CPU-GPU异构计算框架下的高性能用电负荷预测

赵嘉豪[1],周 赣[1],黄 莉[1],陆春艳[2],陶晓峰[2],冯燕钧[1]
(1. 东南大学 电气工程学院,江苏 南京 210096;2. 南瑞集团有限公司(国网电力科学研究院有限公司),江苏 南京 211106)

摘要:随着电网的快速发展,用电信息采集系统的数据计算业务面临着巨大挑战。近年来,图形处理器(GPU)因其在浮点计算速度和存储带宽方面的优势成为高性能计算问题中的研究热点,也被成功应用在电力系统计算分析等科学计算领域。在基于人工智能方法的电力负荷预测问题中,以往大部分研究仅考虑了使用GPU加速预测模型的训练,而并未应用在数据集的获取和计算上。提出了一种基于中央处理器-图形处理器(CPU-GPU)异构计算框架下全流程加速的高性能用电负荷预测方案。首先结合统一计算架构(CUDA)和多线程技术实现了使用多台GPU完成用电负荷的并行预处理,随后在聚类分析后基于XGBoost算法完成了多台区负荷预测,并利用GPU加速了模型的训练计算。最后通过对深圳市43 254个台区用电信息的实例分析,验证了所提方法的高效性与适用性。

关键词:用电信息采集系统;负荷预测;GPU;异构计算;XGBoost
中图分类号:TM 715 **文献标志码**:A DOI:10.16081/j.epae.202106008

0 引言

在智能电网时代,随着智能电表的广泛使用和各类传感器的普及,电力用户侧的数据呈指数级增长。由于数据驱动的计算业务的快速扩展,现有用电信息采集系统在统计实时性和分析智能化方面面临着巨大挑战,如何高效处理和充分挖掘海量用电数据已成为一个亟需解决的问题。电力负荷预测是用电数据挖掘分析的重要业务场景之一,影响到电力系统发电与用电量之间的平衡。精确、高效的负荷预测结果有利于提高电网调度水平和安全稳定运行状况[1]。

电力负荷预测主要包括2个环节:用电数据的获取及预处理,计算模型的训练及结果预测。通常,数据获取方案都是在中央处理器(CPU)支撑的关系型数据库上执行的。在面对数据规模较大的查询、计算时,传统的数据库(如Oracle),受限于硬件成本与CPU计算能力,常以较长的响应时间为代价换取较高的吞吐量,通常在数据的获取和预处理方面耗时较长。另一方面,负荷预测是人工智能技术在电力系统应用最广泛、深入的应用。众多学者对采用人工智能模型预测电力负荷展开了大量研究,主要方法有神经网络[2]、支持向量机[3]和集成学习[4]等。其中,以XGBoost为代表的集成学习方法,在负荷预测中取得了良好的效果。文献[5]对比分析了随机森林(RF)、贝叶斯和K最近邻(KNN)等模型,证明XGBoost在电力负荷预测方面具有优越性。文献[6-8]将XGBoost与模型融合技术相结合,采用组合预测的方式提高了预测精度。然而随着负荷数据和模型规模的增长,算法复杂程度也逐渐提升,基于人工智能方法的预测模型的训练过程变得极为耗时,执行效率同样受限于CPU的计算资源。虽然传统的负荷预测应用大部分集中于批量数据的离线训练和随后的在线预测,但是单纯的批处理学习方法无法实时整合规模日益增长的用电信息,也不能在短期负荷预测问题中及时进行新样本的增量训练[9]。因此,为了满足用电大数据分析的要求,除了增加分布式并行机的节点数量,一种更高效的解决方案是使用计算能力更强的新型硬件和执行性能更高的计算框架,以提高整体预测方案的执行效率。

近年来,图形处理器(GPU)由于其在浮点计算速度和存储带宽方面的优势,已被成功应用于许多科学计算领域[10-12]。在数据库计算上,有学者设计了GPU数据库来加速数据的查询和计算[13-15]。在人工智能相关模型的训练上,主流神经网络框架如Tensorflow、Pytorch都支持GPU为其底层数学运算进行模型加速。但是之前学者在基于人工智能方法进行负荷预测的工作中,通常只使用GPU加速模型的训练过程,并未应用在数据集的获取和处理上。而负荷计算不仅支撑了负荷预测应用的开展,也是用电信息采集系统日常运营维护的基本内容。由于更新频率较高,其计算效率的提高十分重要。

基于以上问题,本文提出并实现了一种基于中央处理器-图形处理器(CPU-GPU)异构计算框架下全流程加速的高性能用电负荷预测方法,其充分利

收稿日期:2020-08-07;**修回日期**:2021-04-15
基金项目:国家自然科学基金资助项目(51877038);国家电网公司总部科技项目(5400-202018421A-0-0-00)
Project supported by the National Natural Science Foundation of China(51877038) and the Science and Technology Project of SGCC(5400-202018421A-0-0-00)

用CPU-GPU异构体系的计算资源，并行化加速了用电负荷数据的预处理和预测模型的训练。本文首先将用电数据的计算从数据库转移到GPU设备上进行并行化处理，并提出结合OpenMP多线程技术实现在多台GPU上同时完成多个并行计算任务。然后，对多个台区负荷数据统一建模预测，在进行K均值（K-means）聚类分析后利用GPU加速XGBoost模型的训练计算，从而预测台区未来的日负荷数据。最后，实际算例结果表明，结合多线程技术的基于GPU实现的并行计算方案具有高效的计算效率和良好的扩展性，基于K-means-XGBoost的台区负荷预测模型在高效执行的同时也保证了预测准确性，CPU-GPU异构计算框架满足了海量用电数据实时统计和充分挖掘的要求。

1　CPU-GPU异构计算框架和CUDA编程模型

CPU-GPU异构计算框架一般由1台CPU和a台GPU组成，CPU和GPU之间通过PCI-Express总线进行通信和数据传输，如图1所示。1台GPU包含c台流式多处理器SM（Streaming Multiprocessor），每台SM上有着成百上千的流处理器SP（Streaming Processor）资源，即计算核心单元，因此GPU可以提供比多核CPU更高的并行度和更强的浮点计算能力。SM采用单指令多线程SIMT（Single Instruction Multiple Thread）方式来管理其中的线程，GPU可以通过调度更多的SM和数量众多的线程实现更细粒度的数据级并行。

