

Qi Liu · Xiaodong Liu · Lang Li ·
Huiyu Zhou · Hui-Huang Zhao *Editors*

Proceedings of the 9th International Conference on Computer Engineering and Networks

Advances in Intelligent Systems and Computing

Volume 1143

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing,
Universidad Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering,
University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,
Gyor, Hungary

Vladik Kreinovich, Department of Computer Science, University of Texas
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao
Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology,
University of Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute
of Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro,
Rio de Janeiro, Brazil

Ngoc Thanh Nguyen , Faculty of Computer Science and Management,
Wrocław University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at <http://www.springer.com/series/11156>

Qi Liu · Xiaodong Liu · Lang Li ·
Huiyu Zhou · Hui-Huang Zhao
Editors

Proceedings of the 9th International Conference on Computer Engineering and Networks

Editors

Qi Liu
School of Computer and Software
Nanjing University of Information
Science and Technology
Nanjing, Jiangsu, China

Xiaodong Liu
School of Computing
Edinburgh Napier University
Edinburgh, UK

Lang Li
College of Computer Science
and Technology
Hengyang Normal University
Hengyang, Hunan, China

Huiyu Zhou
Department of Informatics
University of Leicester
Leicester, UK

Hui-Huang Zhao
College of Computer Science
and Technology
Hengyang Normal University
Hengyang, Hunan, China

ISSN 2194-5357

ISSN 2194-5365 (electronic)

Advances in Intelligent Systems and Computing

ISBN 978-981-15-3752-3

ISBN 978-981-15-3753-0 (eBook)

<https://doi.org/10.1007/978-981-15-3753-0>

© Springer Nature Singapore Pte Ltd. 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

This conference proceedings is a collection of the papers accepted by the CENet2019—the 9th International Conference on Computer Engineering and Networks held on October 18–20, 2019, in Changsha, China.

This proceedings contains four parts: The first part focuses on System Detection and Application (25 papers); second part Machine Learning and Application (26 papers); third part Information Analysis and Application (27 papers); fourth part Communication Analysis and Application (39 papers).

Each part can be used as an excellent reference by industry practitioners, university faculties, research fellows, and undergraduates as well as graduate students who need to build a knowledge base of the most current advances and state-of-practice in the topics covered by this conference proceedings. This will enable them to produce, maintain, and manage systems with high levels of trust-worthiness and complexity.

Thanks to the authors for their hard work and dedication as well as the reviewers for ensuring the selection of only the highest-quality papers; their efforts made the proceedings possible.

Nanjing, China
Edinburgh, UK
Hengyang, China
Leicester, UK
Hengyang, China

Qi Liu
Xiaodong Liu
Lang Li
Huiyu Zhou
Hui-Huang Zhao

Contents

System Detection

Design of Collaboration Engine for Large-Scale Heterogeneous Clusters	3
Hui Zhao and Haifeng Wang	
Research on Multi-stage GPU Collaborative Model	13
Longxiang Zhang and Haifeng Wang	
HSO Algorithm for RRAP Problem in Three-State Device Network	23
Dongkui Li	
A Method of Service Function Chain Arrangement for Load Balancing	35
Zhan Shi, Zanhong Wu, and Ying Zeng	
The Optimal Implementation of Khudra Lightweight Block Cipher	43
Xiantong Huang, Lang Li, and Ying Guo	
Droop Control of Microgrid Based on Genetic Optimization Algorithm	55
Yongqi Tang, Xinxin Tang, Jiawei Li, and Mingqiang Wu	
Research and Implementation of a Remote Monitoring Platform for Antarctic Greenhouse	67
Kaiyan Lin, Chang Liu, Junhui Wu, Jie Chen, and Huiping Si	
Application Research in DC Charging Pile of Full-Bridge DC-DC Converter Based on Fuzzy Control	81
Binjun Cai, Tao Xiang, and Tanxin Li	

Performance Analysis of Multi-user Chaos Measurement and Control System	93
Lili Xiao, Guixin Xuan, and Yongbin Wu	
Effect of Magnetic Field on the Damping Capability of Ni_{52.5}Mn_{23.7}Ga_{23.8}/Polymer Composites	101
Xiaogang Sun, Xiaomin Peng, Lian Huang, Qian Tang, and Sheng Liu	
Knock Knock: A Binary Human–Machine Interactive Channel for Smartphone with Accelerometer	109
Haiyang Wang, Huixiang Zhang, Zhipin Gu, Wenteng Xu, and Chunlei Chen	
Remote Network Provisioning with Terminals Priority and Cooperative Relaying for Embedded SIM for NB-IoT Reliability	117
Yuxiang Lv, Yang Yang, Ping Ma, Yawen Dong, and Wei Shang	
Design of a Fully Programmable 3D Graphics Processor for Mobile Applications	127
Lingjuan Wu, Wenqian Zhao, and Dunshan Yu	
Outer Synchronization Between Complex Delayed Networks with Both Uncertain Parameters and Unknown Topological Structure	135
Zhong Chen, Xiaomei Tian, Tianqi Lei, and Junyao Chen	
Discussion on the Application of Digital Twins in the Wear of Parts	145
Fuchun Xie and Zhiyang Zhou	
Research on Key Algorithms of Segmented Spectrum Retrieval System.	155
Jianfeng Tang and Jie Huang	
Research on Fuzzy Control Strategy for Intelligent Anti-rear-end of Automobile.	165
Shihan Cai, Xianbo Sun, Yong Huang, and Hecui Liu	
Design and Optimization of Wind Power Electric Pitch Permanent Magnet Synchronous Servo Motor	183
Weicai Xie, Xiaofeng Li, Jie Li, Shihao Wang, Li He, and Lei Cao	
Design of Step Servo Slave System Based on EtherCAT	193
Liang Zheng, Zhangyu Lu, Zhihua Liu, and Chongzhuo Tan	
Investigation of Wind Turbine Blade Defect Classification Based on Deep Convolutional Neural Network	207
Ting Li, Yu Yang, Qin Wan, Di Wu, and Kailin Song	
Research on Weakening Cogging Torque of Permanent Magnet Synchronous Wind Generator	215
Quansuo Xiang, Qiuling Deng, Xia Long, Mengqing Ke, and Qun Zhang	

