导读：高性能计算是国家发展战略和规划中的重要环节，其能力和水平更是国家综合实力的体现。随着大数据和人工智能时代的到来，致力于解决高效计算和高吞吐率数据分析的高性能计算技术(High Performance Computing，HPC)得到了越来越广泛的关注，已经应用于能源、航空航天、生物制药、天气预报、人工智能、数据挖掘等诸多领域。为了及时、集中地反映我国高性能计算领域取得的最新技术及应用成果，《电子技术应用》2022 年第 1 期特推出“高性能计算”专栏。专栏精心遴选 9 篇文章，内容涵盖并行计算模型、并行算法设计、性能优化技术与工具、高性能计算应用与平台等。希望对高性能计算领域的相关学者有所帮助。

特约主编：

薛巍：清华大学计算机科学与技术系副教授、博士生导师，青海大学计算机应用与技术系系主任，清华大学计算机系高性能计算研究所所长，中国计算机学会高级会员和信息存储技术专委会委员，IEEE 和 ACM 会员。主要研究领域为大规模科学计算、量化不确定性分析。曾获 ACM“戈登·贝尔”奖，教育部科学技术进步奖一等奖，中国电子学会电子信息科学技术奖一等奖，“清华大学-浪潮集团计算地球科学青年人才奖”。

张为华：复旦大学教授、博士生导师。2007 年复旦大学获博士学位，博士论文获“计算机学会优秀博士论文奖”。研究方向为体系结构、系统软件和并行计算等。作为项目负责人承担国家自然科学基金项目、科技支撑项目，上海市科委重点项目和 863 课题等项目。已在 MICRO、PPoPP、ATC 和 TPDS 等高水平国际会议和期刊发表论文 60 余篇。研究工作获得 ICPP 2015 和 ACA 2014 最佳论文奖。

基于最小割划分的数模混合仿真系统通信性能优化方法*

李亿渊[1]，穆　清[2]，薛　巍[1]
(1.清华大学 计算机科学与技术系，北京 100084；2.中国电力科学研究院，北京 100192)

摘　要：数模混合仿真是理解真实电网运行情况，支撑电网安全保障的重要手段。复杂的电网拓扑与硬实时的仿真需求对其计算性能提出了很高的要求。目前数模混合仿真多采用并行计算技术提高计算性能。随着处理器和集群技术的发展，异构集群系统逐渐成为高性能计算系统的主要构建方式。针对多层次的系统架构，已有的电网划分方式无法充分利用集群计算能力。如何应对多层次核间通信延迟变化问题，及引入设备交互导致的节点资源不对称问题是数模混合仿真任务划分与映射的新挑战。针对中国电力科学研究院自研电磁暂态仿真系统 ADPSS，基于最小割划分设计了两阶段的电网划分与进程映射一体化优化算法，在计算负载均衡和最小化通信上取得更好的平衡，进一步降低了电磁暂态仿真的通信时间。同时，该算法有效解决了集群节点资源不对称情况下的任务优化映射问题。通过在西北和华东真实电网算例上的模拟测试，所提出算法较 ADPSS 默认划分与映射算法取得了平均 40%和 50%的通信性能提升，平均 10%和 12%的总体计算性能提升。

关键词：数模混合仿真；图划分；最小割；进程映射；异构集群系统
中图分类号：TP391　　文献标识码：A　　DOI：10.16157/j.issn.0258-7998.212436

* 基金项目：国家电网公司科技项目(XT71-19-022)

中文引用格式：李亿渊，穆清，薛巍. 基于最小割划分的数模混合仿真系统通信性能优化方法[J].电子技术应用，2022，48(1)：2-11.

英文引用格式：Li Yiyuan，Mu Qing，Xue Wei. Communication optimization method of digital-analog hybrid simulation system based on min-cut partition[J]. Application of Electronic Technique，2022，48(1)：2-11.

Communication optimization method of digital-analog hybrid simulation system based on min-cut partition

Li Yiyuan[1]，Mu Qing[2]，Xue Wei[1]

(1.Department of Computer Science and Technology，Tsinghua University，Beijing 100084，China；
2.China Electric Power Research Institute，Beijing 100192，China)

Abstract：Digital-analog hybrid simulation is essential for understanding the real power grid and supporting power grid security. Complex power network topology and hard real-time simulation put forward high requirements for computing performance. At present，digital-analog hybrid simulation mainly uses parallel computing technology to improve computing performance. With the development of processor and cluster technology，heterogeneous cluster systems have gradually become the primary construction method of high-performance computing systems. For the multi-level system architecture，the existing power grid division methods can not fully use the cluster computing power. Dealing with the high latency of cross-layer communication and the unequal number of available processor cores on each computing node due to heterogeneous acceleration equipment is the main challenge of the partitioning and mapping algorithm. Aiming at the electromagnetic transient simulation system ADPSS developed by China Electric Power Research Institute，this paper designs a two-stage integrated optimization algorithm of power grid division and process mapping，which achieves a better load balance and minimizing communication，and further reduces the calculation time of the electromagnetic transient simulation. The algorithm is based on the min-cut partition and effectively solves the optimal mapping of sub-networks of unequal sizes on heterogeneous cluster systems. The simulation test was realized on the real power grid in Northwest and East China，compared with the ADPSS default partition and mapping algorithm，the proposed algorithm achieves an average communication performance improvement of 40% and 50% and an average overall computing performance improvement of 10% and 12%.

Key words：digital-analog hybrid simulation；graph partition；min-cut；process mapping；heterogeneous cluster system

0 引言

近年来我国经济不断发展，社会对能源的需求不断上升，电力的消耗也随之上升。我国电网的特高压工程持续投运以满足日益增长的用电负荷需求。电力系统整体规模的扩大也给整个系统的稳定和可靠运行带来了更高的安全风险。

电力系统仿真是分析电网特征、分析电网稳定性最重要的量化手段。电力系统仿真分为稳态仿真和动态仿真两类。动态仿真更关注电力系统的动态变化行为，主要包括机电暂态和电磁暂态两种。电磁暂态仿真建模更加精细，是动态安全评估的重要工具。通过电磁暂态仿真，研究人员能更好地理解电网在实际运行中的工作状态及其变化，从而在运行中有效调整控制方案，确保电力系统的安全、稳定运行。

电磁暂态是指电磁从一个稳定状态到另一个稳定状态中所经历的过程。在电力系统运行过程中，通常由于电子元件的开关切换、偶发的交直流故障以及雷击等干扰，造成电磁暂态过程的快速变化[1]。模拟电磁暂态现象一般通过电力系统的时域建模来完成。其目标是求出系统中各个时刻所有节点的电压值和电流值，核心算法是将连续的微分系统离散化，并使用迭代法隐式求解。仿真步长代表离散时间点间隔，步长越短就能模拟更高频的电网行为，故步长大小是衡量电磁暂态仿真系统质量的重要指标。

根据仿真平台计算所需时间和步长的大小关系，电力系统仿真又被分为离线仿真和硬实时仿真两种。离线仿真可以通过更长的计算时间得到单位时间中精准的计算结果，但较长的计算时间也限制了其应用场景。而实时仿真的应用场景更广，但为了能和真实电网保持同步运转，仿真程序的步长越小，计算负载也越大，想单单依靠通用处理器完成仿真有着不小的挑战[2]。

