基于 CUDA 技术的海量电力负荷曲线聚类算法

吴　霜[1]，季　聪[2]，孙国强[3]
（1. 国网江苏省电力有限公司经济技术研究院，江苏 南京 210008；
2. 江苏方天电力技术有限公司，江苏 南京 211102；
3. 可再生能源发电技术教育部工程研究中心（河海大学），江苏 南京 210098）

摘　要：随着用电信息采集、负荷控制等系统中用户负荷数据的爆炸式增长，传统计算框架与方法在处理海量用户负荷聚类、开展负荷特性分析等业务时面临着巨大的计算压力。着眼于计算精度日益提高、计算能力日渐强大的图形处理单元（graphic process unit，GPU），基于 Nvidia 的统一计算设备架构（compute uniform device architecture，CUDA）提出了一种负荷曲线快速并行 *K*-means 聚类算法，采用距离计算并行化、曲线数统计并行化、线程块分配合理化等多个并行加速策略，极大地提升了用户负荷曲线的聚类速度。多个测试算例表明，文中提出的基于 CUDA 的 *K*-means 电力负荷曲线聚类算法加速比高，适应性强，是解决海量负荷曲线聚类问题的好方法。

关键词：GPU；CUDA；并行计算；海量数据；*K* 均值聚类；电力负荷曲线
中图分类号：TM744　　**文献标志码**：A　　**文章编号**：2096-3203（2018）04-0065-06

0　引言

随着电力改革不断推进和电力用户需求的多元化发展，电力企业越来越关注用户负荷、电量数据分析，希望通过数据挖掘来提高服务质量、创造增值效益。而用电信息采集、负荷控制等信息化系统的建设与应用，积累了海量的电力用户负荷数据，为用户数据分析提供了良好的数据基础。

一直以来，电力专家和学者们都持续不断地开展着负荷特性、负荷预测等数据挖掘课题的研究[1-3]，而负荷聚类分析是开展负荷特性分析、负荷建模等工作的基础[4]。但随着用户侧数据量的爆炸式增长，用户负荷聚类分析面临着对象众多、数据量大等问题。Nvidia 基于图形处理单元（graphic process unit，GPU）的统一计算设备架构（compute uniform device architecture，CUDA）技术给出了令人鼓舞的解决方案，它通过大量线程并发机制，极大地加快了海量数据计算与分析的速度[5]。目前，CUDA 技术已在电力系统谐波分析[6]、暂态稳定[7-8]、状态估计[9]等领域得到了广泛的应用。

目前已有学者将 CUDA 技术用于加速文献聚类[10-11]、图形聚类[12]等，但在海量电力负荷曲线聚类中，CUDA 技术的未见应用。因此，本文基于电力负荷曲线的特征，针对 CUDA 的技术特性，深入研究负荷曲线聚类算法的并行处理机制，采用待划分数据与聚类中心的距离计算并行化、类别变化曲线数统计并行化、线程块分配合理化等多个并行加速策略，极大地提升了用户负荷曲线的聚类速度，取得良好的聚类结果和加速性能。

收稿日期：2018-03-20；修回日期：2018-04-08
基金项目：国家自然科学基金资助项目（51277052）

1　*K*-means 聚类算法

K-means 聚类算法原理简单，可操作性强，是目前应用最为广泛的聚类方法之一。它首先随机选定一组初始聚类中心，经迭代使得聚类中心保持类间独立、类内紧密，迭代期间不断更新聚类子集和聚类中心。目前 *K*-means 聚类算法在图形分割、流量监测、负荷聚类等领域得到了广泛的应用[13-15]。

K-means 聚类算法的输入为包含 N 个数据项的待划分数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，其中 $\boldsymbol{x}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$，$D$ 为数据的维度。输出则为 K 个聚类集合 $\boldsymbol{C}=\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_K\}$，$K$ 为用户自定义的聚类中心个数。*K*-means 的目标是使属于集合 $\boldsymbol{c}_K$ 的数据 $\boldsymbol{x}_i$ 到其所属聚类中心 $\boldsymbol{m}_k$ 的距离最小，即：

$$f(\boldsymbol{x}_i)=\sum_{\boldsymbol{x}_i\in\boldsymbol{c}_K}d(\boldsymbol{x}_i,\boldsymbol{m}_k) \tag{1}$$

$$\boldsymbol{m}_k=\sum_{\boldsymbol{x}_i\in\boldsymbol{c}_K}\boldsymbol{x}_i/n_k \tag{2}$$