Research on Edge Network Resource Allocation Mechanism for Mobile Blockchain	225
Qiang Gao, Guoyi Zhang, Jinyu Zhou, Jia Chen, and Yuanyuan Qi	
MATSCO: A Novel Minimum Cost Offloading Algorithm for Task Execution in Multiuser Mobile Edge Computing Systems	233
Jin Tan, Wenjing Li, Huiyong Liu, and Lei Feng	
Medical Data Crawling Algorithm Based on PageRank	243
Maojie Hao, Peng Shu, Zhengan Zhai, Liang Zhu, Yang Yang, and Jianxin Wang	
The Alarm Feature Analysis Algorithm for Communication Network	255
Xilin Ji, Xiaodan Shi, Jinxi Han, Yonghua Huo, and Yang Yang	
Machine Learning	
Analysis of User Suspicious Behavior in Power Monitoring System Based on Text Vectorization and Unsupervised Machine Learning . . .	269
Jing Wang, Ye Liang, Lingyun Wu, Chengjiang Liu, and Peng Yang	
Prediction of Coal Mine Accidental Deaths for 5 Years Based on 14 Years Data Analysis	281
Yousuo Joseph Zou, Jun Steed Huang, Tong Xu, Anne Zou, Bruce He, Xinyi Tao, and Sam Zhang	
Patent Prediction Based on Long Short-Term Memory Recurrent Neural Network	291
Yao Zhang and Qun Wang	
Chinese Aspect-Level Sentiment Analysis CNN-LSTM Based on Mixed Vector	301
Kangxin Cheng, Zhili Wang, and Jiaqi Liu	
Prediction of RNA Structures with Pseudoknots Using Convolutional Neural Network	311
Sixin Tang, Shiting Li, and Jing Chen	
An Improved Attribute Value-Weighted Double-Layer Hidden Naive Bayes Classification Algorithm	321
Huanying Zhang, Yushui Geng, and Fei Wang	
An Improved Fuzzy C-Means Clustering Algorithm Based on Intuitionistic Fuzzy Sets	333
Fei Wang, Yushui Geng, and Huanying Zhang	
SFC Orchestration Method Based on Energy Saving and Time Delay Optimization	347
Zanhong Wu, Zhan Shi, and Ying Zeng	

Predict Oil Production with LSTM Neural Network	357
Chao Yan, Yishi Qiu, and Yongqiong Zhu	
Improved Fatigue Detection Using Eye State Recognition with HOG-LBP	365
Bin Huang, Renwen Chen, Wang Xu, Qinbang Zhou, and Xu Wang	
Portrait Style Transfer with Generative Adversarial Networks	375
Qingyun Liu, Feng Zhang, Mugang Lin, and Ying Wang	
Impossible Differential Analysis on 8-Round PRINCE	383
Yaoling Ding, Keting Jia, An Wang, and Ying Shi	
Multi-step Ahead Time Series Forecasting Based on the Improved Process Neural Networks	397
Haijian Shao, Chunlong Hu, Xing Deng, and Dengbiao Jiang	
An Improved Quantum Nearest-Neighbor Algorithm	405
Ying Zhang, Bao Feng, Wei Jia, and Cheng-Zhuo Xu	
A Trend Extraction Method Based on Improved Sliding Window	415
Ming Lu, Yongteng Sun, Hao Duan, and Zuguo Chen	
Three-Dimensional Dense Study Based on Kinect	423
Qin Wan, Yueping Xiao, Jine Yan, and Xiaolin Zhu	
A Cascaded Feature Fusion Residual Network for Image Super-resolution	431
Wang Xu, Renwen Chen, Bin Huang, Qinbang Zhou, and Yaoyu Wang	
Spp-U-net: Deep Neural Network for Road Region Segmentation	441
Yang Zhang, Dawei Luo, Dengfeng Yao, and Hongzhe Liu	
Research on the Computational Model of Problem-Based Tutoring	447
Yan Liu, Qiang Peng, and Lin Liu	
Research on Face Recognition Algorithms and Application Based on PCA Dimension Reduction and LBP	461
Kangman Li and Ruihua Nie	
Research on Algorithms for Setting up Advertising Platform Based on Minimum Weighted Vertex Covering	471
Ying Wang, Yaqi Sun, and Qinyun Liu	
Lane Detection Based on DeepLab	481
Mingzhe Li	
Hybrid Program Recommendation Algorithm Based on Spark MLlib in Big Data Environment	489
Aoxiang Peng and Huiyong Liu	

