所具有的任务量大、计算复杂度高、时效性需求强等特点。

本文考虑大规模电力系统仿真的计算任务需求, 首先, 基于编码技术算法的计算时间进行了理论建模; 然后, 通过理论模型的机理分析, 确定了最优的编码率的取值范围, 以及在特殊场景中的最优编码率; 最后, 通过算例仿真验证了编码技术的有效性。

1 系统模型

1.1 分布式计算系统架构及编码

本文采用的分布式计算系统模型如图 1 所示, 用户将任务计算请求发送至管理节点, 管理节点对

任务进行编码后, 将编码得到的子任务分配给各计算节点。节点完成计算后将结果返回给管理节点, 由管理节点解码后将最终结果发送回用户。

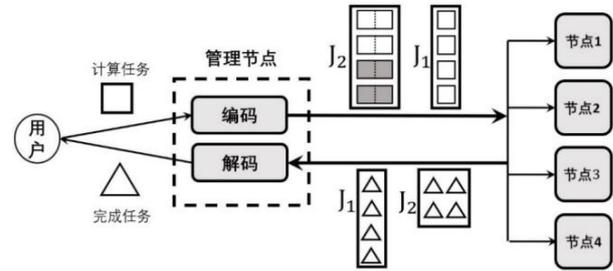


图 1 基于编码的分布计算系统架构

系统中的计算架构是由单个管理节点和多个计算节点组成的经典分布式计算系统架构。系统可将计算任务分割为多个子任务, 每个子任务可被不同的计算节点并行独立执行, 任务类型例如向量或矩阵相乘运算等。

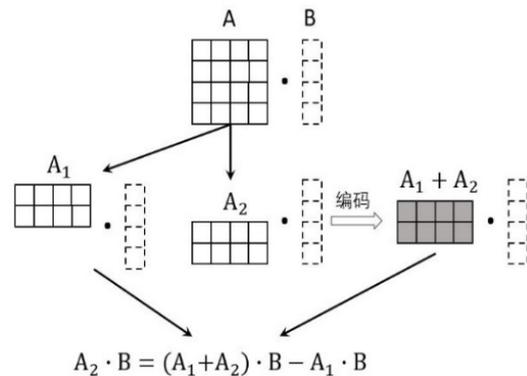


图 2 编码计算举例

管理节点的编码和解码是本文提高系统性能方法的核心。其原理如图 2 所示, 在计算矩阵 \$A\$ 和向量 \$B\$ 相乘时, 可以将 \$A\$ 拆分为两个相同大小的子矩阵 \$A_1\$ 和 \$A_2\$。基于这两个子矩阵, 可以进行简单的编码得到一个新的相同大小的子矩阵 \$A_1 + A_2\$。任意两个矩阵运算完成时, 解码过程如下: 例如当 \$A_1 \cdot B\$ 和 \$(A_1 + A_2) \cdot B\$ 计算完成后, 利用图中所示减法运算可以得到 \$A_2 \cdot B\$ 的结果, 从而获得矩阵运算 \$A \cdot B\$ 的结果。同理, 即任意两个矩阵运算完成时, 可以通过解码来获得矩阵运算 \$A \cdot B\$ 的结果。

综上所述, 该系统的详细计算过程如下。管理节点首先对任务进行预处理, 将任务分割为多个子任务, 并可以通过编码添加冗余。例如预处理任务 \$J_1\$, 管理节点将任务平均分割为 4 个子任务, 且不引入冗余; 预处理任务 \$J_2\$, 管理节点将任务平均分割

为 2 个任务, 并通过使用编码 (例如 [4, 2] 极大距离可分码) 增加 2 个冗余子任务。之后, 管理节点将预处理后的子任务分配给 4 个计算节点分别执行。计算节点执行完毕后, 将结果返回至管理节点。例如任务 J_1 需要 4 个计算节点均执行完毕, 而任务 J_2 则只需要任意 2 个节点执行完毕。之后, 管理节点会对各计算结果进行汇总和解码, 得到最终的任务执行结果。