式中：n_k 为集合 $\boldsymbol{c}_k$ 所包含的数据个数。*K*-means 聚类算法的主要步骤如下。

Step 0：数据准备阶段。准备好待划分数据集 $\boldsymbol{X}$，选择合适的 K，设定算法停止迭代判定条件（一般为最大迭代次数或类别变化曲线条数占比）。

Step 1：初始聚类中心选择。一般采用随机选择法选取聚类中心，目前也有大量学者提出了各种改进方法[16-17]。

Step 2：数据集归类计算。采用欧式距离法计算

各数据到聚类中心的距离，将各数据划分到距离最短的聚类中心。距离计算公式如下：

$$d(\boldsymbol{x}_i,\boldsymbol{m}_k)=\sqrt{\sum_{j=1}^{D}(x_{ij}-m_{ij})^2} \quad (3)$$

Step 3：重新计算聚类中心。依据 Step 2 中的数据分类，重新计算各聚类中心。

Step 4：判断是否满足迭代结束条件，若是，则退出迭代，输出聚类结果；否则返回 Step 2。

2 *K*-means 的 CUDA 加速策略

2.1 CUDA 的加速原理

CUDA 技术由 Nvidia 公司于 2007 年 6 月提出，与之前的通用图形处理单元（general purpose GPU，GPGPU）不同，它采用标准 C 语言进行编码，并且开放了大量基于 C 语言的应用程序接口（application programming interface，API），极大地方便了基于 GPU 的并行编码，为算法开发与设计提供了一种高性能、易维护的 GPU 并行计算平台。

CUDA 技术中，GPU 被组织成 3 层计算资源：线程（Thread）、线程块（Block）、线程网格（Grid），其组织结构如图 1 所示。

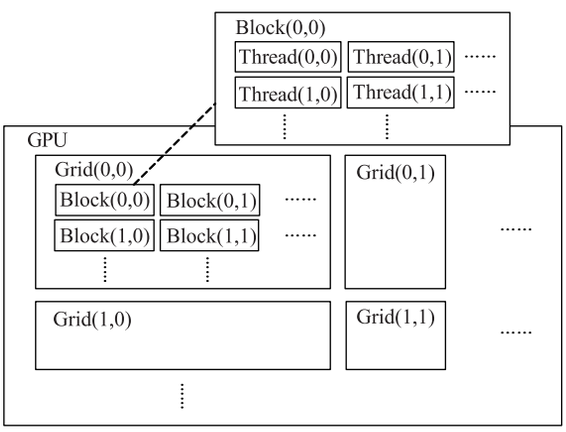

图 1 CUDA 的线程组织形式

Fig.1 Organization mode of threads in CUDA

CUDA 通过将串行程序中相对独立的简单计算操作交由 GPU 中的大量线程并发执行，从而使计算速度得到大幅度的提升。*K*-means 聚类算法的核心部分在于计算待划分数据与聚类中心的距离，该计算步骤恰恰属于计算过程相对简单独立、重复性高的操作，可以交由 GPU 进行并行加速计算。

2.2 *K*-means 的加速策略

从 *K*-means 聚类算法的原理可知，其计算量最大的部分在于待划分数据与聚类中心的距离计算，该步骤具备高度的可拆分性，因此可采用 CUDA 对其进行加速处理。

加速策略 1：待划分数据与聚类中心的距离计算并行化。将 N 个待划分数据分别指派给 N 个线程，各线程分别计算待划分数据与聚类中心的距离，从而实现距离计算的高度并行。

加速策略 2：类别变化曲线数统计的并行化。统计类别变化曲线数时一般采用累加法，采用 CUDA 加速时，可考虑线程块内变化数目叠加后存于线程块内的寄存器，再通过多个线程块寄存器数据并行累加得到总变化数。线程块寄存器的读写速度远远快于内存，因此可以节省大量的数据传输和读写时间。另外，块间数据累加时采用二分并行叠加，计算复杂度可由 N 降低为 $\log_2(N)$。