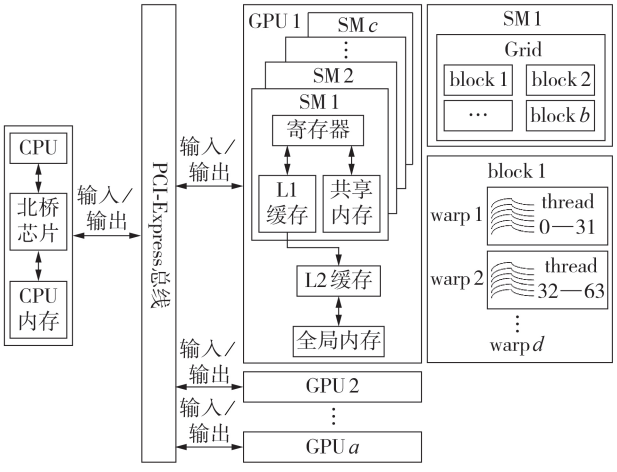

图1　CPU-GPU异构计算框架和CUDA编程模型

Fig.1　CPU-GPU heterogeneous computing framework and CUDA programming model

NVIDIA公司于2007年推出了GPU的统一计算架构（CUDA）模型[16]，使原本负责图形处理的GPU也可进行通用的科学计算。CUDA模型为CPU-GPU异构计算框架的实现提供了支持。其中，CPU作为主机端（Host）运行，而GPU作为设备端（Device）配合CPU协同处理。CPU程序调用GPU上运行的CUDA内核函数（Kernel）时，每个内核函数占用SM上的1个线程网格（Grid），每个线程网格包含b个线程块（block），每个线程块以32个线程（thread）为1组的线程束（warp）形式进行调度，线程束总数为d。通过设定内核函数启用的线程块数量和每个线程块中的线程数量，可以使得每个线程对不同数据并行执行相同的计算命令。同时，为了更好地利用GPU的内存带宽优势，可以通过“合并访存”的高效机制来掩盖线程的访存延迟，其实现的要求是同步执行的线程束中每个线程的读取和存储地址必须连续。

2　基于GPU加速的用电负荷并行预处理

在用电信息采集系统中，用电数据存储于Oracle等关系型数据库中。由于用电信息采集系统数据量日益增大，一直以来基于数据库存储过程的用电采集统计计算工作的耗时也与日俱增。本文提出将计算任务转移到GPU设备上，以GPU并行计算的方法来改进原有串行的数据库计算方案。

2.1　数据交互方式设计

在数据交互方式的设计上，通过Oracle调用接口OCI（Oracle Call Interface）实现负荷数据的接收和计算结果的回传。OCI是Oracle数据库最底层的原生接口，其将结构化查询语言（SQL）和高级编程语言相结合，性能上具有高度的开发灵活性、较高的执行效率和强健的安全性。本文采用短链接的方式为每条数据库执行指令都进行一次数据库连接与端口操作。另外，考虑到数据交互的通用性及易维护性，选择了可扩展标记语言（XML）文件作为存储表格信息的载体，实现数据内容和流程管理的分离。

2.2　基于GPU实现的用电负荷并行计算

由于用电信息采集系统具有通信延迟性，采集到的用户数据通常是混乱的。用电数据的计算业务往往围绕多张表格进行数据查找、匹配。遍历排序是一种逻辑性强的操作，而海量数据的移动又是内存绑定的操作。考虑到CPU擅长处理复杂逻辑运算，GPU擅长处理计算密集型的任务，本文先在CPU端进行数据重排序预处理，然后交由GPU完成相应的数值计算。选取用户ID作为数据记录的区分标识，在CPU上将原始数据按照用户ID大小顺序排列得到一个排序向量$\boldsymbol{P}$，排序结果和原始数据一起从CPU内存拷贝到GPU内存，然后在GPU上执行原始数据实际排序的内存操作。GPU上内核函数执行重排序任务的原理是赋值操作，即以1个线程执行1条记录的数据赋值，将1片内存中的数据按照向量$\boldsymbol{P}$重新写入另一块内存中，完成表格的重新构建。由于表格是按列读取的，连续的线程负责连续的数据，

写入内存是合并访存的。

考虑到采集数据的缺失，表格的重排序操作并不意味着可以直接进行数据匹配，而为后续的表格扫描和建立映射提供了方便。以日负荷量计算为例进行计算说明，日负荷计算取电能表2天的底码相减乘以综合倍率，底码包括正（反）向有（无）功功率的尖峰平谷示值。假设电能表采集数据来自测量点数据表A和日冻结电能表B，表A中的每条记录最多对应表B中的2条记录。本文采用并行扫描的方式建立表A和表B之间的映射关系，通过二分法寻找2个表中ID相同的记录，并将其数据位置存储在1个映射索引Map中。附录A图A1中的算法1解释了在GPU上实现并行扫描生成映射索引的计算原理。

通过并行扫描完成数据匹配之后，进一步根据映射索引对负荷数据进行数值计算，内核函数的设计方式如附录A图A1中的算法2所示。在确定当前ID存在2条数据记录后，启动1个线程根据映射索引获取两日的负荷数据和综合倍率，完成1个ID对应的负荷数据计算。由于硬件限制，GPU可调用的实际线程数量远小于需要处理的数据量，这里需要更新线程索引进行迭代计算。