进一步，现代电力电子技术的发展也给电磁暂态仿真的实现提出了新的挑战。近年来电力系统中开始大量使用可再生新能源元件[3]，区域间电网互联水平大幅提升，总电网规模不断扩大[4-6]，电力电子设备被频繁应用在电网系统中[7-9]。这些都导致电力系统的规模和复杂性显著上涨。同时，大量电力电子设备通常伴随着高频的开关切换，为了依旧能对电力系统进行准确的仿真，

电力系统电磁暂态仿真的步长越变越小[10]。这就需要充分挖掘计算平台的性能才能满足实时计算需求。

计算硬件的发展也同样给仿真算法的设计与优化提出挑战。随着摩尔定律持续的放缓，集成电路特征尺寸的缩小愈发困难，单个通用处理器的计算性能增幅也很难维持十年前的趋势。计算平台只能走向多核、众核、异构等架构。虽然总计算性能继续得以增长[11-15]，但与此同时应用通信性能却随着非一致性内存访问(Non-Uniform Memory Access，NUMA)架构的广泛使用呈现层次化趋势。图1为鲲鹏920处理器的节点示意图，图中标出了跨不同NUMA层时两处理器核MPI点对点通信的延迟情况，可以看到随着两处理器核间跨越的NUMA层数越多，其通信延迟越大。这就要求在算法相对固定的前提下任务到处理器核的映射更为合理，使更多的通信发生在同NUMA内，尽可能减少通信延迟对仿真性能的影响。同时复杂的硬件通信延迟问题也给子网划分提出了更高的要求[16]。

本文针对中国电力科学研究院自研电磁暂态仿真系统(Advanced Digital Power System Simulator，ADPSS)[17]，提出了一种电磁暂态任务划分与映射算法，最大限度地降低通信延迟与抖动，提高仿真性能。该算法利用两阶段优化方法，将子网计算时间、子网间通信量、多NUMA层间不同的通信性能统一纳入考虑，并设计了多规模不等子图最小割算法，有效解决了子网在集群节点资源不对称情况下的任务优化映射问题，并在华东和西北真实电网算例上进行模拟性能测试。在目前ADPSS真实运行中，计算集群由于需配合FPGA进行异构加速，原本8节点、每节点28核的集群只剩下150核可用，且不均匀地分散在8节点上，在子网计算规模较大的西北算例上，计算性能已无法满足75 μs的实时仿真步长。而本文算法较ADPSS默认划分与映射算法取得了平均40%和50%的通信性能提升，平均10%和12%的总体计算性能提升，且所有算例均可满足实时步长需求。

1 ADPSS介绍

随着特高压交直流电网发展，国家电网公司于2009年建成了国家电网仿真中心，并由中国电力科学研究院研发了基于高性能集群的电力系统全数字仿真装置ADPSS。

ADPSS实现交流大电网连多回直流输电系统的数模混合仿真，即该仿真系统中包括了数字模拟电网，同时又包括了真实的控制保护装置，为确保实际控制保护装置能正确运行，数字电网需要和真实电网保持同步运转。故此系统为硬实时系统，数字电网的单时步计算时间需永远小于真实电网单步时间。

数模混合仿真有两个主要作用。一是通过数字电网模拟真实电网，在该场景下测试实际控制保护装置自身工作是否正常，能否真正接入真实电网使用。二是部分实际控制保护装置结构复杂，无法直接在数字系统中建模，必须使用数字系统加控制保护装置共同准确描述电网真实特性，如此一来就能基于准确的电网和设备模型来研究真实电网中出现故障后的运行特点，防患于未然。所以能否对更大规模的电磁暂态网络进行仿真，直接决定对真实电网进行模拟与研究的能力。

为了应对更大规模电网电磁暂态实时仿真需求，对仿真系统提出了更高的性能要求，并行求解变成了必不可少的一环。ADPSS提供了节点分裂分网和传输线分网两种并行方案。节点分裂分网算法需要子网统一到主控节点进行求解[18-19]，而根据传输线划分得到的子网间在同一时步内没有耦合关系，不需要统一求解。对于大规

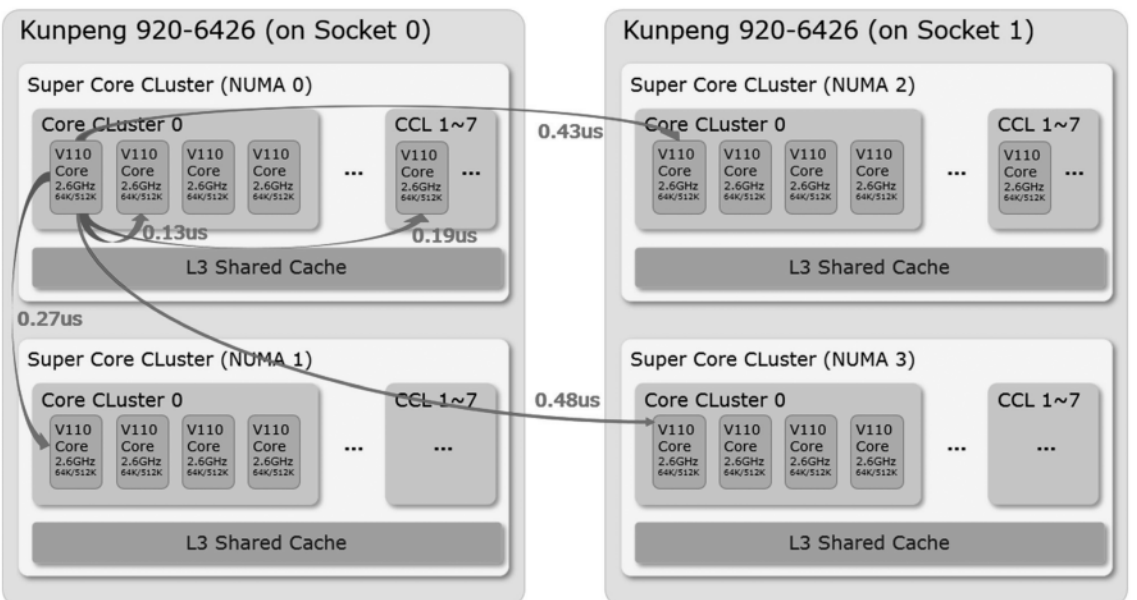

图1 鲲鹏920核间通信延迟示意图

模交直流电网的实时仿真系统而言，节点分裂的方式虽然更加灵活，但会带来更多的固有串行计算部分，当分网断面较多时，串行计算会严重影响仿真的实时性。而基于传输线分网可以让整个网络解耦成为很多子网，每一个子网之间没有导纳阵上的耦合联系，只有信号量的交互。相互交互的信号量自然延迟一个时步，天然并行。故为了确保大规模仿真计算的有效性，ADPSS 采用了传输线分网的并行方案。

如上所述，传输线分网并行方案中交流大电网首先会被分割成多个解耦的小网络。小网数量往往远大于并行计算机的处理器核数，因此需要对小网进行融合完成任务划分。ADPSS 系统中配套了以不同子网间电器元件数量均衡为目标的自动分网程序：该程序先根据长传输线位置将电网分至最小网络状态，并把它们按元件数量从大到小排序。再使用贪心的策略将它们合并至指定个数(总处理器核数)。