A Deep Learning Method for Intrusion Detection by Spatial and Temporal Feature Extraction	499
Haizhou Cao, Wenjie Chen, Shuai Ye, Ziao Jiao, and Yangjie Cao	
Network Traffic Prediction in Network Security Based on EMD and LSTM	509
Wei Zhao, Huifeng Yang, Jingquan Li, Li Shang, Lizhang Hu, and Qiang Fu	
An Introduction to Quantum Machine Learning Algorithms	519
Rongji Li, Juan Xu, Jiabin Yuan, and Dan Li	
Information Analysis	
Design of Seat Selecting System for Self-study Room in Library Based on Smart Campus	535
Jiujiu Yu, Jishan Zhang, Liqiong Pan, Cuiping Wang, and Gang Xiao	
An Optimized Scheme of Information Hiding-Based Visual Secret Sharing	545
Yi Zou, Lang Li, and Ge Jiao	
Constructions of Lightweight MDS Diffusion Layers from Hankel Matrices	555
Qiuping Li, Lang Li, Jian Zhang, Junxia Zhao, and Kangman Li	
Power Analysis Attack on a Lightweight Block Cipher GIFT	565
Jian Zhang, Lang Li, Qiuping Li, Junxia Zhao, and Xiaoman Liang	
Research and Development of Fine Management System for Underground Trackless Vehicles	575
Xu Liu, Guan Zhou Liu, Feng Jin, Xiao Lv, Yuan-Sheng Zhang, and Zhong-Ye Jiang	
A Certificate-Based Authentication for SIP in Embedded Devices	585
Jie Jiang and Lei Zhao	
An Anti-Power Attack Circuit Design for Block Cipher	591
Ying Jian Yan and Zhen Zheng	
A New Data Placement Approach for Heterogeneous Ceph Storage System	601
Fei Zheng, Jiping Wang, Xuekun Hao, and Hongbing Qiu	
Construction-Based Secret Image Sharing	611
Xuehu Yan, Guozheng Yang, Lanlan Qi, Jingju Liu, and Yuliang Lu	
A Novel AES Random Mask Scheme Against Correlation Power Analysis	625
Ge Jiao, Lang Li, and Yi Zou	

Deployment Algorithm of Service Function Chain with Packet Loss Rate Optimization	635
Yingjie Jiang, Xing Wang, Tao Zhao, Ying Wang, and Peng Yu	
Security Count Query and Integrity Verification Based on Encrypted Genomic Data	647
Jing Chen, Zhiping Chen, Linai Kuang, Xianyou Zhu, Sai Zou, Zhanwei Xuan, and Lei Wang	
Homological Fault Attack on AES Block Cipher and Its Countermeasures	655
Ning Shang, Jinpeng Zhang, Yaoling Ding, Caisen Chen, and An Wang	
Researching on AES Algorithm Based on Software Reverse Engineering	667
Qingjun Yuan, Siqi Lu, Zongbo Zhang, and Xi Chen	
MC-PKS: A Collaboration Public Key Services System for Mobile Applications	677
Tao Sun, Shule Chen, Jiajun Huang, Yamin Wen, Changshe Ma, and Zheng Gong	
Security Analysis and Improvement of User Authentication Scheme for Wireless Body Area Network	687
Songsong Zhang, Xiang Li, and Yong Xie	
Research and Implementation of Hybrid Encryption System Based on SM2 and SM4 Algorithm	695
Zhiqiang Wang, Hongyu Dong, Yaping Chi, Jianyi Zhang, Tao Yang, and Qixu Liu	
STL-FNN: An Intelligent Prediction Model of Daily Theft Level	703
Shaochong Shi, Peng Chen, Zhaolong Zeng, and Xiaocheng Hu	
WeChat Public Platform for Customers Reserving Bank Branches Based IoT	713
Jie Chen, Xiaoman Liang, and Jian Zhang	
Metadata Management Algorithm Based on Improved LSM Tree	725
Yonghua Huo, Ningling Ge, Jinxi Han, Kun Wang, and Yang Yang	
A Text Information Hiding Method Based on Sentiment Word Substitution	735
Fufang Li, Han Tang, Liangchen Liu, Binbin Li, Yuanyong Feng, Wenbin Chen, and Ying Gao	
Research on Information Hiding Based on Intelligent Creation of Tang Poem	745
Fufang Li, Binbin Li, Yongfeng Huang, WenBin Chen, Lingxi Peng, and Yuanyong Feng	

Design and Evaluation of Traffic Congestion Index for Yancheng City 757
Fei Ding, Yao Wang, Guoxiang Cai, Dengyin Zhang, and Hongbo Zhu

A Survey on Anomaly Detection Techniques in Large-Scale KPI Data 767
Ji Qian, Guangfu Zeng, Zhiping Cai, Shuhui Chen, Ningzheng Luo, and Haibing Liu

A Counterfactual Quantum Key Distribution Protocol Based on the Idea of Wheeler’s Delayed-Choice Experiment 777
Nan Xiang

A Non-intrusive Appliances Load Monitoring Method Based on Hourly Smart Meter Data 785
Chunhe Song, Zhongfeng Wang, Shuji Liu, Libo Xu, Dapeng Zhou, and Peng Zeng