2.3 结合OpenMP实现的多GPU任务调度机制

为了解决更大规模数据的并行计算，本文通过OpenMP多线程技术设计了多台GPU完成多个并行计算任务的调度机制。OpenMP是用于共享内存并行系统的多处理器程序设计的一套指导性编译处理方案，提供了对并行算法的高层抽象描述。OpenMP采用fork-join的执行模式，开始时只存在一个主线程，当需要进行并行计算时，派生出若干个分支线程来执行并行任务，当并行代码执行完成后，分支线程汇合，并把控制流程交给单独的主线程。在本文的异构计算方案中，CPU主线程先根据计算业务合理划分数据，然后开启OpenMP多线程并设定与GPU设备数量相等的CPU线程，CPU的线程编号和GPU的设备编号进行绑定。此时，相应计算业务的数据流通过CPU线程控制传输到绑定的GPU内存中，实现计算数据的独立；CPU的1个线程负责1个业务的GPU计算流程，实现计算流程的独立。这样就实现了CPU多线程控制多个GPU同时开展并行计算任务。在所有独立计算业务完成后，不同GPU设备上的计算结果进行拼接汇总，回传到CPU端，最后通过OCI写入数据库。

多线程多GPU并行计算框架同样适用于单一计算业务的执行。当单一计算业务的数据量过大而超过GPU单卡内存时，可将业务数据分拆到多台GPU设备上进行计算，最后将各个计算结果进行拼接得到总体的计算结果。

3 基于GPU加速XGBoost的台区负荷预测

基于GPU的并行计算加速了用电数据的获取，缩短了用电信息的处理时间。利用计算结果中的负荷数据可以进一步开发台区负荷预测应用。不同于神经网络算法，XGBoost是一种通过多个分类回归树（CART）的迭代计算拟合残差的集成学习算法[17]，在众多回归预测问题上表现出较好的效果。由于XGBoost具有良好的可扩展性和并行性，能够有效解决大数据的快速处理问题，针对大数据环境下的电力负荷预测具有较好的应用前景。

3.1 XGBoost算法原理

XGBoost是优化后的树集成模型，通过对梯度提升树（GBDT）的改进力争将速度和效率发挥到极致。XGBoost从数学角度可建立一个泛函优化问题，如式（1）所示，对n个样本建立K_t棵树的目标函数$L(\boldsymbol{\Phi})$分为损失函数和正则化项2个部分。损失函数项$l(\hat{y}_i-y_i)$表示训练误差，即样本i的预测值$\hat{y}_i$和真实值y_i的差距，鼓励模型尽量拟合训练数据；正则化项$\boldsymbol{\Omega}(f_k)$表达式如式（2）所示，表示模型f_k复杂程度越低，泛化能力越强。

$$L(\boldsymbol{\Phi})=\sum_{i=1}^{n}l(\hat{y}_i-y_i)+\sum_{k=1}^{K_t}\boldsymbol{\Omega}(f_k) \tag{1}$$

$$\boldsymbol{\Omega}(f_k)=\gamma T_k+\frac{1}{2}\lambda\left\|\boldsymbol{w}_k\right\|_2^2 \tag{2}$$

式中：T_k为叶子节点的个数；$\boldsymbol{w}_k$为叶子节点的权重向量；γ和λ为惩罚系数。

在XGBoost模型训练过程中，每一轮迭代都在现有树的基础上增加1棵树拟合前面树的预测结果与真实值之间的残差，新加入增量函数$f_t(x_i)$的目标是使目标函数尽可能降低，第t轮目标函数可写为：

$$L^{(t)}(\boldsymbol{\Phi})=\sum_{i=1}^{n}l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)+\boldsymbol{\Omega}(f_t)+C \tag{3}$$

式中：C为常数项。对式（3）中近似误差函数$l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)$进行泰勒级数展开，并且保留二次项，假设$g_i=\partial_{\hat{y}_i^{(t-1)}}l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)$，$h_i=\partial^2_{\hat{y}_i^{(t-1)}}l\left(y_i,\hat{y}_i^{(t-1)}+f_t(x_i)\right)$。由于第$t-1$轮树的预测值$\hat{y}_i^{(t-1)}$已知，可与常数项一起移除后将原目标函数推导为：

$$\begin{aligned}L^{(t)}(\boldsymbol{\Phi})\approx&\sum_{i=1}^{n}\left(l\left(y_i,\hat{y}_i^{(t-1)}\right)+g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)\right)+\\&\boldsymbol{\Omega}(f_t)+C\approx\sum_{i=1}^{n}\left(g_if_t(x_i)+\frac{1}{2}h_if_t^2(x_i)\right)+\boldsymbol{\Omega}(f_t)=\\&\sum_{i=1}^{n}\left(g_iw_{q(x_i)}+\frac{1}{2}h_iw_{q(x_i)}^2\right)+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2=\\&\sum_{j=1}^{T}\left[w_j\sum_{i\in I_j}g_i+\frac{1}{2}w_j^2\left(\sum_{i\in I_j}h_i+\lambda\right)\right]+\gamma T\end{aligned} \tag{4}$$

定义每个叶子节点j的样本下标集合表达式为

$I_j=\{i|q(x_i)=j\}$，即每个样本值都能通过函数$q(x_i)$映射到树上的某个叶子节点。这时，目标函数只依赖于每个数据点在误差函数上的一阶导数和二阶导数。进一步定义$G_j=\sum_{i\in I_j}g_i$，$H_j=\sum_{i\in I_j}h_i$，可得最终的简化公式为：

$$L^{(t)}(\Phi)=\sum_{j=1}^{T}\left[G_jw_j+\frac{1}{2}(H_j+\lambda)w_j^2\right]+\gamma T \tag{5}$$

令w_j的导数为0，可得到叶子节点j最优权重w_j^*，将此最优解代入目标函数可得：

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{6}$$

$$L^{(t)}(\Phi)=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j}{H_j+\lambda}+\gamma T \tag{7}$$

此时$\frac{G_j}{H_j+\lambda}$表示每个叶子节点对当前模型损失的贡献程度。由于损失函数越小代表模型越好，采用贪心算法对每个特征进行线性扫描确定其最优分割点，再对所有特征进行叶子节点分裂后，选择获得最大增益的特征。分割点L_{split}的计算表达式为：

$$L_{\text{split}}=\frac{1}{2}\left[\frac{G_{\text{L}}^2}{H_{\text{L}}+\lambda}+\frac{G_{\text{R}}^2}{H_{\text{R}}+\lambda}-\frac{(G_{\text{L}}+G_{\text{R}})^2}{H_{\text{L}}+H_{\text{R}}+\lambda}\right]-\gamma \tag{8}$$