在完成分网后，在每个进程上的计算则相对固定，其流程如下：

(1)各子网计算基于 LU 分解矩阵的前代和回代的求解；

(2)各子网(进程)间通信，传输分网线路上的端口电压、电流和控制系统的交互量，此处只完成发送操作；

(3)各子网计算每一个元件状态；

(4)等待各元件(含外部元件)返回的结果，即接收第(2)步中的通信数据；

(5)数据收齐后进入下一时步，重新回到第(1)步。

在以上 5 步中，步骤(1)和步骤(3)的计算时间相对稳定，且计算所需时间与真实小网中电子元件有关，无法进一步缩短，只有步骤(2)和步骤(4)中的通信性能可能进一步优化，同时也成为了性能瓶颈。而通信量也和小网中电子元件相关，可优化幅度有限，如何更好地映射进程，将每条通信更合理地分配在不同 NUMA 层间，成为优化关键。

同时 ADPSS 面临的仿真规模越来越大，需要越来越大规模的计算设备才能有效完成实时仿真的问题。当跨不同 NUMA 间通信性能差异越来越大，此时原有自动分网程序只针对电器元件数量均衡为目标的优化算法和默认进程映射方案就会出现性能不足的问题。同时，这种通信容易出现性能抖动，使整个硬实时仿真失效。

同时，交直流混合电磁暂态仿真计算也需要更复杂的异构硬件来支持。为了对外接口以及提高性能，ADPSS 在硬件架构上整合了现场可编程逻辑门阵列(Field Programmable Gate Array，FPGA)，导致部分集群计算节点的处理器核需配合 FPGA 计算，无法参与仿真。因此，集群节点层面存在不同节点上的可用核数不同的问题。如图 2 所示，其中每个小圈代表一个处理器核，核内显示了该核是否可用。显然如此分配的 16 核比集中

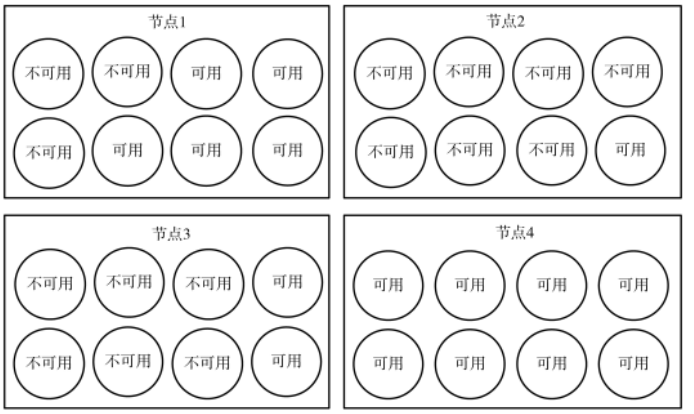

图 2　节点资源异构示意图

在 2 个节点上的 16 个核更难于映射，容易在节点间暴露更多的通信量，这就对仿真任务的映射提出了更高的要求。

针对以上需求，ADPSS 遇到的问题是如何自动划分、合并子网，如何在考虑节点资源异构的前提下更合理地将子网映射到处理器核上进行计算，使性能最佳。

2 ADPSS 的任务划分与映射需求

根据第 1 节介绍，ADPSS 需要优化算法完成的工作如图 3 所示。从传输线分网后的小网开始，将其合并至特定个数子网(图 3(b)中同种数字小网合并)，并映射至处理器核(图 3(c)中代表每个处理器核的状态，不可用或计算某个子网任务)。

针对上述目标，分网问题可以看作一个优化问题。该问题的输入为全部小子网的信息(计算成本、对外连接情况等)，目标为将子网聚合成与计算核数相等个子网，并映射在对应核上，在运行时每步抖动不超过阈值的情况下，尽可能快。

给出如下问题定义：

无向图 $G_O=(V_O,E_O)$，V_O 为点集，E_O 为边集，代表最小子网状态下各子网的连接状态。$|V_O|=N_O$ 为子网总数，$|E_O|=M_O$ 为子网间通信边数，集合 $\mathrm{Calc}_O=\{\mathrm{Calc}_1,\mathrm{Calc}_2,\cdots,\mathrm{Calc}_{N_O}\}$ 为每子网的计算时间，计算时间可由子网中每种电子元件计算时间累加近似得到或者实测得到。$\mathrm{WO}_{(u,v)}=\mathrm{Comm}_{\mathrm{WO}(u,v)}$ 代表子网节点 u、v 间存在一条通信量为 $\mathrm{Comm}_{\mathrm{WO}(u,v)}$ 的通信边，即 $E_O=\{\mathrm{WO}_1,\mathrm{WO}_2,\cdots,\mathrm{WO}_{M_O}\}$。

无向图 $G_P=(V_P,E_P)$，V_P 为点集，E_P 为边集，代表子网合并后各子网的连接状态。$|V_P|=N_P$ 为子网总数，$|E_P|=M_P$ 为子网间通信边数，集合 $\mathrm{Calc}_P=\{\mathrm{Calc}_1,\mathrm{Calc}_2,\cdots,\mathrm{Calc}_{N_P}\}$ 为每子网的计算时间，计算时间由合并至此子网中每个小网计算时间累加得到，假设为原始小网 X、Y 合并为子网 Z，则：

$$\mathrm{Calc}_{P_Z}=\mathrm{Cala}_{O_x}+\mathrm{Cala}_{O_y}+f(\mathrm{Comm}_{W(X,Y)}) \tag{1}$$

其中 $f(\mathrm{Comm}_{W(X,Y)})$ 为合并 $W(X,Y)$ 的代价，由小网 X、Y 间传输线的组成决定。$\mathrm{WP}_{(u,v)}=\mathrm{Comm}_{\mathrm{WP}(u,v)}$ 代表合并后子

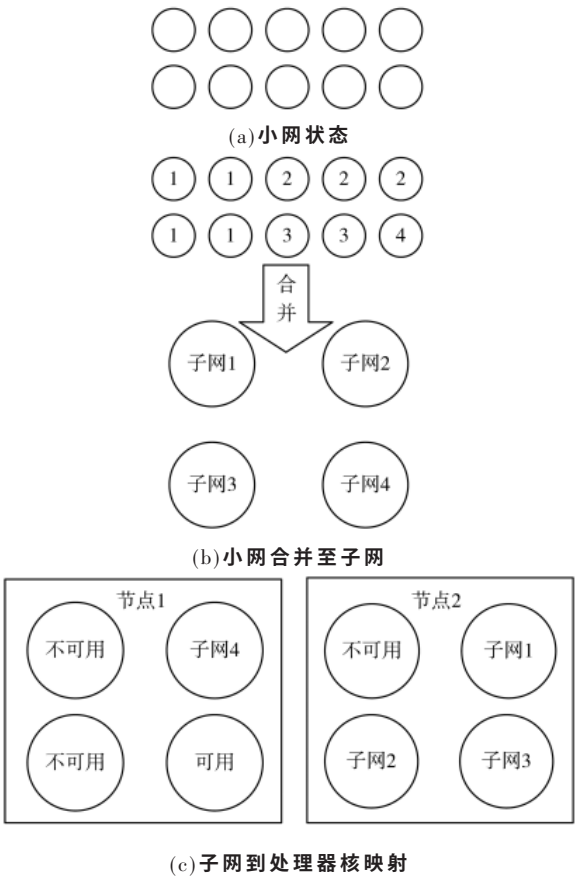

图 3 小网划分、子网映射算法工作流程示意图

网节点 u、v 间存在一条通信量为 $\mathrm{Comm}_{\mathrm{WP}(u,v)}$ 的通信边，即 $E_{\mathrm{O}}=\{\mathrm{WP}_1,\mathrm{WP}_2,\cdots,\mathrm{WP}_{M_P}\}$ 在合并后对应边权叠加即可。