Design and Implementation of VxWorks System Vulnerability Mining Framework Based on Dynamic Symbol Execution 801
Wei Zheng, Yu Zhou, and Boheng Wang

Communication Analysis

Deep Web Selection Based on Entity Association 815
Song Deng, Wen Luo, and Xueke Xu

Bayesian-Based Efficient Fault Location Algorithm for Power Bottom-Guaranteed Communication Networks 827
Xinzhan Liu, Weijian Li, Peng Liu, and Huiqing Li

Fault Diagnosis Algorithm for Power Bottom-Guaranteed Communication Network Based on Random Forest 837
Zhan Shi, Ying Zeng, Zhengfeng Zhang, and Tingbiao Hu

Design of a Real-Time Transmission System for TV- and Radar-Integrated Videos 847
Yingdong Chu

Research on Automated Penetration Testing Framework for Power Web System Integrating Property Information and Expert Experience 855
Zesheng Xi, Jie Cui, and Bo Zhang

VNF Deployment Method Based on Multipath Transmission 867
Zhan Shi, Ying Zeng, and Zanhong Wu

Service Chain Orchestration Based on Deep Reinforcement Learning in Intent-Based IoT 875
Zhan Shi, Ying Zeng, and Zanhong Wu

Design and Implementation of Wi-Fi-Trusted Authentication System Based on Blockchain	883
Qiang Gao, Zhifeng Tian, and Yao Dai	
Efficient Fault Location Algorithm Based on D-Segmentation for Data Network Supporting Quantum Communication	893
Kejun Xie, Ziyang Zhao, Dequan Gao, Bozhong Li, and Hao Chen	
Equipment Fault Prediction Method in Power Communication Network Based on Equipment Frequency Domain Characteristics. . . .	901
Ruide Li, Zhirong Peng, Xi Yang, Tianyi Zhang, and Cheng Pan	
VNF Placement and Routing Algorithm for Energy Saving and QoS Guarantee	911
Ying Zeng, Zhan Shi, and Zanhong Wu	
Research on IoT Architecture and Application Scheme for Smart Grid	921
Dedong Sun, Wenjing Li, Xianjiong Yao, Hui Liu, Jinlong Chai, Kunyi Xie, Liang Zhu, and Lei Feng	
A Method of Dynamic Resource Adjustment for 5G Network Slice . . .	929
Qinghai Ou, Jigao Song, Yanru Wang, Zhiqiang Wang, Yang Yang, Diya Ran, and Lei Feng	
Network Slice Access Selection Scheme for 5G Network Power Terminal Based on Grey Analytic Hierarchy Process	937
Yake Zhang, Xiaobo Jiao, Xin Yang, Erpeng Yang, Jianpo Du, Yueqi Zi, Yang Yang, and Lei Feng	
Resource Allocation Mechanism in Electric Vehicle Charging Scenario for Ubiquitous Power-IoT Coverage Enhancement	945
Yao Wang, Yun Liang, Wenfeng Tian, Xiaoyan Sun, Xiyang Yin, Liang Zhu, and Diya Ran	
Computation Resource Allocation Based on Particle Swarm Optimization for LEO Satellite Networks.	955
Shan Lu, Fei Zheng, Wei Si, and Mengqi Liu	
Security Analysis and Protection for Charging Protocol of Smart Charging Pile	963
Jiangpei Xu, Xiao Yu, Li Tian, Jie Wang, and Xiaojuan Liu	
The Power Distribution Control Strategy of Fully Active Hybrid Energy Storage System Based on Sliding Mode Control	971
Zhangyu Lu, Chongzhuo Tan, and Liang Zheng	
Secure Communication with Two-Stage Beamforming for Wireless Energy and Jamming Signal Based on Power Beacon	979
Dandan Guo, Jigao Song, Xuanzhong Wang, Xin Wang, and Yanru Wang	

A Research of Short-Term Wind Power Prediction Based on Support Vector Regression 991
Shixiong Bai and Feng Huang

Droop Control Strategy of Microgrid Parallel Inverter Under Island Operation 997
Xia Long, Qiuling Deng, Quansuo Xiang, Mengqing Ke, and Qun Zhang

ZigBee-Based Architecture Design of Imperceptible Smart Home System 1003
Juanli Kuang and Lang Li

Data Collection of Power Internet of Things Sensing Layer Based on Path Optimization Strategy 1013
Xianjiong Yao, Dedong Sun, Qinghai Ou, Yilong Chen, Liang Zhu, Diya Ran, and Yueqi Zi

STFRC: A Multimedia Stream Congestion Control Algorithm Based on TFRC 1021
Fuzhe Zhao and Yuqing Huang

Multi-controller Cooperation High-Efficiency Device Fault Diagnosis Algorithm for Power Communication Network in SDN Environment 1031
Ruide Li, Zhenchao Liao, Minghua Tang, Jiajun Chen, and Weixiong Li

Service Recovery Algorithm for Power Communication Network Based on SDN Multi-mode Channel 1041
Deru Guo, Xutian He, Guanqiang Lin, Cizhao Luo, and Baiwei Zhong

A Multi-channel Aggregation Selection Mechanism Based on TOPSIS in Electric Power Communication Network 1051
Deru Guo, Peifang Song, Cizhao Luo, Shuqing Li, and Feida Jiang