式(8)中等号右侧第1、2项分别表示左、右子树分裂后产生的增益；第3项为不进行子树分裂的增益，最后一项为对引入新叶子的惩罚项，对树进行了剪枝。

为了提高计算效率，XGBoost可以在GPU上实现决策树的生成，从而加速训练过程。如附录B算法所示，针对叶子节点的分裂方式，GPU先并行计算得到特征的梯度直方图，然后对直方图进行并行扫描，通过并行前缀和操作来计算每个特征和每个分位数的分割增益，从而获得最佳分裂点[18-19]。值得注意的是，单棵树的生成仍然是具有时序依赖性的。但是，在每次迭代中需要计算每个训练样本对于目标函数的一阶和二阶梯度，而GPU在浮点计算速度和内存带宽方面具有巨大优势，在这些指标的计算上也表现出更高的效率。

3.2　特征工程

在训练模型前，合理的特征工程对于预测效果的提升至关重要。区域用电情况往往按照划分的台区进行采集。传统的统计方法只能对单一台区单独建模分析，无法解决区域性的用电负荷预测问题。考虑区域内多台区用电情况的多样性，首先通过*K*-means聚类算法进行台区聚类分析，然后采用XGBoost模型对区域内多个台区的日负荷情况进行统一建模预测。*K*-means聚类算法根据距离相似性将样本集$\boldsymbol{x}$聚类成K个簇$\boldsymbol{C}_i(i=1,2,\cdots,K)$，$\boldsymbol{C}_i$的均值向量$\boldsymbol{\mu}_i$为该簇的聚类中心，迭代过程中每分配一个样本会重新计算当前聚类中心，直至聚类中心不再变化，即误差平方和E最小。

$$\boldsymbol{\mu}_i=\frac{1}{|\boldsymbol{C}_i|}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\boldsymbol{x} \tag{9}$$

$$E=\sum_{i=1}^{K}\sum_{\boldsymbol{x}\in\boldsymbol{C}_i}\|\boldsymbol{x}-\boldsymbol{\mu}_i\|_2 \tag{10}$$

本文选取附录C表C1所示的台区特征作为聚类分析输入数据，通过主成分分析(PCA)进行特征降维后，选择多个聚类数量，并根据肘部法则找到畸变程度得到极大改善的临界点，也即取斜率变化较大的拐点作为效果最好的聚类数量。

针对聚类结果，对于每一类别台区进行模型训练。本文通过人工经验选择了历史负荷数据、天气信息、时间规则3个方面特征作为输入数据，从而预测未来一天的日负荷数据，所选的输入特征如附录C表C2所示。历史负荷数据除了选择前一天和前一周的负荷数据，还对过去一周的负荷和负载率做均值处理，使得输入数据更加平滑。天气信息包括了每个台区每日的平均温度和平均湿度。时间规则信息包含了预测目标对应的星期、日期、月度、季节以及是否属于节假日。为了防止各个连续数据间互相影响，对于时间规则的特征采用独热编码(one-hot)的离散处理形式。

4　算例分析

本文提出的基于CPU-GPU异构计算框架实现的用电负荷预测方案如附录C图C1所示。本方案涉及CPU和GPU之间的任务分配，其设计原则是：复杂的逻辑运算和流程控制任务在CPU上完成，密集型的数值计算任务在GPU上并行完成，同时尽量减少CPU和GPU之间的数据流动开销。因此，在异构计算方案中，CPU负责完成用电信息的获取、预处理和负荷数据的清洗、台区聚类分析、特征构造及生成数据集等环节，并通过OpenMP多线程动态分配计算任务；GPU负责完成用电信息统计数据的并行计算和XGBoost算法的模型训练等环节，GPU的Kernel_1—Kernel_N表示为计算业务进行重排序赋值、并行扫描建立映射、数值计算等操作设计的CUDA内核函数。

算例数据来自深圳市2018年43 254个台区的用电信息。实验测试平台操作系统为Ubuntu 16.08，服务器硬件配置为2台NVIDIA K40C GPU和1台Intel XeonE5 CPU，软件配置为Oracle 11.2，CUDA 9.0。

4.1　用电信息计算效率性能分析

首先，本文在实验平台进行用电信息中的日负荷量和日负荷极值的并行计算，配置的2台GPU分

别负责1个计算任务。日负荷极值的并行计算流程与日负荷量计算大致相同，不同的是数值计算的内核函数改为负荷极值的计算方式，即提取每块电能表一天最大/最小的负荷和电压电流值。算例相关数据表格如表1所示。测试内容A为多卡并行计算，序号1、2用于日负荷量计算，序号1、3、4用于日负荷极值计算。测试内容B为针对3个不同规模数据量算例进行的单卡计算性能测试，序号5由数据表KH_CLDZB_1、CJ_RDJDNL_1组成；序号6、7的数据量分别约为序号5的2、3倍，2个测试的总数据量都已达到GB级别。

表1 数据说明

Table 1 Data explanation

测试内容	序号	表名称	注释	数据记录条数	数据量/MByte
A	1	KH_CLDZB	测量点数据主表	3220269	429.9
	2	CJ_RDJDNL	日冻结电能量	6177487	630.3
	3	CJ_GLQX	运行电能表功率曲线	7028940	449.1
	4	CJ_DYDLQX	运行电能表电压电流曲线	6557788	450.3
B	5	KH_CLDZB_1	日负荷计算测试1数据	3220269	429.9
		CJ_RDJDNL_1		6177487	630.3
	6	KH_CLDZB_2	日负荷计算测试2数据	7777590	1038.3
		CJ_RDJDNL_2		12857237	1311.8
	7	KH_CLDZB_3	日负荷计算测试3数据	10370120	1384.4
		CJ_RDJDNL_3		17142097	1749.1