1.2 数学建模

基于上述系统模型, 假设分布式计算系统总共有 n 个计算节点共同执行任务。管理节点首先将任务平均分割为 k 个子任务, 即 Y_1, Y_2, \dots, Y_k 。其次, 管理节点将通过使用极大距离可分码 (Maximum Distance Separable Codes, MDS) 编码, 添加 $n-k$ 个冗余任务, 即 $Y_{k+1}, Y_{k+2}, \dots, Y_n$ 。随后, 管理节点将 n 个子任务分配给计算节点。根据 MDS 编码的特性, 任意 k 个计算节点执行完毕后, 管理节点将解码任务执行结果, 同时取消剩余计算节点未完成的子任务。

本文主要通过优化分布式计算系统的任务完成时间的期望来满足大规模电力系统仿真的时效性需求。由于电力系统的数量大, 仿真运行的计算任务也通常比较大, 计算时间会明显随着任务大小变化。同时, 仿真中相关算法的计算复杂度比较高, 随着任务的增大算法本身也越来越复杂, 从而导致其计算时间的波动更加明显。

本文采用的分布式计算时间模型为位移指数分布, 即时间 $X \sim \text{ShiftedExp}(\Delta, \lambda)$, 其中位移参数 Δ 是任务完成所需要的固定时间, 率参数 λ 是计算时间的随机波动决定。结合上述大规模电力系统仿真相关算法的特点, 本文提出了任务计算时间随着任务大小的变化模型: 假设上述位移指数分布时间模型中, 固定时间 Δ 会随着任务大小的变化而变化, 即当一个任务完成所需的固定时间为 Δ , 计算节点完成一个 $1/k$ 大小的子任务时, 所需时间为 Δ/k 。同时, 假设上述时间模型中, 随机波动时间是由计算任务的复杂度决定, 然而任务的复杂度与任务大小也有一定的相关性。这里当一个任务完成所需的随机波动时间服从指数分布 $\text{Exp}(\lambda)$ 时, 计算节点完成一个 $1/k$ 大小的子任务时, 波动时间服从 $\text{Exp}(kl(k)\lambda)$, 其中 $l(k)$ 是计算复杂度随子任务大

小变化的函数。由于本文主要研究使用编码提高分布式计算的性能, 因此为了简化理论分析的难度, 下文的分析将会假设 $l(k) = 1$ 。在未来的工作中, 可以进一步结合实际场景分析计算复杂度 $l(k)$ 对分布式计算系统性能的影响。

为了清晰地展现下文所进行的理论分析, 表 1 中罗列了本文所使用的参数与对应的定义。

表 1 参数列表

参数	定义
n	系统中计算节点的数量
k	完成任务所需要的计算节点数量
$X_{k:n}$	任务完成时间
λ	位移指数分布的率参数
Δ	位移指数分布的位移参数
$l(k)$	计算复杂度随子任务大小变化的函数

2 理论分析

2.1 任务完成时间期望

假设分布式计算系统采用 $[k, n]$ MDS 编码, 考虑到编码在分布式计算系统的应用时, 计算任务完成时间需要使用顺序统计量的相关定理和结论。具体的定理推导参考了文献^[13]。设 X 是计算节点完成一个子任务所需要的时间, 且 X 是服从位移指数分布 $X \sim \text{ShiftedExp}(\Delta/k, kl(k)\lambda)$ 的随机变量。令 X_1, X_2, \dots, X_n 是随机变量 X 的 n 次采样。则其排序后最小的第 k 个随机变量是 $X_{k:n}$, 称之为 X_1, X_2, \dots, X_n 中的第 k 顺序统计量。根据顺序统计量的定义, 可知 $X_{k:n}$ 满足式 (1)。

$$X_{k:n} = X_{1:n} + X_{1:(n-1)} + \dots + X_{1:(n-k+1)} \quad (1)$$

因此, 经计算可以得到第 k 顺序统计量的期望 $E[X_{k:n}]$ 如式 (2) 所示。

$$\begin{aligned} E[X_{k:n}] &= \frac{\Delta}{k} + \frac{1}{kl(k)\lambda} \sum_{i=1}^k \frac{1}{n-k+i} \\ &= \frac{\Delta}{k} + \frac{1}{kl(k)\lambda} (H_n - H_{n-k}) \end{aligned} \quad (2)$$