加速策略 3：线程块分配最优化。由于 GPU 的一个线程束包括 32 个线程，因此无论计算能力高低，目前线程块所包含的线程数均为 32 的倍数，在线程块的分配上，需要根据聚类数据量的大小，合理分配线程块的大小和线程块的个数，保证每条负荷曲线有对应的线程处理。本文采用的 GPU 为 Nvidia GeForce GTX960，其线程块最大可处理 1024 个线程，在数据量较小时（例如 4.1 的 2000 条负荷曲线），线程块大小可设置为 64、128，在数据量较大时（例如 4.2 的 40 000 条负荷曲线），线程块大小可设置为 1024。

通过上述 3 个加速策略，可显著提升 *K*-means 聚类算法的计算效率，这对于海量数据聚类具有极大的技术价值。

3 基于 CUDA 的 *K*-means 聚类算法实现步骤

基于 CUDA 的 *K*-means 聚类算法实现步骤如图 2 所示。

Step 1：读入用户负荷数据，设定聚类个数、收敛条件等参数，本文以变化类别曲线占比 ε 为收敛判据，$\varepsilon < 0.001$ 则认为达到收敛要求。

Step 2：将用户负荷曲线从内存复制到 GPU，并随机选定 K 条负荷曲线作为初始聚类中心。

Step 3：根据用户负荷曲线数量，采用加速策略 3 设定线程块的大小、以及线程块的数量。目前 Maxwell 架构的 GPU 线程块支持最大线程数为 1024，因此尽可能选择线程块大小为 1024，线程块数量为 $1+N/1024$ 取整。

Step 4：N 个线程与 N 条负荷曲线一一对应，N 个线程分别计算 N 条负荷曲线与 K 个聚类中心的距离。

Step 5：根据距离大小将负荷曲线归入最近的聚类中心，并计算生成新的聚类中心。

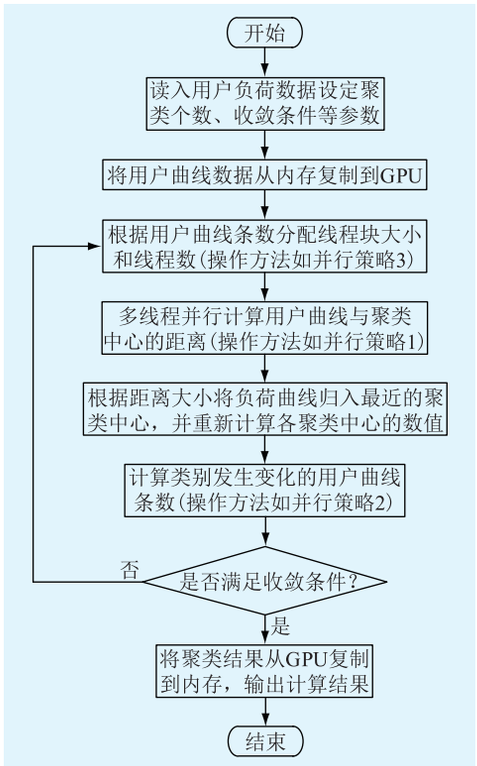

图 2　基于 CUDA 的 *K*-means 聚类算法实现步骤

Fig.2　Flow chart of CUDA based *K*-means clustering

Step 6:采用加速策略 2 统计类别发生变化的负荷曲线条数,并计算变化条数占比 ε ,若 $\varepsilon < 0.001$,则执行 Step 7,否则转到 Step 4。

Step 7:将聚类结果从 GPU 复制到内存,并输出。

4　算例分析

以江苏电网企业用户典型负荷日数据为例,进行聚类分析,验证本文提出的并行聚类算法的效率。

4.1　基本测试

首先以江苏电网 2000 个企业用户的典型负荷日数据为例,进行聚类分析。计算机 CPU 为 Intel Core I5-4590 @ 3.3GHz,GPU 为 NVIDIA GeForce GTX960,运行环境为 Win 7,编译环境为 Visual Studio 2010。算法参数方面:设定聚类数 $K=4$,收敛条件 $\varepsilon < 0.001$。