无向图 $G_{\mathrm{D}}=(V_{\mathrm{D}},E_{\mathrm{D}})$，$V_{\mathrm{D}}$ 为点集，E_{D} 为边集，代表无向图 G_{P} 中的子网映射在计算节点上的状态。$|V_{\mathrm{D}}|=N_{\mathrm{D}}=N_{\mathrm{P}}$ 为总处理器核心数，$|E_{\mathrm{D}}|=|E_{\mathrm{P}}|=M_{\mathrm{P}}$，代表子网间通信实际在计算硬件上的通信链路。$\mathrm{WD}_{(u,v)}=\mathrm{Comm}_{\mathrm{WD}(u,v)}$ 代表处理器核 u、v 间存在一条通信延迟为 $\mathrm{Comm}_{\mathrm{WD}(u,v)}$ 的边，代表当某两个子网被映射在处理器核 U 和 V 上时，子网间的通信延迟为多少。S 为硬件集群节点数，C_i 为 i 号节点上的可用核数，$\sum_{1}^{S} C_i=V_{\mathrm{D}}$。

上述优化问题的目标为将图 G_{O} 合并为图 G_{P}，并将 G_{P} 中的子网映射在 G_{D} 上。为了使抖动尽可能低，需要跨节点通信量尽可能低，同时每步的计算时间尽可能低。下面，将介绍一种两阶段的优化方案，即先完成 G_{O} 到图 G_{P} 的子图划分，再考虑 G_{P} 到 G_{D} 的任务映射。

3 基于贪心策略和最小割图划分的两阶段优化算法

3.1 基于贪心策略的小网划分算法

根据 ADPSS 算法流程可知，通信需要在第一步 LU 分解完成后才能开始，而第三步中电子元件的计算时间又远小于 LU 分解，故每时步时间可近似看作 LU 的计算时间加通信时间。LU 计算时间越小，就能越早开始通信，更早地完成时步的计算。而硬实时仿真需要每个子网永远都能在预设的时间内完成一步仿真，换个角度看，即需要所有子网中每时步所需时间最长的子网，能永远在阈值时间内完成，即可满足要求。根据以上分析，可得子网合并的优化目标：使计算时间最大的子网，计算时间尽可能小。

同时根据式(1)定义可知，任意两节点合并，可使全图总边权下降，同时总计算时间上升，即对最小割图划分有益，而对尽可能小的计算时间有害。因此每次合并的策略很明确：使每次合并后得到的新子网时间，是当前所有满足合并条件的子网对中最小的，这样无论从总计算时间上考虑，还是从单个子网计算时间上考虑，都是计算时间增长最小的。

根据上述思想可直接得到贪心算法：每次遍历边集 E_{O}，计算合并每一对 $\mathrm{WO}_{(u,v)}$ 子网后得到的新子网计算时间 $\mathrm{Calc}_{\mathrm{p}}=\mathrm{Calc}_{U_o}+\mathrm{Calc}_{V_o}+f(\mathrm{Comm}_{W(U,V)})$，并找出此步内最小的 $\mathrm{Calc}_{\mathrm{p}}$，并合并所对应的子网 U 和 V。重复该策略多次，直至合并后的子网数等于总处理器核心数 N_P。

3.2 基于最小割算法的计算节点核数不对称问题的映射优化

3.2.1 最小割算法建图分析

在上述划分完成的图 G_{P} 基础上，由于计算处理器规格相同，电磁暂态仿真计算时间相对固定，核心是通过进程映射实现通信性能尽可能优化。

基于进程映射的通信优化主要有两方面：

(1)进程间 MPI 通信可以使用节点内共享内存通信，性能较节点间更优。同时，电磁暂态仿真的通信量仅仅是子网边界的节点信息，总量相对较低，带宽竞争问题较少。因此，要保证尽可能多的通信是在同一个计算节点内发生，使用共享内存通信方式，让更少的通信通过性能不佳且容易出现抖动的跨节点 MPI 进程通信接口完成。这一策略的优化目标可以直接将通信量看为图上的边权，直接选择图上的最小割算法完成。

(2)尽可能让容易抖动的跨节点核间通信与计算时间较长子网的计算同时进行。利用计算通信重叠，掩藏掉核间通信的时间和抖动。如果每时步最后完成的时刻为计算时间最长子网完成计算，并通过节点内完成通信的时间，其他子网间通过节点间通信的时间就会被完全覆盖，无需担心性能抖动带来的影响。完成这一目标，不能简单地将通信量作为边权并求最小割。如图 4 所示，有 3 个子网，子网 1 的计算时间最长，子网 1、2，子网 2、3 间各有一条通信，且子网 2、3 间的通信量更大。3 个子网需分在 2 个节点上计算，如果直接按通信量划分，则会将子网 2、3 映射在一个节点内，如图 4(a)所示，虽然更大的通信量得以更高效的完成，但当子网 1 计算完毕后，需要通过更低效的核间通信完成与子网 2 的数据交换，使得最终的完成时间被进一步拖慢。而如果能将计算通信重叠纳入考虑，将子网 1、2 映射在一个节点

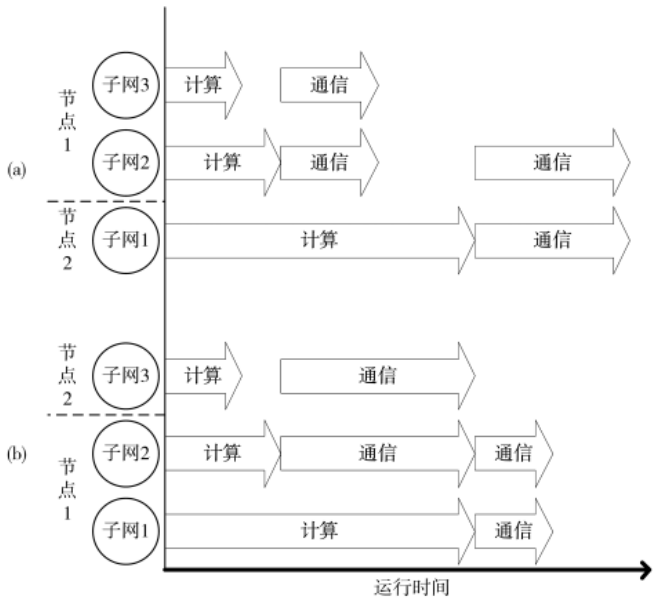

图 4 计算通信重叠示意图

内，虽然子网 2、3 间更大的通信需要花费更长的时间完成，但这些通信时间都被子网 1 的计算所覆盖了，而当子网 1 完成计算后，可以通过高效的节点内通信迅速完成与子网 2 的数据交换，使得整体时间更短。