针对测试内容A，表2比较了CPU-GPU异构计算方案与Oracle数据库存储过程串行计算方案的性能，2种方案的总耗时分别为55.34、655.80 s。从表中结果可看出，相比Oracle数据库计算方案，CPU-GPU异构计算方案虽然牺牲了一定的时间进行数据交互和预处理，但是在GPU端加速了用电信息的业务计算。业务1、2的数值计算部分在数据库方案中分别需要232.24、423.56 s，而在GPU上实现计算仅耗时1.73 s和1.78 s（表2中CPU-GPU异构计算方案业务1和业务2的时间同行统计，表示分别在2台GPU上同时进行并行计算，在最后的总时间中只计入耗时最长的业务计算时间），效率均提高了2个数量级。同时，由于本文采用多线程技术拓展了多个GPU并行计算的流程，在CPU-GPU异构计算方案中业务1和业务2分别在2台GPU上同时进行并行计算，实际总时间只计以单个GPU上耗时最长的业务时间（即业务2），这进一步提高了用电数据的计算效率，使得总体计算流程的加速比达到了11.85倍。表3进一步比较了不同数据规模下单台GPU并行计算的性能情况，测试1—3分别对表1中测试内容B的3组数据进行了日负荷计算，对应的计算总耗时分别为31.53、161.69、225.94 s。由上述分析可知测试1中相同规模的数据在Oracle数据库中计算需要232.24 s，在CPU-GPU异构计算方案中只耗时31.53 s，而在数据量扩大约3倍的测试3中也仅耗时225.94 s，仍小于Oracle数据库计算方案所需的时间，这体现了CPU-GPU异构计算方案在处理计算密集型任务时的优越性。

表2 CPU-GPU异构计算方案与Oracle数据库计算方案性能分析

Table 2 Performance analysis for CPU-GPU heterogeneous computing scheme and Oracle database computing scheme

序号	计算步骤	计算时间/s	
		CPU-GPU异构计算方案	Oracle数据库计算方案
1	数据库读取	18.10	
2	数据预处理	11.63	
3	业务1：日负荷量计算	1.73/1.78	232.24
4	业务2：日负荷极值计算		423.56
5	计算结果写入数据库	23.83	

表3 日负荷计算性能分析

Table 3 Performance analysis for daily load calculation

序号	计算步骤	计算时间/s		
		测试1	测试2	测试3
1	数据库读取	9.79	49.15	70.73
2	数据预处理	7.39	21.61	28.01
3	日负荷量计算	1.74	26.03	33.90
4	计算结果写入数据库	12.61	64.90	93.30

4.2 多台区负荷预测效果性能分析

根据CPU-GPU异构计算方案中日负荷量的计算结果，采用XGBoost模型实现多台区负荷预测。负荷数据来自于2018年深圳市的日用电负荷记录，预测目标为未来一天的日负荷数据，预测时段为2019年1月1日至1月7日。数据集按照80%比例划出训练集，其余为验证集，测试集为预测日期的真实数据。

4.2.1 实验评价指标

为了评估预测方法的性能，评价指标采用均方根误差e_{RMSE}、平均绝对误差e_{MAE}、平均绝对百分比误差e_{MAPE}和决定系数λ_{R2}，计算式分别见式(11)—(14)。

$$e_{\mathrm{RMSE}}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)^2} \tag{11}$$

$$e_{\mathrm{MAE}}=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i-y_i\right| \tag{12}$$

$$e_{\mathrm{MAPE}}=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i-y_i}{y_i}\right|\times 100\,\% \tag{13}$$

$$\lambda_{\mathrm{R2}}=1-\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)^2}{\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2} \tag{14}$$

式中：$\bar{y}$为整体样本的平均值。e_{RMSE}、e_{MAE}和e_{MAPE}描述了预测值和真实值之间的偏差情况，指标越小表示预测效果越好。λ_{R2}描述了模型对数据的拟合程度，指标越接近1表示模型拟合得越好。

4.2.2　数据处理

由于通信错误或者数据丢失等原因，历史负荷信息中有异常数据存在，其将影响预测模型的预测精度。本文通过数据清洗剔除了缺失较多（缺失率大于90%）的台区样本。在清洗后的43 052个有效台区样本中，先对缺失值采用相邻日期数据的平均值进行填充，然后基于正态分布检测出异常数据点，根据文献[20]提供的方法，针对不同的出错原因进行以下几种数据修正方式：①对大事故日负荷或明显负荷曲线异常的日负荷进行剔除或用正常曲线置换；②采用数据横向对比方法消除由于采样错误带来的负荷毛刺；③对于某些时段的异常负荷数据，由现场人员根据经验进行修正。

4.2.3　*K*-means-XGBoost方法的预测结果分析

对预处理后的台区负荷数据进行*K*-means聚类分析，如图2所示，根据肘部法则选择*K*=3作为最佳聚类数。3个类别的台区数量分别为7 767、14 788和20 497。

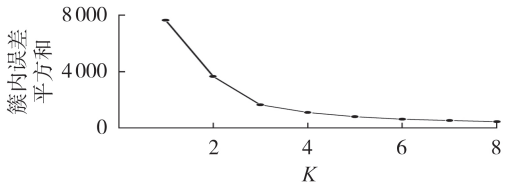

图2　不同聚类数量的误差曲线

Fig.2　Error curve of different numbers of clusters

本文同时选择了长短时记忆神经网络（LSTM）和RF模型进行了负荷预测。LSTM改进了传统的循环神经网络，通过输入门、遗忘门和输出门来决定数据的更新和丢弃，解决了训练中梯度爆炸和梯度消失的问题，可用于学习时间序列长短期依赖信息。RF模型则是一种组合决策树模型，通过有放回的重复采样方式排列组合多个决策树，然后根据多个树的结果进行投票决定最后的结果。利用交叉验证和网格搜索法对XGBoost、LSTM和RF模型参数进行寻优，选定的参数配置如附录C表C3所示。模型的训练都在GPU上进行。LSTM和RF通过Tensorflow搭建，GPU加速了底层的数学运算。XGBoost通过GPU加速构建决策树，提高了叶子节点对特征的分裂选择。