其中, $H_n = \sum_{j=1}^n \frac{1}{j}$ 是调和数。

本文将根据公式 (2), 对基于电力系统仿真任务特点的分布式计算系统的性能进行分析, 希望能

够找到最优的编码率 (k/n) 使得任务完成时间的期望 $E[X_{k,n}]$ 最小。由于系统的计算节点数量 n 往往是固定的, 因此上述目标转化为寻找最优的 k 值。考虑到公式 (2) 较为复杂, 本文假设 $l(k)=1$ 。下文将对最优编码率的选取范围和极端场景下的最优编码率获取进行理论分析。

2.2 最优编码率的选取范围

研究 $E[X_{k,n}]$ 随 k 的变化规律是确定最优编码率的前提条件。而研究函数的单调性有助于得到函数的极值点, 进而得到 $E[X_{k,n}]$ 的最小值。因此本文通过对比 $E[X_{k,n}]$ 和 $E[X_{k+1,n}]$ 的大小确定 $E[X_{k,n}]$ 的单调性。根据式 (2) 可得式 (3), 显然公式 (3) 随着 k 的大小而变化。

$$\begin{aligned} & E[X_{k,n}] - E[X_{k+1,n}] \\ &= \frac{\Delta}{k} + \frac{1}{k\lambda}(H_n - H_{n-k}) - \frac{\Delta}{k+1} - \frac{1}{(k+1)\lambda}(H_n - H_{n-k-1}) \quad (3) \\ &= \frac{\Delta}{k(k+1)} + \frac{H_n}{\lambda k(k+1)} - \frac{1}{\lambda} \left(\frac{H_{n-k}}{k} - \frac{H_{n-k-1}}{k+1} \right) \\ &= \frac{1}{\lambda k(k+1)} (\Delta\lambda + H_n - \frac{k}{n-k} - H_{n-k}) \end{aligned}$$

2.2.1 任务完成时间的期望随 k 递增

假设 $E[X_{k,n}] < E[X_{k+1,n}]$, 由于 λ 和 k 均大于 0, 根据式 (3) 推导可得假设成立的条件如式 (4) 所示。

$$\Delta\lambda < \frac{k}{n-k} - H_n + H_{n-k} \quad (4)$$

已知 $H_n - H_{n-k} \geq \frac{k}{n}$, 对式 (4) 进行变换可得式 (5)。

$$\Delta\lambda < \frac{k^2}{(n-k)n} \quad (5)$$

最后, 经过简单推导可得, 式 (5) 成立的必要条件如式 (6) 所示。

$$k > n \sqrt{\frac{\Delta^2 \lambda^2}{4} + \Delta\lambda} + \frac{n\Delta\lambda}{2} \quad (6)$$

综合以上分析, 对于服从位移指数分布 $ShiftedExp(\Delta, k\lambda)$ 的分布式计算系统, 当 k 的取值范围满足式 (6) 时, 任务的平均完成时间 $E[X_{k,n}]$ 随着 k 的增大而增大。当 $\Delta\lambda$ 的值越小时, 满足 k 的范围会增大; 当 $\Delta\lambda$ 接近 0 时, $k=1$ 使得 $E[X_{k,n}]$ 达到