基于上面的思路，对边权 E_P 的定义进行调整，将边权 $\mathrm{Comm}_{WP(u,v)}$ 定义为：

$$\mathrm{Comm}_{WP(u,v)}=\mathrm{Max}(\mathrm{Calc}_{P_u},\mathrm{Calc}_{P_v})+\alpha\cdot\mathrm{Comm}_{WP(u,v)} \quad (2)$$

其中，α 为硬件通信抖动系数，α 需在不同平台上通过点对点和模拟真实 Mesh 场景的 MPI 基准测试得到。每条边权代表的意义由原来的通信延迟，变成每步内完成该边通信的最晚时刻，通信开始的时间为映射在两侧核上子网的计算时间的较大值，而通信取最坏情况下抖动上限，$\mathrm{Comm}_{WP(u,v)}$ 取跨多 NUMA 层通信时性能的最大值。

通过建图，就将子网本身的通信与计算集群的通信性能相结合，子网到计算核的映射问题，就转化成了经典的图划分问题，计算集群有几个计算节点、每个节点中有几个处理器核，对应图划分问题中将原图划分为几个子图，各个子图中有几个图节点。图划分的割边之集总和越小，跨节点通信的总边权就越小。下面几节将介绍如何解决在已建完的图上求出一个子图规模不等的最小割划分，只要得到最小割图划分，就能对应得到子网到计算节点的映射关系。

3.2.2 基于 KERNIGHAN-LIN ALGORITHM 算法的规模不等子图最小割算法

为了解决上述图的最小割划分问题，本文选择使用 KERNIGHAN-LIN ALGORITHM 算法[20]（后续简称 KL 算法），使用递归分支的策略将图逐层划分至最终目标。

传统的 KL 算法只能解决将图划分成两个大小均等的子图。根据第 2 节中优化问题的定义，在本文优化需求下，计算子网映射到的计算节点是资源异构的，即每个计算节点上的可用处理器核数不尽相同。从图划分角度看，即为将全图划分成若干个大小不同的子图，使割边之和尽可能小且最长计算时间最小。要解决这个优化问题，就需要对原始的 KL 算法做调整。

下面将先分别解决两个子问题：(1)划分成二的幂次个大小相等的子图；(2)划分成两个大小不等的子图。再使用划归的思想将这两个子问题的解决方案合并，实现划分成若干个大小不等的子图，得到最终的最小割算法。

3.2.3 划分二的幂次个大小相等子图的最小割算法

该子问题对应真实场景中不使用计算资源异构的情况，即每颗处理器上能使用的核数都是相等的，映射到每个处理器节点上的子网数是相等的。

假设需要将图划分成 $S=2^n$ 大小相等的子图，直观地，可以通过 n 轮的递归分治，将原图划分为 S 部分。即先通过 KL 算法将原图划分为 2 个均匀子图，再分别对这 2 个子图使用 KL 算法，将它们划分为 4 个大小均等的子图，以此类推。总共通过调用 $S-1$ 次 KL 算法即可完成划分。需要注意的是，如果每次递归划分都是在不同 NUMA 层上，需要根据当前情况下的最大通信延迟调整图中的边权。如图 1 中，假设第一次划分代表将子网映射在 socket0 还是 socket1 上，而具体在哪个 NUMA 上并不由该次划分决定，则边权中通信延迟取 48 μs；当第二轮递归划分，决定已经在 socket0 上的子网，是映射在 numa0 还是 numa1 上时，需要将边权按通信延迟 27 μs 调整。

但是，KL 算法是随机设定初值的算法，仍需要通过多次运行取最优的方式来找到尽可能优的全局解。针对递归的计算模式，可以选择直接将整个递归过程运行若干次取最优，也可以在每层递归调用的 KL 内部，运行多次随机，取最优。在本文实现中，选取的是每次递归调用 KL 时都运行多次的方案，运行 10 000 次。

3.2.4 划分两个规模不等子图的最小割算法

假设图中有 N 个节点，需要划分成分别包含 N_1 和 N_2 个节点的子图 $N=N_1+N_2$。

利用化归的思想，可以通过向图中添加虚拟节点的方式，把问题变成划分成两个大小相等的子图进行求解。具体方案如图 5 所示。

设 $N_1>N_2$，则向图中添加 N_1-N_2 个虚拟节点，并让它们全连接，并把边权设置为无穷大，这样在划分过程中这些虚拟节点必然不会被划分开。接着对这个包含了 $N+N_1-N_2$ 个节点的图调用 KL 算法。算法为了保证划分得到的两个子图大小相等，这 N_1-N_2 个虚拟节点必然会和 N_2 个原图节点分在一组形成一个拥有 N_1 个节点的子图，与另一个包含 N_1 个原图节点的子图大小相等。图5 展示了当 $N=8$，$N_1=6$，$N_2=2$ 时的添加虚拟节点、划分的过程，先添加了 4 个全连接的虚拟节点(用正方形表示，以方便区分)，后正常划分为均包含 6 个节点的子图。

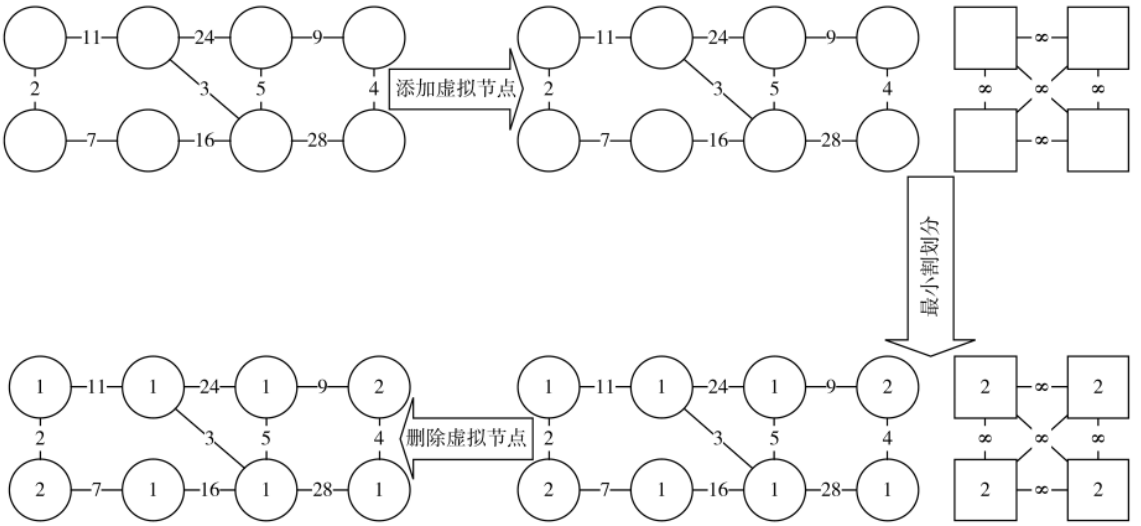

图 5 KL 算法不均等二分示意图

3.2.5 划分若干个规模不等的子图的最小割算法

假设需要将图划分成 S 个大小不等的子图，子图大小为 $N_1N_2\cdots N_S$。

首先解决 $S=2^n$ 的情况。同样利用递归分治的策略，先将图划分成两个子图，并将前一半子图节点之和作为第一个子图的节点个数，后一半子图节点之和作为第二个子图的节点个数。即定义 $N_x=N_1+N_2+\cdots+N_{0.5S-1}$，$N_y=N_{0.5S}+N_{0.5S+1}+\cdots+N_S$，设 $N_x\geqslant N_y$，则在图中添加 N_x-N_y 个虚拟节点，并调用 KL 算法进行图划分。依此类推，直至划分完成。图 6 为假设 $S=4$，$N_1=1$，$N_2=2$，$N_3=2$，$N_4=3$ 时的示意图。