Fault Diagnosis and CAN Bus/Ethernet Redundancy Design of a Monitoring and Control System 1063
Weizhi Geng, Daojun Fu, and Mengxin Wu

Terminal Communication Network Fault Diagnosis Algorithm Based on TOPSIS Algorithm 1071
Ran Li, Haiyang Cong, Fanbo Meng, Diying Wu, Yi Lu, and Taiyi Fu

Real-Time Pricing Method Based on Lyapunov for Stochastic and Electric Power Demand in Smart Grid 1081
Yake Zhang, Yucong Li, and Diya Ran

**Resource Allocation Algorithm for Power Bottom-Guaranteed
Communication Network Based on Network Characteristics
and Historical Data 1089**
Xinzhan Liu, Zhongmiao Kang, Yingze Qiu, and Wankai Liu

**An Outlier Detection Algorithm for Electric Power Data
Based on DBSCAN and LOF 1097**
Hongyan Zhang, Bo Liu, Peng Cui, You Sun, Yang Yang,
and Shaoyong Guo

**Power Anomaly Data Detection Algorithm Based on User Tag
and Random Forest 1107**
JianXun Guo, Bo Liu, Hongyan Zhang, Qiang Li, Shaoyong Guo,
and Yang Yang

**Optimal Distribution in Wireless Mesh Network with Enhanced
Connectivity and Coverage. 1117**
Hao Zhang, Siyuan Wu, Changjiang Zhang, and Sujatha Krishnamoorthy

**A Traffic Anomaly Detection Method Based on Gravity Theory
and LOF 1129**
Xiaoxiao Zeng, Yonghua Huo, Yang Yang, Liandong Chen, and Xilin Ji

**Construction of Management and Control Platform for Bus Parking
and Maintenance Field Under Hybrid Cloud Computing Mode. 1139**
Ying-long Ge

**A Data Clustering Method for Communication Network
Based on Affinity Propagation 1151**
Junli Mao, Lishui Chen, Xiaodan Shi, Chao Fang, Yang Yang,
and Peng Yu

**Priority-Based Optimal Resource Reservation Mechanism
in Wireless Sensor Networks for Smart Grid 1161**
Hongfa Li, Jianfa Pu, Duanyun Chen, Yongtian Xie, Wenming Fang,
and Shimulin Xie

**Energy-Efficient Clustering and Multi-hop Routing Algorithm
Based on GMM for Power IoT Network 1169**
Yuanjiu Li, Junrong Zheng, Hongpo Zhang, Xinsheng Ye, Zufeng Liu,
and Jincheng Li

System Detection

Design of Collaboration Engine for Large-Scale Heterogeneous Clusters



Hui Zhao and Haifeng Wang

Abstract Aiming at the low utilization rate of intensive computing cores in large heterogeneous clusters and the high complexity of collaborative computing between GPU and multi-core CPUs, this paper aims to improve resource utilization and reduce programming complexity in heterogeneous clusters. A new heterogeneous cluster cooperative computing model and engine design scheme are proposed. The complexity of communication between nodes and cooperative mechanism within nodes is analyzed. Coarse-grained cooperative execution plan is represented by template technology, and fine-grained cooperative computing plan is created by finite automata. The experimental results validate the effectiveness of the collaborative engine. Comparing with the manual programming scheme, it is found that the computational performance loss is less than 4.2%. The collaborative computing engine can effectively improve the resource utilization of large-scale heterogeneous clusters and the programming efficiency of ordinary developers.

Keywords Collaborative computing model · Finite automata · Computing engine · Template technology

1 Introduction

With the development of large data applications, data-intensive computing has become an important area of high-performance computing. GPU general computing is very suitable for data-intensive tasks. Heterogeneous cluster built by GPU and multi-core CPU has become a major data processing solution [1]. It can effectively improve the computing performance and efficiency and has a wide application

H. Zhao · H. Wang

College of Information Science and Engineering, Linyi University, Shuangling Road, Linyi 276002, China

H. Wang (✉)

Shandong Key Laboratory of Network, Research Institute of Linyi University, Shuangling Road, Linyi 276002, China
e-mail: gadfly7@126.com

© Springer Nature Singapore Pte Ltd. 2021

Q. Liu et al. (eds.), *Proceedings of the 9th International Conference on Computer Engineering and Networks*, Advances in Intelligent Systems and Computing 1143, https://doi.org/10.1007/978-981-15-3753-0_1

prospect [2, 3]. In 2010, “Tianhe 1” supercomputer took the lead in adopting the heterogeneous system of GPU + CPU, using 7168 NVIDIA GPUs. And in the top ten supercomputers in the world, two systems adopt the structure of GPU + CPU. Industry has also begun to use GPU heterogeneous clusters to process large data. Cloud computing service Amazon provides users with GPU computing examples in EC2 cluster to accelerate the performance of oil exploration, meteorological analysis, image processing and other applications. In heterogeneous cluster of GPU, the cooperative computing behavior of GPU and multi-core CPU not only increases the density of computing resources within a single node, but also improves the utilization and efficiency of computing resources.