串行方法迭代 13 次,计算时间为 0.092 s,并行方法迭代次数 12 次,计算时间为 0.302 s,计算速度反而降低。计算结果表明,串行方法与并行方法得到了相同的聚类结果,如图 3 所示,其中负荷曲线均已归一化处理。

2000 条负荷曲线被划分为 4 类,各类曲线条数如图 4 所示。

(1) 全天型负荷。全天 24 h 负荷差异较小,这类曲线有 941 条。该负荷特性一般出现在三班倒的轻工、电子、食品加工等行业。

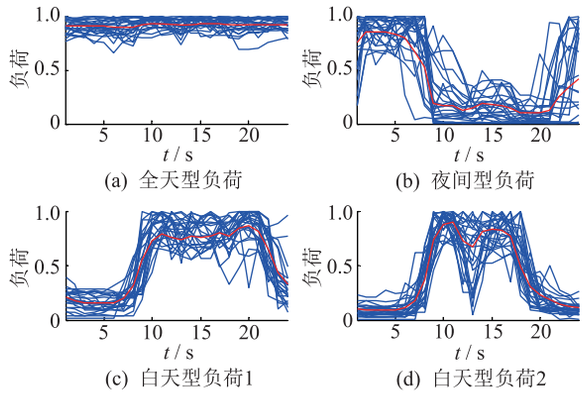

图 3　负荷曲线聚类结果

Fig.3　Clustering results of power load curves

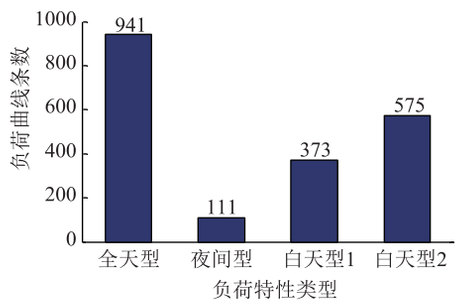

图 4　各负荷特性类型所含曲线条数

Fig.4　Number of all four type of load characteristics

(2) 夜间型负荷。夜间负荷水平较高,白天反之,这类曲线有 111 条。该负荷特性一般出现在冶金、水泥等大型制造行业。

(3) 白天型负荷 1。白天负荷较高,夜间反之,这类曲线有 373 条。该负荷特性一般出现在计算机软件、商业等行业。

(4) 白天型负荷 2。与白天型负荷 1 的区别在于中午会发生短时的负荷下降,这类曲线有 575 条。该负荷特性一般出现在轻工、餐饮等行业。

4.2　性能测试

为了说明 4.1 节遇到的并行方法计算时间反而增加的问题,仍然以 4.1 节的江苏电网 2000 条企业典型负荷曲线为例,进行算例分析。测试手段为不断增加聚类个数 K,串行方法与并行方法计算时间及迭代次数如图 5、6 所示。由图 5—6 可以得出以下结论:

(1) 在聚类中心个数为 4,10 时,计算量较小,而并行方法由于增加了数据传输(负荷数据自内存拷贝 GPU、聚类结果自 GPU 拷贝至内存)和线程同步时间,导致计算时间反而有所增加;

(2) 但随着聚类中心个数的增加,计算量逐渐

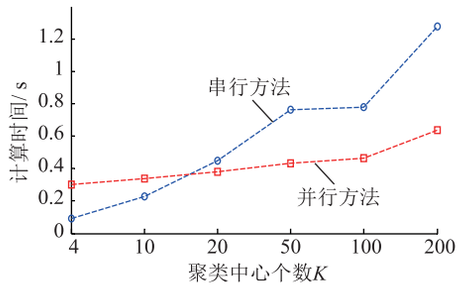

图 5 计算时间比较

Fig.5 Comparison of computing time

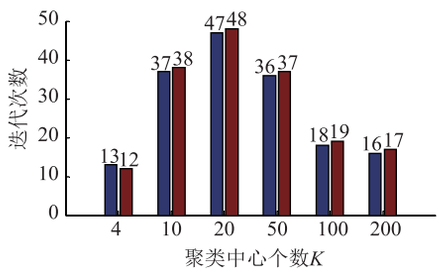

图 6 迭代次数比较

Fig.6 Comparison of iteration times

增加,并行方法的速度优势逐渐体现,在 $K=200$ 时,加速比为 2.01;