表4给出了3个模型在不同类别台区上的预测效果和总计算时间。在预测精度上，从评价指标可以看出XGBoost模型在3个台区类别上的预测误差都小于LSTM和RF模型，对于数据的拟合程度也优于LSTM和RF模型。图3描述了从每个类别中各随机抽取3个台区的负荷预测情况，其中XGBoost模型的预测结果更接近真实数值曲线。这是由于XGBoost模型对损失函数进行二阶泰勒级数展开，优化过程使用了一阶和二阶导数信息进行更新迭代，使模型训练更充分。在计算效率上，借助GPU加速的XGBoost模型的计算耗时是最少的，只需要41.786 s就完成了模型训练。这是由于本身树形结构的生成效率就高很多，而GPU并行计算加速了贪心算法对叶子节点分裂特征的选择，进一步加快了算法的迭代过程。综合分析，基于XGBoost模型的电力负荷预测在预测精度和计算时间2个方面的指标都优于其他方法，表现出良好的性能。

表4　负荷预测模型性能分析

Table 4　Performance analysis for load forecasting model

模型	台区类别	e_{RMSE} / MW	e_{MAE} / MW	e_{MAPE} / %	λ_{R2}	总计算时间 / s
XGBoost	1	25.535	13.967	3.980	0.966	41.786
	2	134.409	40.135	8.560	0.947	
	3	29.225	14.288	7.398	0.947	
LSTM	1	29.076	16.197	4.399	0.956	1 586.361
	2	158.027	54.615	9.924	0.929	
	3	32.462	16.610	9.451	0.935	
RF	1	27.243	14.383	4.128	0.959	503.153
	2	152.485	41.760	8.927	0.932	
	3	29.690	14.301	7.646	0.940	

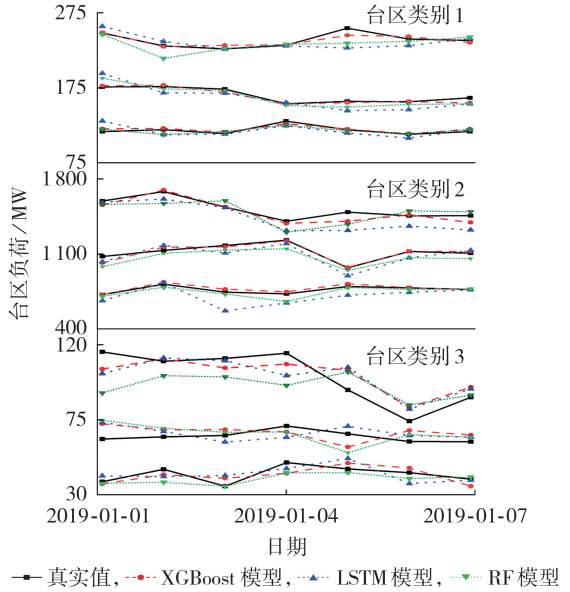

图3　台区负荷预测结果

Fig.3　Results of station load forecasting

5 结论

本文研究了CPU-GPU异构计算框架下的高性能用电负荷预测方法，将GPU应用在了电力负荷预测的数据获取和模型训练上，实现了GPU并行加速的用电负荷数据计算和台区负荷预测。算例结果表明，结合OpenMP多线程技术的GPU并行计算方案对业务处理的效率更高，结合*K*-means聚类分析的基于XGBoost算法的多台区负荷预测方案在预测精度和计算效率上都表现出良好的性能。此外，在实际生产环境中，本文提出的异构计算方案综合考虑了海量用电数据的统计实时性和挖掘充分性的要求，在更大规模的用电计算业务和短期负荷预测应用中有较好的应用前景，为新一代计算业务数据中台的硬件选型提供了参考和借鉴。

附录见本刊网络版(http://www.epae.cn)。

参考文献：

[1] 张素香，赵丙镇，王风雨，等. 海量数据下的电力负荷短期预测[J]. 中国电机工程学报，2015，35(1)：37-42.
ZHANG Suxiang，ZHAO Bingzhen，WANG Fengyu，et al. Short-term power load forecasting based on big data[J]. Proceedings of the CSEE，2015，35(1)：37-42.

[2] CECATI C，KOLBUSZ J，RÓŻYCKI P，et al. A novel RBF training algorithm for short-term electric load forecasting and comparative studies[J]. IEEE Transactions on Industrial Electronics，2015，62(10)：6519-6529.

[3] YANG Y L，CHE J X，LI Y Y，et al. An incremental electric load forecasting model based on support vector regression[J]. Energy，2016，113：796-808.

[4] 吴潇雨，和敬涵，张沛，等. 基于灰色投影改进随机森林算法的电力系统短期负荷预测[J]. 电力系统自动化，2015，39(12)：50-55.
WU Xiaoyu，HE Jinghan，ZHANG Pei，et al. Power system short-term load forecasting based on improved random forest with grey relation projection[J]. Automation of Electric Power Systems，2015，39(12)：50-55.

[5] LI Guangye，LI Wei，TIAN Xiaolei，et al. Short-term electricity load forecasting based on the XGBoost algorithm[J]. Smart Grid，2017，7(4)：274-285.

[6] 史佳琪，张建华. 基于多模型融合Stacking集成学习方式的负荷预测方法[J]. 中国电机工程学报，2019，39(14)：4032-4042.
SHI Jiaqi，ZHANG Jianhua. Load forecasting based on multi-model by Stacking ensemble learning[J]. Proceedings of the CSEE，2019，39(14)：4032-4042.

[7] 刘波，秦川，鞠平，等. 基于XGBoost与Stacking模型融合的短期母线负荷预测[J]. 电力自动化设备，2020，40(3)：147-153.
LIU Bo，QIN Chuan，JU Ping，et al. Short-term bus load forecasting based on XGBoost and Stacking model fusion[J]. Electric Power Automation Equipment，2020，40(3)：147-153.