At present, in the research of GPU and CPU collaborative computing, a large number of data-intensive algorithms have been transplanted to GPU to improve the performance. For example, to solve the problem of subset summation, GPU and CPU multi-core cooperative algorithm is proposed to solve the load balancing between two types of processors and the optimization of communication between nodes and nodes [4]; to query protein sequence data, GPU and multi-core CPU are used to reduce the operation time, mainly to implement two widely used sequence alignment algorithms, SW and NW, and to study the load balancing strategy between cluster nodes. Fine-grained collaborative operation process in nodes [5]; 3-D terrain reconstruction task in UAV perception system belongs to data-intensive real-time computing. Song et al. explored data access operation in GPU-CPU collaborative computation of 3-D reconstruction algorithm and achieved good acceleration effect [6]; proposed CPU/GPU collaborative algorithm for parallel difference algorithm and applied it to mass LIDAR point cloud generation digital. In the elevation model, researchers designed the first-in-first-out queue with concurrent access shared by CPU and GPU to manage tasks and proposed a collaborative framework of parallel collaborative difference [7]; proposed a collaborative parallel computing algorithm between CPU and GPU for local search in optimization algorithm; designed a HySyS prototype system to solve a series of combinatorial optimization problems [8]; Moim implements a collaborative computing system based on MapReduce mode. To decompose the three computing stages of MapReduce, it realizes task allocation and load balancing between GPU and CPU multi-core.

However, there is a lack of research on the commonality of collaborative computing. In collaborative computing programming of GPU and multi-core CPU for different domains, users with different application backgrounds need to write GPU and CPU collaborative computing programs according to business logic. They should not only pay attention to the algorithm logic in their respective domains, but also to the collaborative computing logic. Therefore, the development of GPU collaborative computing program is very difficult, which requires users to have a thorough understanding of thread synchronization, data transmission between GPU and CPU and communication between network nodes. In order to reduce the programming difficulty of GPU collaborative computing, a complete design scheme of collaborative computing engine is proposed in this paper. Firstly, the common problems of collaborative processes in different computing jobs are extracted, and then, the

collaborative logic processes in computing jobs are extracted to reduce the programming complexity of users by reducing repetitive programming. The organizational structure of this paper is as follows: Sect. 2 introduces the main problems to be solved by the collaborative computing engine; Sect. 3 provides the specific scheme for the design of the collaborative computing engine; Sect. 4 summarizes and looks forward to future work.

2 Problem Description

Firstly, the main problems of collaborative computing between GPU and multi-core CPU are analyzed from the perspective of computing system. When GPU heterogeneous cluster processes large data jobs in a distributed way, the input data is divided into several data sets according to load balancing scheduling strategy. Each computing node is responsible for processing the corresponding data, and finally, the final results are summarized. In a single computing node, the data is divided into two parts, which are accomplished by GPU and multi-core CPU, respectively. Therefore, collaborative computing is divided into two parts: intra-node collaboration and inter-node collaboration. The input data of large data processing jobs is loaded into CPU memory by network or external memory. There are three areas in CPU memory: CPU memory area, buffer area and memory area exchanged with GPU. Among them, CPU memory area provides data to CPU computing core; memory area exchanged with GPU is the area where CPU control core transmits data to GPU; computing data is transferred to GPU display by exchange memory, and then computed by GPU and returned to CPU switching memory area; the buffer is paging memory, which is used to aggregate GPU and CPU calculation results in nodes [3]. It can be seen that the cooperative computing in a single node includes inter-thread synchronization, lock operation and complex data communication operation. On the other hand, the process of collaborative computing among nodes is as follows: each node process is responsible for the calculation and collaborative control of the node, and then keeps synchronization and data communication with other nodes in the network. Finally, the global collaborative computing results are summarized. Collaboration among nodes includes process synchronization and complex network communication operations.

Secondly, the problem of collaborative computing is analyzed from the perspective of computing operations. User jobs are logically divided into two parts: domain algorithm logic and collaborative computing logic. The concrete realization forms of domain algorithm logic are GPU kernel function, CPU thread function and CPU data merging process. The design goal of the collaborative computing engine is to extract the collaborative computing logic, so that ordinary users only pay attention to the design of CUDA kernel function, thread function and merging process. After judging the characteristics of large data computing jobs, the cooperative execution plan between CPU computing core and GPU is generated and distributed to heterogeneous cluster nodes for execution. Collaborative execution planning is a coarse-grained activity sequence, which guides the calculation of GPU and multi-core CPU.

3 Cooperative Computing Engine

This section introduces the specific design scheme of the collaborative computing engine and the formal representation of the performance of collaborative computing.

3.1 Design of Cooperative Computing Engine

The architecture of the collaborative computing engine is shown in Fig. 1. The collaborative computing engine is divided into two levels. Coarse-grained collaborative planning is used between computing nodes, while fine-grained collaborative planning is used within computing nodes. Jobs are extracted from job queues, and their types are judged. Cooperative planning templates are imported into template libraries according to job types to achieve basic task allocation and computing data deployment in a distributed network environment, and synchronization and concurrent operations between nodes are specified. On the other hand, fine-grained finite state machine (FSM) is used to generate cooperative execution plan. When a specific event occurs, the state changes are triggered and the corresponding actions or operations are executed to achieve more detailed cooperative computing function within the node.