(3) 在迭代次数方面,串行方法与并行方法基本相当,并行方法由于 GPU 的计算精度为 float 类型(而 CPU 为 double 类型),因此迭代次数略增(基本增加 1 次)。

很显然,2000 条负荷曲线的聚类分析远远称不上是海量负荷数据,也不足以体现并行方法的加速性能,因此本文选取了江苏电网 4.1 万用户典型负荷曲线,进行更大规模的测试与分析,串行方法与并行方法的计算时间及加速比如图 7 所示。

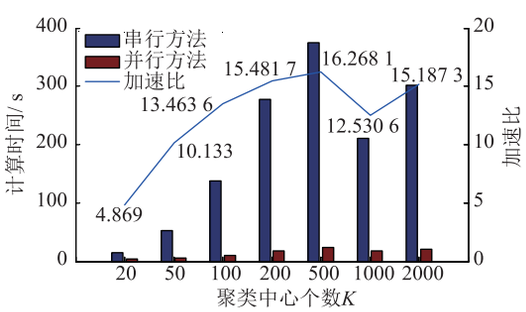

图 7 计算时间及加速比($N=41\ 000$,K 变化)

Fig.7 Computing time and speed-up ratio ($N=41\ 000$, K changes)

在数据量较大时,并行方法大幅度地提高了 K-means 聚类算法的计算效率,便最高加速比达到了 16.2681 倍,可见基于 CUDA 的并行 K-means 聚类算法加速效果显著。需要说明的是:$K=1000$ 和 $K=2000$ 时计算时间与加速比均小于 $K=500$ 时,这主要是因为迭代次数的减少($K=1000$ 时迭代 33 次,$K=2000$ 时 21 次,而 $K=500$ 时 101 次)。

保持聚类中心个数 $K=500$ 不变,负荷曲线数量从 10 000 增加到 200 000 条,进行聚类计算,串行方法与并行方法的计算时间及加速比如图 8 所示。

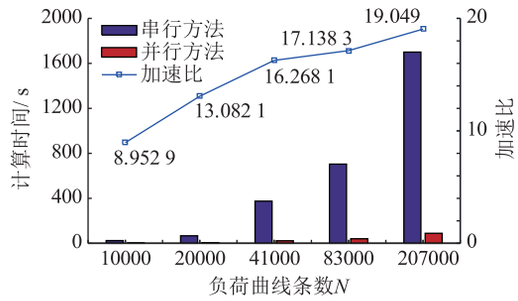

图 8 计算时间及加速比($K=500$,N 变化)

Fig.8 Computing time and speed-up ratio ($K=500$, N changes)

随着负荷曲线条数的增加,K-means 聚类算法的计算量成倍增加,而 GPU 的加速比也在持续增加,在 $N=207\ 000$ 时,加速比达到了 19.049,可见本文提出的并行方法计算速度快、适用能力强,极大地提升了海量负荷数据背景下的电力用户负荷曲线聚类分析效率。

5 结语

在电力体制改革和售电市场放开的大环境下,电力用户负荷特性分析将得到前所未有的关注,但用户数据数量巨大,负荷分析计算量过大。在这样的背景下,本文提出了基于 CUDA 技术的并行 K-means 聚类算法,并以江苏电网企业用户典型负荷曲线为例,进行了多组算例测试与分析,结果表明本文提出的并行方法加速比高、适应性强,可以作为海量负荷曲线聚类分析的有力工具。

参考文献:

[1] 张翠芝, 智 明. 泰州电网负荷特性分析及负荷预测[J]. 江苏电机工程, 2011, 30(4): 45-47.
ZHANG Cuizhi, ZHI Ming. Load characteristics and load forecasting of Taizhou Power Grid[J]. Jiangsu Electrical Engineering, 2011, 30(4): 45-47.

[2] 黄 伟,陈 雪,林怀德,等. 考虑光伏不确定性的配电网负荷特性概率评估[J]. 广东电力,2018,31(6):84-90.
HUANG Wei, CHEN Xue, LIN Huaide, et al. Evaluation on load characteristic probability of power distribution network considering photovoltaic uncertainty [J]. Guangdong Electric Power,2018,31(6):84-90.