其次，解决 $S\neq 2^n$ 的情况。同样按递归分治的方法逐层划分，唯一的区别是当某次划分时需要划出的子图数无法被 2 整除时，需要额外处理。假设 $S \bmod 2\equiv 1$，定义 N_x 为前 $(S+1)\cdot 0.5$ 个子图的节点之和，N_y 为 $(S-1)\cdot 0.5$ 个子图的节点之和，设 $N_x\geqslant N_y$，则同样在图中添加 N_x-N_y 个虚拟节点并调用 KL 进行划分，则可以正常将原图分为包含 $(S+1)\cdot 0.5$ 和 $(S-1)\cdot 0.5$ 个子图的子图。依次类推，即可划分得到所有 S 个子图。图 7 为假设 $S=5$，$N_1=1$，$N_2=2$，$N_3=1$，$N_4=1$，$N_5=3$ 时的示意图。

通过上述最小割算法构建，只要将需要图信息、需要划分至几个子图、每个子图上有多少个节点作为算法输入，就能得到以最小割为目标的可行解。对应真实场景下，只要将划分完成的子网信息、计算集群节点资源信息输入，就能得到子网到处理器核的映射关。

3.3 两阶段优化算法总结

优化算法的完整流程如图 8 所示。在运行 ADPSS 前，先将传输线划分后的小网数据和集群计算资源的可使用情况输入给优化程序，小网数据包括每个子网内电子元件数据和子网间的通信关系。优化程序会运行一遍子网划分的贪心算法，再根据用户指定的迭代次数，运行若干次最小割算法，保留最优解，并输出进程映射关系。此后，ADPSS 的仿真程序读入划分结果和进程映射关系，并根据该映射关系分配每个进程上的子网计算任

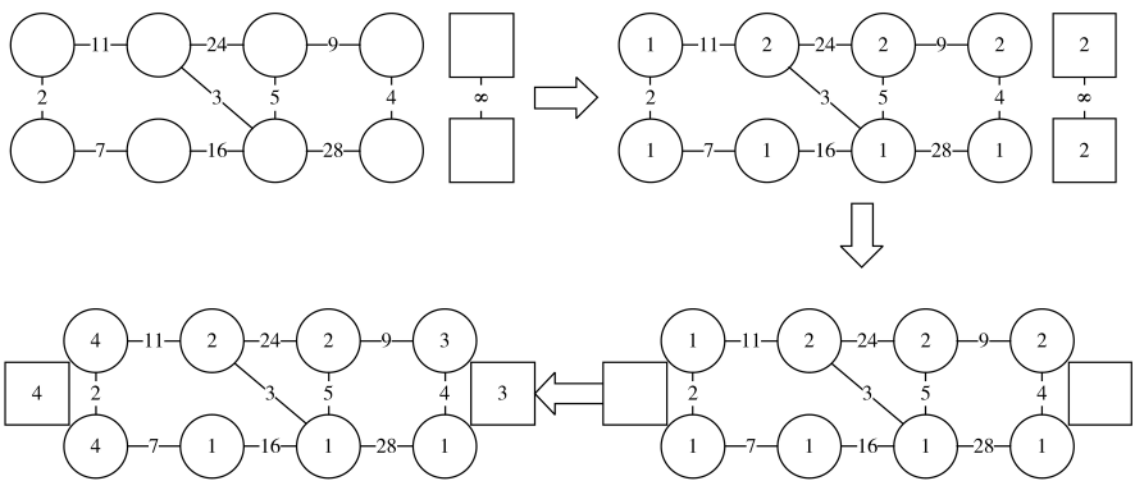

图 6 KL 算法不均等 2 的幂次划分示意图

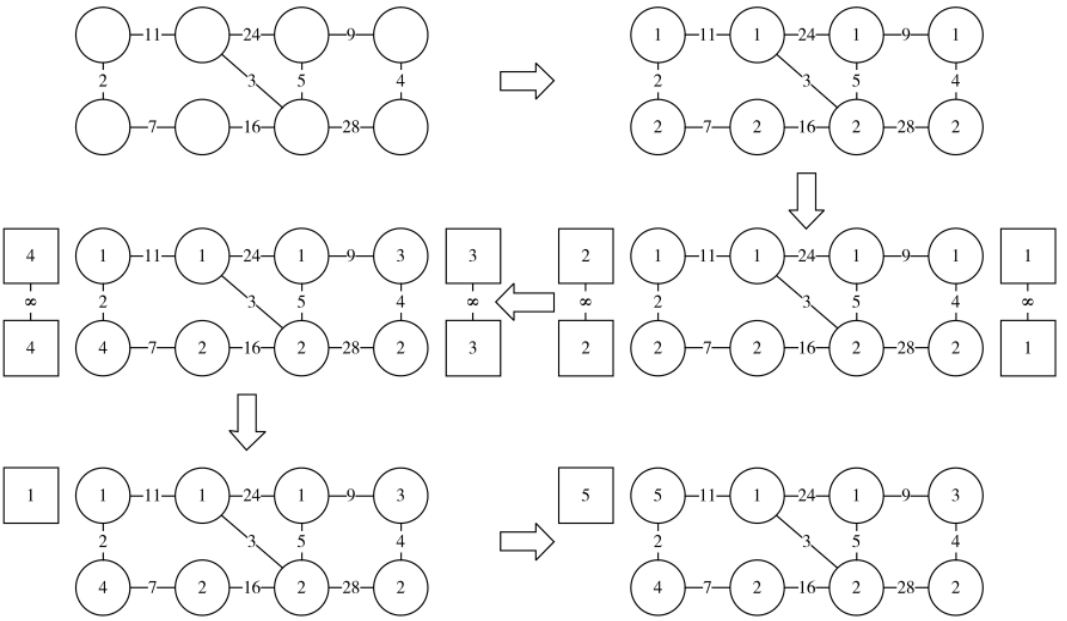

图 7　KL 算法若干不均子图划分示意图

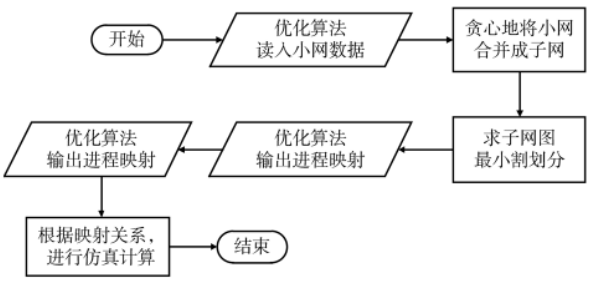

图 8　优化算法与仿真程序运行关系流程图

务，进入正常仿真流程。

4　算法性能评测

本算法在 ADPSS 中的应用仍在电力科学研究院进一步开发之中。本文仅能利用各计算小网的计算部分实测得到的时间，以及网间通信量的记录结构，加上本文构建的底层通信中间件搭建的模拟程序在真实集群系统上进行评测，并与已有方案进行性能对比。