[8] 陈振宇，刘金波，李晨，等. 基于LSTM与XGBoost组合模型的超短期电力负荷预测[J]. 电网技术，2020，44(2)：614-620.
CHEN Zhenyu，LIU Jinbo，LI Chen，et al. Ultra short-term power load forecasting based on combined LSTM-XGBoost model[J]. Power System Technology，2020，44(2)：614-620.

[9] VON KRANNICHFELDT L，WANG Y，HUG G. Online ensemble learning for load forecasting[J]. IEEE Transactions on Power Systems，2021，36(1)：545-548.

[10] ZHOU G，FENG Y J，BO R，et al. GPU-accelerated batch-ACPF solution for *N*-1 static security analysis[J]. IEEE Transactions on Smart Grid，2017，8(3)：1406-1416.

[11] 张宸赓，许寅，陈颖，等. 基于GPU加速的大电网*N*-1故障扫描批量计算方法[J]. 电力自动化设备，2020，40(8)：167-176.
ZHANG Chengeng，XU Yin，CHEN Ying，et al. Batch computing method for *N*-1 fault scanning of large power grid based on GPU acceleration[J]. Electric Power Automation Equipment，2020，40(8)：167-176.

[12] CHEN X M，REN L，WANG Y，et al. GPU-accelerated sparse LU factorization for circuit simulation with performance modeling[J]. IEEE Transactions on Parallel and Distributed Systems，2015，26(3)：786-795.

[13] KACZMARSKI K，RUDNY T. MOLAP cube based on parallel scan algorithm[C]// 16th East-European Conference on Advances in Databases and Information Systems. Poznan，Poland：ADBIS，2012：125-138.

[14] MALIK M，RIHA L，SHEA C，et al. Task scheduling for GPU accelerated hybrid OLAP systems with multi-core support and text-to-integer translation[C]//2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum. Shanghai，China：IEEE，2012：1987-1996.

[15] HE B S，LU M，YANG K，et al. Relational query coprocessing on graphics processors[J]. ACM Transactions on Database Systems，2009，34(4)：1-39.

[16] NVIDIA Corporation. NVIDIA CUDA C programming guide[EB/OL]. [2020-08-07]. http://docs.nvidia.com/cuda/cuda-c-programming-guide/.

[17] CHEN T Q，GUESTRIN C. XGBoost：a scalable tree Boosting system[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco，USA：ACM，2016：785-794.

[18] BHATTACHARYA S，SIVA RAMA KRISHNAN S，MADDIKUNTA P K R，et al. A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU[J]. Electronics，2020，9(2)：219.

[19] MITCHELL R，FRANK E. Accelerating the XGBoost algorithm using GPU computing[J]. PeerJ Computer Science，2017，3(1)：127.

[20] 贺蓉，曾刚，姚建刚，等. 天气敏感型神经网络在地区电网短期负荷预测中的应用[J]. 电力系统自动化，2001，25(17)：32-35，52.
HE Rong，ZENG Gang，YAO Jiangang，et al. Application of weather sensitivity neural network model in short-term load forecasting on area[J]. Automation of Electric Power Systems，2001，25(17)：32-35，52.

作者简介：

赵嘉豪

赵嘉豪(1995—)，男，江苏江阴人，硕士研究生，主要研究方向为高性能计算在电力系统中的应用(**E-mail**：zhaojiahao@seu.edu.cn)；

周 赣(1978—)，男，江苏丹阳人，副教授，博士，主要研究方向为高性能计算在电力系统中的应用、智能配用电技术(**E-mail**：zhougan2002@seu.edu.cn)。

(**编辑 王欣竹**)

(下转第198页 continued on page 198)

Expansion planning of transmission network with high proportion of hydropower considering guidance of marketized electricity price signal

WANG Fuyang[1], LIU Youbo[1], XU Weiting[2], GOU Jing[2], LIU Fang[2], SU Yunche[2], LIU Junyong[1]
(1. College of Electrical Engineering, Sichuan University, Chengdu 610065, China;
2. Economic and Technological Research Institute of State Grid Sichuan Electric Power Company, Chengdu 610041, China)

Abstract: The traditional power grid expansion planning model is difficult to reflect the quantitative impact of global supply and demand situation on investment income of power grid, willingness of market entities and social benefit under the competitive market mode for a period of time in the future, and the network congestion and transaction surplus problems in transmission network with high proportion of hydropower caused by the mismatch between the capacity of hydropower transmission section and the difference between power flows in wet and dry seasons are increasingly serious, for which, an expansion planning model of transmission network under the scenario with high proportion of hydropower is proposed considering guidance of marketized electricity price signal. The upper layer model takes the maximum annual line investment income as its objective function to form the planning decision lines. The lower layer model adopts the improved k-means clustering algorithm to obtain the typical annual operation mode of transmission network with high proportion of hydropower, quantifies load increment and network utilization rate after line planning based on the clearing results of market transaction and the response characteristic of regional load, and guides the expansion planning of power grid with the marketized electricity price signal to promote regional hydropower consumption. KKT optimal conditions are used to realize the coupling of the upper and lower layer models, and the bi-level programming problem is transformed into a mixed integer programming problem for solution. A practical power grid with high proportion of hydropower in southwest China is taken as an example, and the impact of market signal on the expansion planning results under high proportion of hydropower integration is verified by comprehensive comparison of congestion surplus, investment income and proportion of hydropower consumption among different planning methods.