[3] 刘亚南, 卫志农, 季 聪,等. 基于 D-S 证据理论的母线负荷预测[J]. 江苏电机工程, 2014, 33(5): 21-24, 27.
LIU Yanan, WEI Zhinong, JI Cong, et al. Bus load forecasting based on D-S evidence theory[J]. Jiangsu Electrical Engineer-

ing, 2014, 33(5): 21-24, 27.

[4] 颜庆国, 薛溟枫, 范　洁,等. 有序用电用户负荷特性分析方法研究[J]. 江苏电机工程, 2014, 33(6): 48-50, 54.
YAN Qingguo, XUE Mingfeng, FAN Jie, et al. Load property analysis method for demanders participating orderly power utilization[J]. Jiangsu Electrical Engineering, 2014, 33(6): 48-50, 54.

[5] GREEN R C, WANG Lingfeng, ALAM M. Applications and trends of high performance computing for electric power systems focusing on smart gid [J]. IEEE Transactions on Smart Grid, 2013, 4(2): 922-931.

[6] 韩志伟, 刘志刚, 鲁晓帆,等. 基于 CUDA 的高速并行小波算法及其在电力系统谐波分析中的应用[J]. 电力自动化设备, 2010, 30(1): 98-101, 105.
HAN Zhiwei, LIU Zhigang, LU Xiaofan, et al. High-speed parallel wavelet algorithm based on CUDA and its application in power system harmonic analysis[J]. Electric Power Automation Equipment, 2010, 30(1): 98-101, 105.

[7] 江　涵, 江全元. 基于 GPU 计算平台的大规模电力系统暂态稳定计算[J]. 电力系统保护与控制, 2013, 41(4): 13-20.
JIANG Han, JIANG Quanyuan. A parallel transient stability algorithm for large-scale power system based on GPU platform [J]. Power System Protection and Control, 2013, 41(4): 13-20.

[8] LIAO Xiaobing, WANG Fangzong. Parallel computation of transient stability using symplectic gauss method and GPU [J]. IEEE Generation, Transmission & Distribution, 2016, 10(15): 3727-3735.

[9] HADIS K, VENKATA D. Extended kalman filter-based parallel dynamic state estimation[J]. IEEE Transactions on Smart Grid, 2015, 6(3): 1539-1549.

[10] 戴　涛, 杨　洲, 方　勇,等. 基于 CUDA 的 *K*-means 文档聚类算法并行优化[J]. 计算机工程与设计, 2013, 34(11): 4032-4036, 4071.
DAI Tao, YANG Zhou, FANG Yong, et al. Parallel optimization algorithm for *K*-means document clustering based on CUDA[J]. Computer Engineering and Design, 2013,34(11): 4032-4036,4071.

[11] 谢　成,金涌涛,胡叶舟,等. 基于相关系数分析的配电网单相接地故障研判方法与试验研究[J]. 浙江电力,2017, 36(3):17-23.
XIE Cheng,JIN Yongtao,HU Yezhou,et al. Diagnosis and experimental research on single-phase-to-earth fault of distribution networks based on correlation [J]. Zhejiang Electric Power,2017,36(3):17-23.

[12] 张　宇, 刘　坡, 杨敏华,等. 基于 GPU 的二部图联合聚类并行算法研究[J]. 地理与地理信息科学, 2013, 29(4): 99-103, 108.
ZHANG Yu, LIU Po, YANG Minhua, et al. Accelerating bipartite graph clustering based on GPU[J]. Geography and Geo-Information Science, 2013, 29(4): 99-103, 108.

[13] 曹志广, 陈　玮, 马如豹. *K*-均值和最大加权熵在彩色图像分割中的应用[J]. 计算机工程与应用, 2012, 48(21): 174-177.
CAO Zhiguang, CHEN Wei, MA Rubao. Application of *K*-means and maximum weighted entropy on color image segmentation[J]. Computer Engineering and Applications, 2012, 48(21): 174-177.