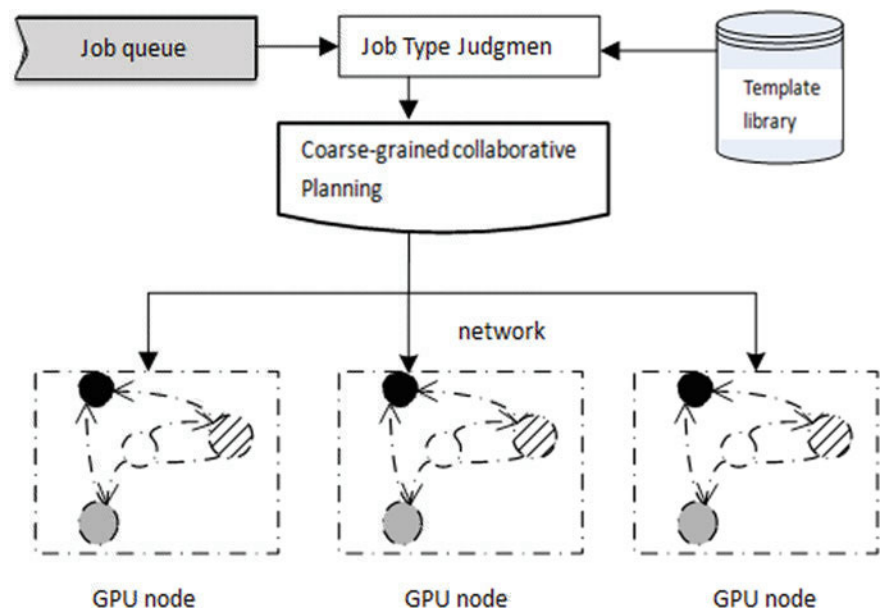


Fig. 1 GPU structure diagram of cooperative computing engine

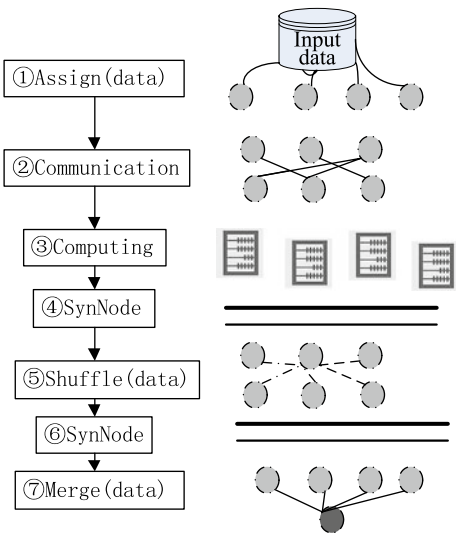
The design of coarse-grained and fine-grained mixing has the following advantages: The template technology is used to realize the generation function of coarse-grained collaborative planning. Firstly, job types are classified simply, and corresponding collaborative computing models are provided according to different job types. This technology reduces the complexity of collaborative engine and improves the feasibility of implementation.

3.2 Template Cooperative Execution Plan

Template technology is a knowledge expression method based on reuse technology and similarity principle of things. The basic idea is to abstract a framework template from a kind of similar things. Collaborative computing engine extracts common processes from collaborative processes among computing nodes and defines common processes as templates for reuse. Job collaborative computing flow for large data processing is shown in Fig. 2.

Figure 2 shows the cooperative mode of MapReduce batch processing mode. Firstly, the reasonable deployment of input data is realized according to the load balancing strategy, and the data needed for calculation is obtained through the network transmission of each computing node [9]. Then, each computing node asynchronously completes its own computing tasks. The computing process of each node is synchronized [10]. The intermediate results are mixed and synchronized, and finally merged computing is realized. In this collaborative process framework, the differences among different computing jobs are data allocation strategy, data shuffling strategy and computing load balancing.

Fig. 2 Inter-node collaborative process framework



3.3 State Machine Execution Plan

Finite state machine is used to generate fine-grained collaboration processes in real time. Finite state machine (FSM) is a mathematical model that represents finite states and the migration and action between states [11]. The CPU, GPU, memory, buffer and GPU display are regarded as different data storage and computing objects. The finite state machine (FSM) models the behavior of each object, describes the state sequence of the object in each computing stage, and how to respond to various external events.

FSM can be expressed as $M = (Q, q_0, \Sigma, \delta, F)$, where Q is a non-empty finite set of states, $q \in Q$, A state in which q is M ; q_0 is the start or start state of M ; Σ represents the input set of the system; δ is a state transition function; F is the set of termination states of M , if $q \in F$, q is the termination state of M . The cooperative operation process in GPU computing nodes is represented by FSM; Table 1 provides the basic state set and system input set.

After modeling the collaborative computing process in nodes with finite state machine, the mapping between collaborative process and finite state machine is needed. Collaborative computing process is composed of several specific operation steps, which have some logical steps, including shunting, aggregation, parallel, jump. In finite state machines, all data-centric states, such as full memory area and empty GPU display memory area, are the relationship between producers and consumers. Therefore, the mapping idea is that every step of collaborative computing is regarded as a state, when one state migrates to another state under certain conditions, it is also considered as a step; when a state has multiple inputs, there must be aggregation in the process of collaboration, from different states to the same state. For example,

Table 1 Symbol table of finite state machine system in node

System state		System input	
Symbol	Meaning	Symbol	Meaning
S_0	Ready state	x_1	Start threads or receive synchronization signals
S_1	GPU computing end	x_2	Terminate thread or task queue empty
S_2	GPU computing end	x_3	Terminate thread or task queue empty
S_3	GPU full display area	x_4	Transfer data to equipment
S_4	CPU full memory area	x_5	Loading data from external memory
S_5	GPU display memory space	x_6	Start GPU computing
S_6	Cache full	x_7	Memcpy data replication operation
S_7	Cache space	x_8	Memcpy data replication operation
S_8	Calculation failure	x_9	Abnormal, fault events
S_9	Calculation successful	x_{10}	Calculate startup events
S_{10}	Mission queue full	x_{11}	Task entry operation
S_{11}	Mission queue empty	x_{12}	Merge computing end

when a thread receives synchronization signals and the thread computation ends, the thread enters the ready state; when a state has multiple outputs, there must be shunting and concurrency in the collaborative process, which can be migrated from the current state to different states. From a technical point of view, a class is designed according to the finite state machine to generate fine-grained cooperative execution plan in real time, but the judgment of state migration will cause some performance loss.