最小值, 此时系统中会引入尽可能多的冗余子任务。

2.2.2 任务完成时间的期望随 k 递减

假设 $E[X_{k,n}] \geq E[X_{k+1,n}]$, 由于 λ 和 k 均大于 0, 根据式 (3) 推导可得假设成立的条件为式 (7)。

$$\Delta\lambda \geq \frac{k}{n-k} - H_n + H_{n-k} \quad (7)$$

已知 $H_n - H_{n-k} \geq \frac{k}{n-k+1}$, 对式 (7) 进行变换可得式 (8)。

$$\Delta\lambda \geq \frac{k}{(n-k)(n-k+1)} \quad (8)$$

最后, 经过简单推导可得, 式 (8) 成立的必要条件为式 (9)。

$$k \leq n + \frac{1}{2} + \frac{1}{2\Delta\lambda} - \sqrt{\frac{1}{4} \left(\frac{1}{\Delta\lambda} + 1 \right)^2 + \frac{n}{\Delta\lambda}} \quad (9)$$

综上所述, 对于服从位移指数分布 $ShiftedExp(\Delta, k\lambda)$ 的分布式计算系统, 当 k 的取值范围满足式 (9) 时, 任务的平均完成时间 $E[X_{k,n}]$ 随着 k 的增大而减小。当 $\Delta\lambda$ 的值越大时, 满足 k 的范围会增大; 当 $\Delta\lambda$ 接近无穷大时, $k=n$ 使得 $E[X_{k,n}]$ 达到最小值, 此时系统中并不需要通过编码引入冗余。

2.2.3 最优 k 值取值范围

尽管上文的推导和分析无法直接得到最优的 k 值, 使得平均完成时间 $E[X_{k,n}]$ 最小, 但是相关结论可以极大地缩小判断最优 k 值的范围。显然, 当 k 满足式 (6) 时, $E[X_{k,n}]$ 是单调递增的; 当 k 满足式 (9) 时, $E[X_{k,n}]$ 是单调递减的。因此, 对于服从位移指数分布 $ShiftedExp(\Delta, k\lambda)$ 的分布式计算系统, 使任务的平均完成时间 $E[X_{k,n}]$ 达到最小值的 k 的取值范围如式 (10)、(11) 所示。

$$k > n + \frac{1}{2} + \frac{1}{2\Delta\lambda} - \sqrt{\frac{1}{4} \left(\frac{1}{\Delta\lambda} + 1 \right)^2 + \frac{n}{\Delta\lambda}} \quad (10)$$

$$k \leq n \sqrt{\frac{\Delta^2 \lambda^2}{4} + \Delta\lambda} + \frac{n\Delta\lambda}{2} \quad (11)$$

2.3 极端场景下的最优编码率

在上文推导最优编码率的取值范围的过程中, 根据式 (6) 和 (9) 可推测出: 当 $\Delta\lambda$ 的值很大或很小时, 最优的 k 取 n 或 1。这里考虑两种极端的场

景, 即计算时间是固定值和计算时间服从指数分布。

2.3.1 固定值计算时间

当分布式计算系统的任务计算时间是固定值 Δ , 根据式 (2), 子任务的计算时间均为 Δ/k , 即 $E[X_{k:n}] = \Delta/k$ 。显然 $E[X_{k:n}]$ 随着 k 的增大而减小。因此, 在任何参数条件下, 任务的平均完成时间 $E[X_{k:n}]$ 随着 k 的增大而减小, 且当 $k=n$ 时 $E[X_{k:n}]$ 最小。

2.3.2 指数分布计算时间

当分布式计算系统的任务计算时间服从指数分布 $Exp(k\lambda)$, 根据式 (2) 可得式 (12)。

$$E[X_{k:n}] = \frac{1}{k\lambda} (H_n - H_{n-k}) \quad (12)$$

此时, 再次对比 $E[X_{k:n}]$ 和 $E[X_{k+1:n}]$ 的大小, 可得式 (13)。

$$\begin{aligned} & E[X_{k:n}] - E[X_{k+1:n}] \\ &= \frac{1}{k\lambda} (H_n - H_{n-k}) + \frac{1}{(k+1)\lambda} (H_n - H_{n-k-1}) \quad (13) \\ &= \frac{1}{\lambda k} \left(\frac{H_n - H_{n-k-1}}{k+1} - \frac{1}{n-k} \right) \end{aligned}$$