测试算例为华东和西北真实电网算例。真实算例 1 为西北电网算例。该算例电网包含 8 400 个母线，1 000 台发电机，3 900 条线路，共 673 个小网，步长大小75 μs。真实算例 2 为华东电网算例。该算例电网包含 5 000 个母线，400 台发电机，4 640 条线路，共 364 个小网，步长大小 35 μs。两个算例的小网每时步内计算量和通信量如图 9 所示。真实算例性能对比如图 10 所示。

具体评测方案如下：首先在 APDSS 真实运行场景

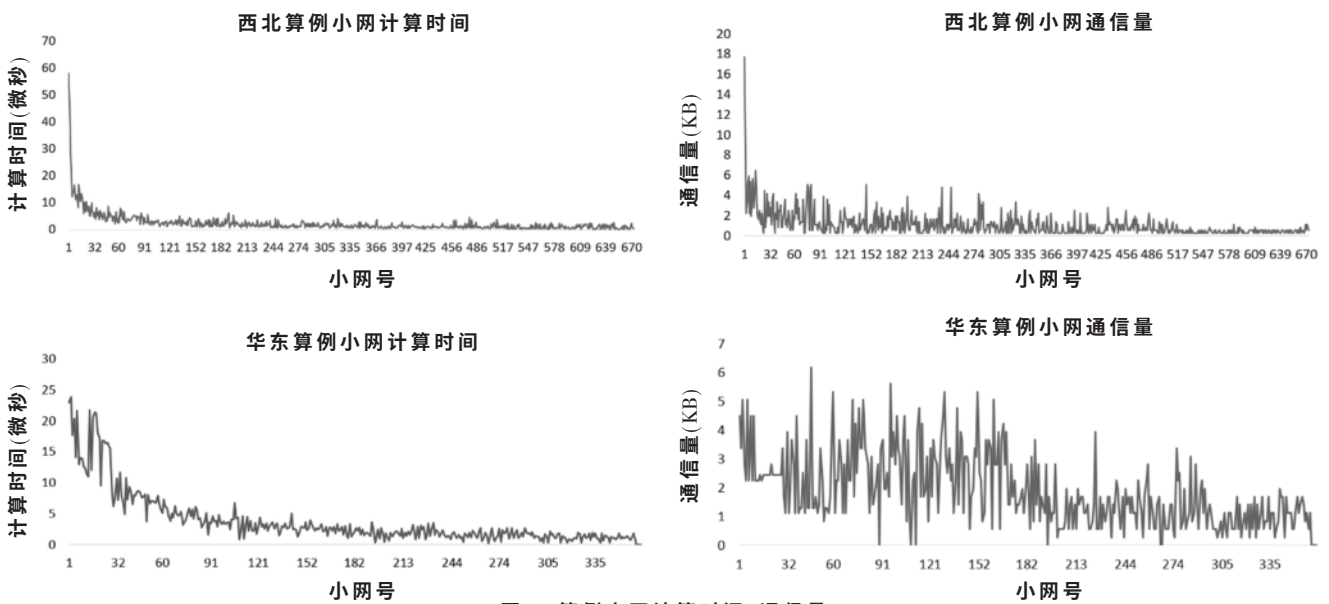

图 9　算例小网计算时间、通信量

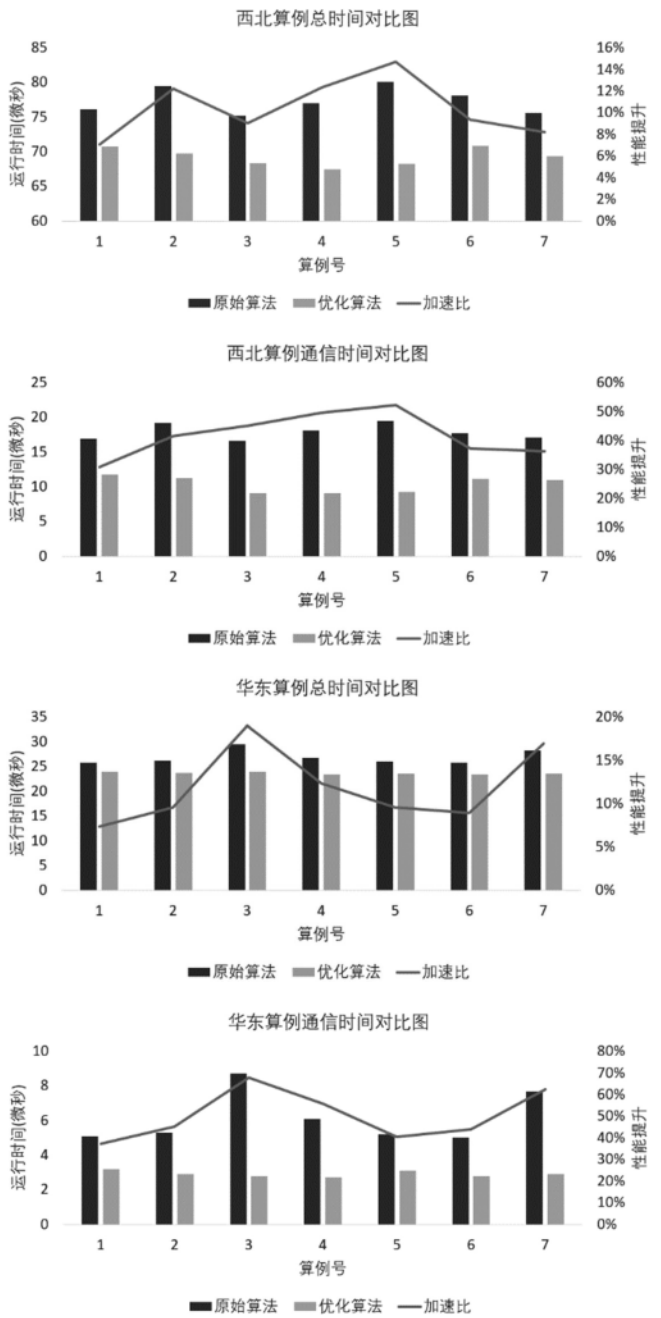

图 10　真实算例性能对比

中，最大并行度为 150 进程，所以所有测试中并行度固定为 150 进程。其次设置了多种资源异构场景，不同算例中每个节点资源情况如表 1 所示，既包括了 8 节点计算资源平均的情况，也包括模拟真实场景各节点资源不一的情况，以及极端不均衡的情况(某个节点无核可用，或只有 1 个核可用)。极端不均衡的情况容易对 ADPSS 默认划分、映射算法带来巨大的性能影响，因为子网一般按照计算时间顺序排列，无论是从大到小还是从小到大，计算时间最长的子网都会在某个极端算例下被单独映射在只有 1 个核能用的节点上，从而大大影响仿真性能。同时，不均衡的算例也可以理解成白盒测试，在已知默认算法的划分、映射策略的情况下，人为选择可能对其性能影响最大的处理器核分配方式。因为默认算法只包含贪心策略，故无论子网按何种贪心策略排列，总能出现对其性能影响最大的算例。而如同前文所述，ADPSS 为按计算时间大小排序，故测试中的极端算例将核数较少的处理器分配在头或尾。最后，电网算例皆为图 9 中展示的真实算例。测试对比算法为 ADPSS 目前使用的默认划分与映射算法。