[14] 林成虎, 李晓东, 金　键,等. 基于 W-*K*-means 算法的 DNS 流量异常检测[J]. 计算机工程与设计, 2013, 34(6): 2104-2108.
LIN Chenghu, LI Xiaodong, JIN Jian, et al. DNS traffic anomaly detection based on W-*K*-means algorithm [J]. Computer Engineering and Design, 2013, 34(6): 2104-2108.

[15] 陈　凡, 刘海涛, 黄　正,等. 基于改进 *K*-均值聚类的负荷概率模型[J]. 电力系统保护与控制, 2013, 41(22): 128-133.
CHEN Fan, LIU Haitao, HUANG Zheng, et al. Probabilistic load model based on improved *K*-means clustering algorithm [J]. Power System Protection and Control, 2013, 41(22): 128-133.

[16] 王秀芳, 王　岩. 优化 *K* 均值随机初始中点的改进算法[J]. 化工自动化及仪表, 2012, 39(10): 1302-1304.
WANG Xiufang, WANG Yan. Three improved algorithms for optimizing of randomly-initiated *K*-means midpoints [J]. Control and Instruments in Chemical Industry, 2012, 39(10): 1302-1304.

[17] 谢娟英, 郭文娟, 谢维信,等. 基于样本空间分布密度的初始聚类中心优化 *K*-均值算法[J]. 计算机应用研究, 2012, 29(3): 888-892.
XIE Juanying, GUO Wenjuan, XIE Weixin, et al. *K*-means clustering algorithm based on optimal initial centers related to pattern distribution of samples in space[J]. Application Research of Computers, 2012, 29(3): 888-892.

作者简介:


吴　霜

吴　霜(1989—),女,硕士,工程师,从事电力系统规划与建设工作(E-mail:ws_nj025@163.com);

季　聪(1988—),男,硕士,工程师,从事电力系统分析与控制、电力大数据技术相关工作(E-mail:jcxx01@163.com);

孙国强(1978—),男,博士,副教授,研究方向为电力系统运行分析与控制、输配电系统自动化等。

A Clustering Algorithm Based on CUDA Technology for Massive Electric Power Load Curves

WU Shuang[1], JI Cong[2], SUN Guoqiang[3]

(1. State Grid Jiangsu Electric Power Co., Ltd. Economic Research Institute, Nanjing 210008, China;
2. Jiangsu Frontier Electric Technology Co., Ltd., Nanjing 211102, China; 3. Research Center for Renewable Energy Generation Engineering, Ministry of Education (Hohai University), Nanjing 210098, China)

Abstract: With the explosive growth of user load data in power consumption information collection and load control systems, traditional computing frameworks and methods are faced with tremendous computational pressure when dealing with massive user load clustering and carrying out load characteristic analysis. In this paper, with a view to increasing accuracy and computational power of graphic process unit (GPU), the fast parallel *K*-means clustering algorithm for load curves is proposed based on Nvidia compute uniform device architecture (CUDA). This algorithm uses parallel acceleration strategies, such as parallelization of distance computing and curves counting, and rational allocation of thread blocks, which greatly improve the clustering speed of user load curves. A number of test examples show that the proposed clustering algorithm in this paper has a high acceleration ratio and strong adaptability, which is a good way to solve the problem of massive load curve clustering.

Key words: GPU; CUDA; parallel computing; mass data; *K*-means clustering; power load curve

(编辑 钱 悦)

(上接第 37 页)

Intermittent Load Selection and Recovery Strategy for Network Load Interactive User

QIU Chenguang[1], CHENG Jinmin[1], LI Xinjia[2], XIONG Zhen[2], LI Ping[2]

(1. State Grid Jiangsu Electric Power Co., Ltd., Nanjing 210024, China;
2. Jiangsu Frontier Electric Technology Co., Ltd., Nanjing 211102, China)

Abstract: It proposes the participating network load interaction users and their interruptible load selection, participation strategy and recovery principle in different running state and demand scene of power grid. So as to achieve fast load removal and recovery in power grid frequency emergency control, on the other hand, ensure that the important load in emergency control process is not affected, and ensure the safe operation of power grid under emergency condition.

Key words: network load interaction; interruptible load power; grid frequency

(编辑 钱 悦)