根据调和数的定义可知:

$$H_n - H_{n-k-1} \leq \frac{k+1}{n-k-1} \quad (14)$$

将式 (14) 应用于式 (13) 中可得式 (15)。

$$\begin{aligned} & E[X_{k:n}] - E[X_{k+1:n}] \leq \\ & \frac{1}{\lambda k} \left(\frac{1}{n-k-1} - \frac{1}{n-k} \right) < 0 \quad (15) \end{aligned}$$

综合以上分析, 在任何参数条件下, 任务的平均完成时间 $E[X_{k:n}]$ 随着 k 的增大而增大, 且当 $k=1$ 时 $E[X_{k:n}]$ 最小。

3 仿真分析

本文的仿真基于 Python 完成, 仿真过程是基于前文中公式的数值仿真。本文考虑该分布式计算系统的计算节点数量为 $n=20$, 编码率和计算时间分布函数参数值会根据不同仿真图变化。

图 3 展示了任务完成时间的期望 $E[X_{k:n}]$ 随任务固定计算时间 Δ 的变化的仿真结果。图中对比了分布式计算系统采用 3/4 编码率的方案 (图中虚线) 和不使用冗余的方案 (图中实线) 时, 任务完成时间的期望的变化规律。该分布式计算系统的计算节点数

量为 $n=20$, 任务的随机波动时间服从指数分布且率参数 $\lambda=1$ 。由图可知, 两种情况下 $E[X_{k:n}]$ 均随着 Δ 的增大而线性增大。根据式 (1) 可知, $E[X_{k:n}]$ 与 Δ 是线性相关的。当 $\Delta < 5.5$ 时, 3/4 编码率方案的任务完成时间期望更小; 当 $\Delta > 5.5$ 时, 无冗余方案更好。因此, 在系统中适当的引入冗余 (即采用编码方案) 可以有效地降低任务完成时间。以下将通过仿真寻找不同系统参数条件下的最优编码率。

图 4 展示了任务完成时间的期望 $E[X_{k:n}]$ 随着完成任务所需节点数 k 的变化的仿真结果。根据任务固定计算时间和任务随机波动时间的大小, 考虑三种不同的情况: 情况 1, $\lambda=0.2$ 且 $\Delta=5$ (图中标星线), 此时任务的固定计算时间和随机波动时间的期望相同; 情况 2, $\lambda=0.1$ 且 $\Delta=0.1$ (图中标三角线), 此时任务的固定计算时间远小于随机波动时间的期望; 情况 3, $\lambda=5$ 且 $\Delta=5$ (图中标圆圈线), 此时任务的固定计算时间远大于随机波动时间的期望。

实验表明, 对于情况 1, 当 $k=15$ 时, $E[X_{k:n}]$ 达到最小值, 即采用 3/4 编码率可比无编码方案平均减少任务时间约 18%。对于情况 2, 当 $k=2$ 时, $E[X_{k:n}]$ 达到最小值, 即采用 1/10 编码率可比无编码方案平均减少任务时间约 67%。同时, 当固定计算时间与随机波动时间的差距进一步增大, 最优编码率会减小到 1/20, 即引入更大的冗余量。此观察结果与推论 2 中的结论一致。对于情况 3, 当 $k=20$ 时, $E[X_{k:n}]$ 达到最小值, 此时无编码方案即为最优方案。

总之, 当固定计算时间增大或随机波动时间减小时, 分布式计算系统应当采用的更大的编码率, 从而减小冗余的引入。反之, 系统则应采用更小的编码率, 从而增加冗余的引入。