模拟程序测试的底层平台为基于 Intel® Xeon® Gold 6132 处理器，共 8 计算节点，每节点 2 颗 CPU，24 核(全集群 192 核)，配有 192 GB 内存。通信采用 Infiniband 互联，带宽 100 Gb/s，测试代码采用 Intel 2019 套件编译，MPI 采用 Intel 2019 编译版本。在测试中采用向集群申请全部 192 核的方式，并根据模拟的处理器核可用情况和优化算法的图划分结果，将子网映射到对应的处理器核上进行模拟计算。

从上述两个算例(图 10)可以看出，本文算法较 ADPSS 默认算法在通信性能上分别取得了平均 50%和 40%的性能提升，而由于计算时间不变的影响，在总时间上取得的提升略低，分别为平均 12%和 10%。而且在西北算例中，由于初始小网中最大计算时间就高达 58 μs，已与 75 μs 的步长相差不大，原始算法是在不同核数分配下均无法完场硬实时仿真，而优化后的算法在不同核数下都能稳定在 70 μs 下，完成仿真。

当节点资源出现不同情况的异构时，本文优化算法的鲁棒性更好。在整体性能相当稳定，波动不到 10%。与原有算法相比，本文算法在算例 3、4、5 的性能甚至略有提升。这主要因为这几个算例中最大的计算资源为 24 核，大于算例 1 平均状态下的 19 核，这样就有更多

表 1　不同算例中各节点资源情况

算例号	节点 1	节点 2	节点 3	节点 4	节点 5	节点 6	节点 7	节点 8	总核数
1	19	19	19	19	19	19	18	18	150
2	20	14	15	11	24	24	20	22	150
3	14	23	11	15	21	24	19	23	150
4	0	6	24	24	24	24	24	24	150
5	24	24	24	24	24	24	6	0	150
6	1	5	24	24	24	24	24	24	150
7	24	24	24	24	24	24	5	1	150

的空间去将关键通信链路藏在同一节点内，而算例6、7虽然也同样有大量24核的节点，但却包含一个核的节点，则不可避免地会出现较多的跨节点通信，整体性能依然与平均状态相当。

反观原始算法，在华东和西北算例上随着资源异构都出现了不同程度的性能波动，上下波动幅度甚至高达60%。在默认映射状态下，进程随着异构核被动变化，跨节点通信链路的多少无法针对通信性能进行优选。若出现映射不佳情况，就会出现大量通信处于跨节点状态，大大影响通信性能。

综合来看，本文优化算法取得了可观的优化效果。ADPSS计算性能的持续提升未来将有赖于对计算部分的进一步优化。

5 结论

本文针对中国电力科学研究院自研的电磁暂态仿真系统ADPSS，在资源异构集群上进行了任务划分和进程映射算法的优化与改进，取得了通信性能的有效优化。本文所提出的优化算法实现了从初始小网合并划分，再到进程映射的全自动优化，并基于KL算法设计了能解决不同节点资源各异的任务映射问题的方案。在华东和西北两大真实电网算例上的模拟测试显示，相较ADPSS默认划分算法，本文算法在整体仿真性能上取得了平均10%和12%的提升，通信性能上取得了平均40%和50%的提升。

参考文献

[1] DOMMEL H W, MEYER W S. Computation of electromagnetic transients[J]. Proc. IEEE, 1974, 62(7): 983-993.

[2] DOMMEL H W. Techniques for analyzing electromagnetic transients[J]. IEEE Computer Applications in Power, 1997, 10(3): 18-21.

[3] HOLLMAN J A, MARTÍ J R. Real-time network simulation with PC-Cluster[J]. IEEE Power Engineering Review, 2002, 22(8): 64-64.

[4] MORCHED A, MARTI L. A high frequency transformer model for the EMTP[J]. IEEE Transactions on Power Delivery, 1993, 8(3): 1615-1626.

[5] IBRAHIM E S. Interconnection of electric power systems in the Arab world[J]. Power Engineering Journal, 1996, 10(3): 121-127.

[6] RODRIGO A S, PERERA C U. Modeling and simulation of current source converter for proposed India-Sri Lanka HVDC interconnection[C]//2015 IEEE 10th International Industrial and Information Systems(ICIIS). IEEE, 2015.

[7] BIRLA Y, SARADA P. Future energy system integrating renewable energy sources into the smart grid through industrial electronics[J]. International Journal of Electrical and Electronics Engineers, 2015, 7(2).

[8] GUERRERO J M, HANG L, UCEDA J. Control of distributed uninterruptible power supply systems[J]. IEEE Transactions on Industrial Electronics, 2008, 55(8): 2845-2859.

[9] BLAABJERG F, LISERRE M, KE M. Power electronics converters for wind turbine systems[C]//Energy Conversion Congress and Exposition(ECCE). IEEE, 2011.

[10] BELANGER J, VENNE P, PAQUIN J N. The what, where and why of real-time simulation[Z]. 2010.

[11] 谢立前. 大规模电磁暂态实时仿真系统的快速设计与实现[D]. 上海：上海交通大学, 2020.

[12] 饶鑫宇. 具有高扩展性的电磁暂态仿真平台硬件设计与实现[D]. 上海：上海交通大学, 2020.

[13] 陈颖，宋炎侃，黄少伟，等. 基于GPU的大规模配电网电磁暂态并行仿真技术[J]. 电力系统自动化, 2017, 41(19): 7.

[14] 王明轩，陈颖，黄少伟，等. 适用于CPU+GPU协同架构的大规模病态潮流求解方法[J]. 电力系统自动化, 2018, 42(10): 5.

[15] SONG Y, YING C, HUANG S, et al. Efficient GPU-based electromagnetic transient simulation for power systems with thread-oriented transformation and automatic code generation[J]. IEEE Access, 2018, 6: 25724-25736.

[16] CHAN K W, DUNN R W, DANIELS A R. Efficient heuristic partitioning algorithm for parallel processing of large power systems network equations[J]. IEE Proceedings-Generation, Transmission and Distribution, 1995, 142(6): 625-630.

[17] 王玭，李亚楼，陈绪江，等. 基于ADPSS新一代仿真平台的大规模交直流电网数模混合仿真[J]. 电网技术, 2021, 45(1): 227-234.

[18] SHU J, XUE W, ZHENG W. A parallel transient stability simulation for power systems[J]. IEEE Transactions on Power Systems, 2005, 20(4): 1709-1717.

[19] Xue Wei, Shu Jiwu, Wu Yongwei, et al. Parallel algorithm and implementation for realtime dynamic simulation of power system[C]//2005 International Conference on Parallel Processsing, Proceedings(ICPP 2005), 2005: 137-144.

[20] KERNIGHAN B W, LIN S. An efficient heuristic procedure for partitioning graphs[J]. The Bell System Technical Journal, 1970, 49(2): 291-307.

（收稿日期：2021-12-06）

作者简介：

李亿渊(1995-)，男，博士研究生，主要研究方向：高性能计算、自动代码生成。

穆清(1983-)，男，博士，高级工程师，主要研究方向：高压直流输电和电力电子仿真。

薛巍(1974-)，男，博士，副教授，主要研究方向：高性能计算和量化不确定性分析。

扫码下载电子文档

《电子技术应用》http://www.chinaaet.com

版权声明