Key words: expansion planning; hydropower consumption; congestion surplus; price elasticity; locational marginal price

(上接第146页 continued from page 146)

High-performance electricity load forecasting under CPU-GPU heterogeneous computing framework

ZHAO Jiahao[1], ZHOU Gan[1], HUANG Li[1], LU Chunyan[2], TAO Xiaofeng[2], FENG Yanjun[1]
(1. School of Electrical Engineering, Southeast University, Nanjing 210096, China;
2. NARI Group Corporation / State Grid Electric Power Research Institute, Nanjing 211106, China)

Abstract: With the rapid development of power grid, the electricity information acquisition system faces great challenges in the data computing business. Recently, GPU (Graphics Processing Unit) has become important issues in high-performance computing problems due to its superior performance on floating-point computing speed and memory bandwidth. GPU has been successfully applied to scientific computing fields such as power system caculating analysis. When facing power load forecasting problem based on artificial intelligence methods, most of the past researches only considered training of GPU-accelerated forecasting model, but not applied to data acquisition and calculation. A high-performance electricity load forecasting solution under the CPU-GPU (Central Processing Unit-Graphics Processing Unit) heterogeneous computing framework is proposed. Firstly, with the help of CUDA (Compute Unified Device Architecture) and multi-threading technology, power data is computed in parallel by multi-GPUs. Afterwards, with the help of cluster analysis, multi-station load forecasting is completed based on XGBoost algorithm, where GPU accelerates the model training calculation. Finally, through the case analysis of the electricity information of 43 254 stations in Shenzhen, the efficiency and applicability of the proposed method are verified.

Key words: electricity information acquisition system; electric load forecasting; GPU; heterogeneous computing; XGBoost

附录 A

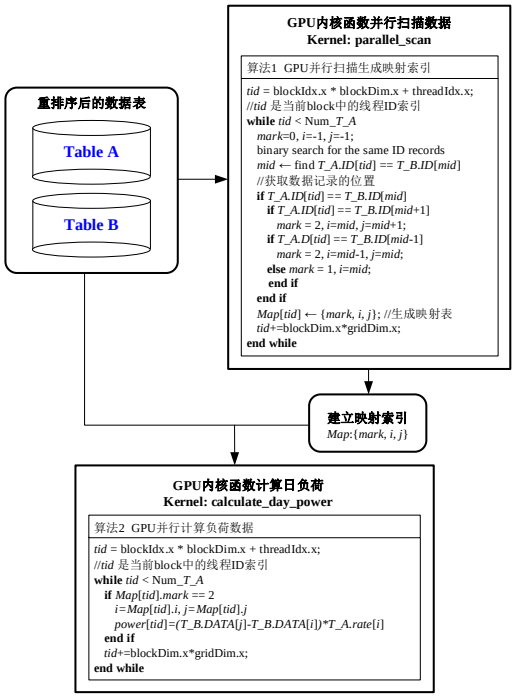

图 A1　GPU 并行计算日负荷量示意图

Fig.A1　GPU-based parallel daily load calculation

附录 B

算法　GPU 实现的决策树构建

输入： X_1, X_2...X_p $\in X$

输出： *tree*

tree ← {}

expand_queue ← InitRoot()

while expand_queue is not empty **do**

　　expand_entry ← expand_queue.pop()

　　tree.insert(expand_entry)

　　//在 GPU 上处理训练样本

　　for i in 0…p do

　　　//根据之前的分裂节点对样本特征进行排序

　　　RepartitionInstances(expand_entry, X_i)

　　　//并行扫描建立梯度直方图

　　　BuildPartialHistograms(expand_entry, X_i, g_i)

　　end for

　　//合并 GPU 上所有特征的直方图

　　AllReduceHistograms(expand_entry)

　　//找到左右子树的最佳分裂点

　　left_expand_entry ← EvaluateSplit (expand_entry.left_histogram)

　　right_expand_entry ← EvaluateSplit (expand_entry.right_histogram)

　　expand_queue.push(left_expand_entry)

　　expand_queue.push(right_expand_entry)

end while

附录 C

表 C1　台区负荷聚类分析所用特征

Table C1　Features for cluster analysis of station load

序号	台区信息字段	说明
1	EDRL	额定容量
2	YXRL	运行容量
3	PJGLYS	平均功率因数
4	GDL	供电量
5	SDL	售电量
6	DLBPHL	电流不平衡率
7	DYBPHL	电压不平衡率
8	XSL	线损率

表 C2　负荷预测模型输入特征

Table C2　Input features for load forecasting model

影响因素	特征	说明
历史负荷数据	前一天日负荷	
	前一周日负荷	
	过去一周日负荷均值	
	过去一周负载率均值	
天气信息	平均温度	
	平均湿度	
时间规则	星期	标记为 1—7 的整数
	月度	标记为 1—12 的整数
	日期	标记为 1—31 的整数
	季节	标记为 1—4 的整数
	是否属于节假日	如果为节假日，标记为 1，反之则为 0

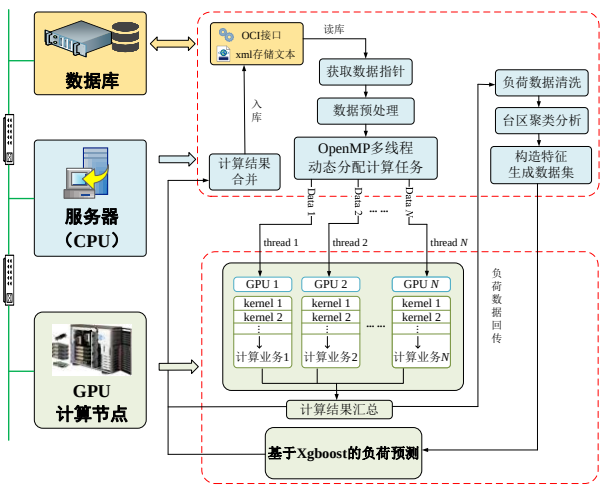

图 C1　异构计算框架下的计算方案

Fig.C1　Computing solution under the framework of heterogeneous computing

表 C3　预测模型参数配置

Table C3　Parameter configuration of forecasting model

算法	超参设置
XGBoost	树深度 8，学习率 0.15，树的数目 300，最小叶子节点样本权重 3，随机采样的比例 0.7，叶子节点分裂最小损失 0.1
LSTM	隐藏层 2 层，每层为 LSTM 和 Dropout，神经元数量 128，输出层为 1 层全连接层，训练次数 50
RF	叶子节点最小样本数量 1，节点划分最小样本数量 2，树的数目 300