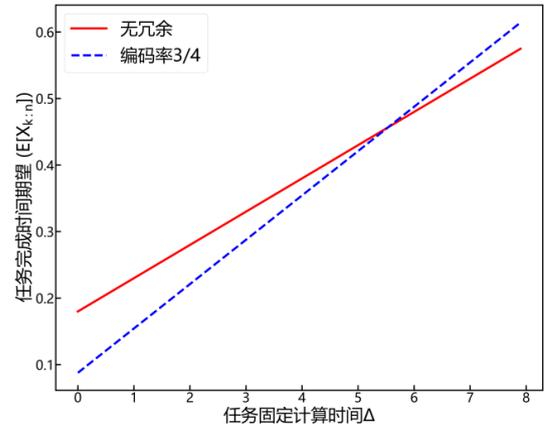


图 3 任务完成时间的期望 $E[X_{k:n}]$ 随任务固定计算时间 Δ 的变化

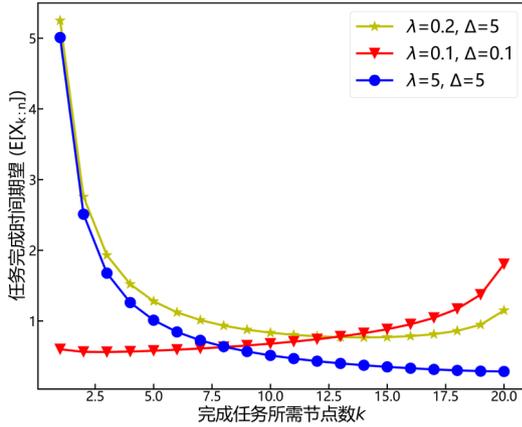


图 4 任务完成时间的期望 $E[X_{k,n}]$ 随着完成任务所需节点数 k 的变化

上述两个仿真结果验证了编码的有效性并展现了最优的编码率。这里, 将考虑不同参数条件的系统中, 最优编码率对比无冗余方案能够带来的系统性能提升。图 5 展示了计算时间减少比例随任务固定计算时间 Δ 的变化的仿真结果, 其中计算时间减少百分比的计算公式为 $(E[X_{n,n}] - E[X_{k,n}]) / E[X_{n,n}]$ 。考虑三种不同情况: $\lambda=1$ (图中实线)、 $\lambda=0.5$ (图中虚线) 和 $\lambda=0.1$ (图中点划线)。当 $\Delta=0$ 时, 三种情况均达到最大值, 此时计算时间减少大约 70%; 随着 Δ 的增大, 三种情况下计算时间减少幅度均有所下降, 且 λ 越大计算时间的减少的幅度越小; 图中即使任务固定时间 $\Delta=5$ 且任务随机波动时间的期望为 1, 使用合适的编码依然可以减少近 10% 的计算时间。

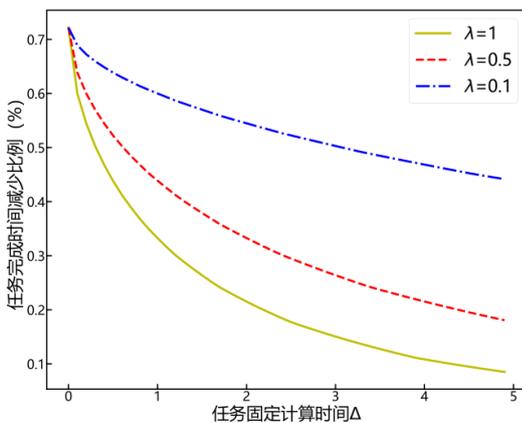


图 5 计算时间减少比例随任务固定计算时间 Δ 的变化

4 总结与展望

本文结合了大规模电力系统仿真的应用场景, 尝试应用编码技术解决分布式系统中少数问题计算

节点性能下降对系统性能产生的负面影响。首先, 基于电力系统仿真相关算法的任务特点, 创新的进行模型构建。其次, 通过理论分析确定最优的编码率的取值范围, 以及在特殊应用场景下的最优编码率。最后, 通过仿真分析结果验证编码技术的有效性和最优编码率的取值。综上所述, 应用编码技术并采用合适的编码率, 可有效地提高分布式计算的效率, 从而为大规模新型电力系统仿真提供算力支持。

本文假设大规模电力系统仿真中的任务计算时间模型为位移指数分布, 然而在实际系统中, 时间模型可能更为复杂, 未来可将本文的最优编码获取方式应用于真实系统, 进一步优化编码对分布式计算系统的性能提升。此外, 本文未考虑任务在节点间传输的通信时延与解码成本, 未来可结合真实并行系统仿真平台, 引入多任务优先级调度机制。

参考文献

- [1] 薛禹胜, 谢东亮, 薛峰, 等. 支持信息-物理-社会系统研究的跨领域交互仿真平台[J]. 电力系统自动化, 2022, 46(10): 138-148.
- [2] 杨争林, 曾丹, 冯树海, 等. 电力市场实验能力建设面临的挑战及关键技术[J]. 电力系统自动化, 2022, 46(10): 111-120.
- [3] Dalcin LD, Paz RR, Kler PA, et al. Parallel distributed computing using Python [J]. Advances in Water Resources, 2011,34(9): 1124-39.
- [4] 刘溪桥, 刘尧, 刘行健, 等. 基于分布式计算的电网模型拼接设计与实现[J]. 微型电脑应用, 2025, 41(2): 34-37+46.
- [5] 董雪涛, 冯长有, 朱子民, 等. 新型电力系统仿真工具研究初探[J]. 电力系统自动化, 2022, 46(10): 53-63.
- [6] 高毅, 王成山, 李继平. 改进十字链表的稀疏矩阵技术及其在电力系统仿真中的应用[J]. 电网技术, 2011, 35(5): 33-39.
- [7] 陈荃鞞, 龚成明, 赵晋泉, 等. 并行稀疏系统直接求解库 SuperLU_MT 在状态估计中的应用[J]. 电力系统自动化, 2017, 41(3): 83-88.
- [8] Karimipour H, Dinavahi V. Accelerated parallel WLS state estimation for large-scale power systems on GPU [C]. 2013

- North American Power Symposium (NAPS), Manhattan, IEEE, 2013: 1-6.
- [9] 郑腾飞, 周桐庆, 蔡志平, 等. 编码计算研究综述[J]. 计算机研究与发展, 2021, 58(10): 2187-2212.
- [10] Wang D, Joshi G, Wornell G. Using straggler replication to reduce latency in large-scale parallel computing [J]. ACM SIGMETRICS Performance Evaluation Review, 2015, 43(3): 7-11.
- [11] Aktas MF, Peng P, Soljanin E. Straggler mitigation by delayed relaunch of tasks [J]. ACM SIGMETRICS Performance Evaluation Review, 2018, 45(3): 224-31.
- [12] 王艳, 王希龄, 赖宏达, 等. 面向大规模矩阵乘法的编码计算性能研究[J]. 华东交通大学学报, 2021, 38(3): 41-51.
- [13] Gardner K, Zbarsky S, Doroudi S, et al. Reducing latency via redundant requests: Exact analysis [J]. ACM SIGMETRICS Performance Evaluation Review, 2015, 43(1): 347-60.
- [14] Joshi G. Synergy via redundancy: Boosting service capacity with adaptive replication [J], ACM SIGMETRICS Performance Evaluation Review, 2018, 45(3):21-8.
- [15] Gardner K, Harchol-Balter M, Scheller-Wolf A, et al. A better model for job redundancy: Decoupling server slowdown and job size [J]. IEEE/ACM transactions on networking, 2017, 25(6): 3353-67.
- [16] Peng P, Soljanin E, Whiting P. Diversity/parallelism trade-off in distributed systems with redundancy [J], IEEE Transactions on Information Theory, 2021, 68(2): 1279-95.
- [17] 张基, 谢在鹏, 毛莺池, 等. MapReduce 框架下结合分布式编码计算的容错算法[J]. 计算机工程, 2021, 47(4): 173-179.
- [18] 刘怡, 张磊. 基于 LT 码的分布式矩阵计算研究[J]. 计算机工程, 2024, 50(8): 328-335.

版权声明: ©2025 作者与开放获取期刊研究中心(OAJRC)所有。本文章按照知识共享署名许可条款发表。

<https://creativecommons.org/licenses/by/4.0/>



OPEN ACCESS