4 Experiments and Analysis

In order to verify the effectiveness of the function of the collaborative computing engine and the performance loss of the algorithm for generating the collaborative execution plan, a typical large data computing job was selected to carry out the experiment. GPU heterogeneous cluster with four computing nodes in the Laboratory Bed, Network is Gigabit Ethernet, GPU Computing Node is configured as GPU Tesla K40 with global memory 12 GB. CPU is Intel Xeon E5-2660 8 core, 2.2 GHz, 32 GB Memory and Hard disk 2T. The operating system is Ubuntu 14.04, CUDA 7.5. Selecting large-scale microblog analysis jobs and Sort jobs in Hibench [12] as collaborative computing jobs in heterogeneous clusters of GPU, microblog data analysis operation can flexibly control data size and belongs to typical data-intensive operation, which is very suitable for GPU computing; Sort job is an I/O-intensive large data processing job, whose input data, intermediate data and output data are the same size. In the experiment, the judgment part of computing job type is simplified, and the corresponding template is used directly for two kinds of jobs, while collaborative computing within nodes is the execution plan generated by engine. (Collaborative engine abbreviated as CE). The benchmark procedure for experimental comparison is written by more skilled personnel (short for MC).

The actual computing time of microblog data analysis is relatively short. Table 2 shows that with the increase of microblog data size, the performance loss caused by the computational complexity of finite state machines in nodes decreases gradually. For such flow computing jobs with large input data and small output data, the performance loss can be controlled within 3%. The data scale of Sort job is large and the actual calculation time is relatively long. There are many types of cooperative state switching within nodes, so the performance loss of cooperative computing engine is between 3.1 and 4.2%. As can be seen from Table 3, the performance loss

Table 2 Performance comparison table of microblog data analysis

Data size (GB)	4	6	12	20
MC (s)	1.288	2.096	3.040	3.984
CE (s)	1.325	2.148	3.109	4.067
Performance loss (%)	2.9	2.5	2.3	2.1

Table 3 SortPerformance comparison table

Data size (GB)	40	63	85	105	126	150
MC (s)	41.21	55.22	102.36	126.52	156.31	190.28
CE (s)	42.47	57.15	106.04	131.70	162.87	198.27
Performance loss (%)	3.1	3.5	3.6	4.1	4.2	4.2

increases with the increase of data size in Port. Through performance tracking, it is found that because Sort belongs to I/O-intensive large data computing operations [13], the memory data area, GPU display data area and memory buffer area in the node will appear the phenomenon of local data waiting for synchronization lock, so the performance loss is larger than that of microblog data analysis.

Typical large data computing jobs verify that the collaborative computing engine can effectively generate the collaborative execution plan. The results of microblog data analysis and Sort are consistent with the result of experts. The performance loss caused by co-generation logic can be controlled to the acceptable range of ordinary users.

5 Summary and Prospect

In heterogeneous GPU clusters, multi-core CPU is a computational resource that is easily overlooked. CPU/GPU cooperative computing mode increases the computing resource density within a single node and can effectively improve the performance of parallel computing. Cooperative computing engine generates corresponding collaborative execution process for user jobs, which reduces the difficulty of user writing collaborative computing logic. The function of collaborative computing engine is validated by data-intensive flow computing tasks and I/O-intensive Sort tasks. The performance loss of collaborative computing is also controlled within acceptable range. However, when a finite state machine generates fine-grained cooperative operations within a node, the problem of multiple threads competing for lock resources is obvious in I/O-intensive tasks, resulting in greater performance loss than common tasks. In order to solve this problem, probabilistic finite state machine (PFSM) is introduced into the design of cooperative generation within nodes. Empirical probability is used to solve the problem of multi-state transition conflict.

Acknowledgements This work was supported in part by the Shandong Province Key Research and Development Program of China (No. 2018GGX101005), the Shandong Province Natural Science Foundation, China (No. ZR2017MF050).

References

1. Zhang, F., Zhai, J.D., He, B.S., et al.: Understanding co-running behavior on integrated CPU/GPU architectures. *IEEE Trans. Parallel Distrib. Syst.* **28**(3), 905–918 (2017)
2. Zhang, H., Zhang, L.B., Wu, Y.J.: Large-scale graph data processing based on multi-GPU platform. *J. Comput. Res. Dev.* **55**(2), 273–288 (2018)
3. Li, T., Dong, Q.K., et al.: Research on parallel computing mode of GPU task based on thread pool. *Chin. J. Comput.* **41**(10), 2175–2192 (2018)
4. Wan, L.J., Li, K.L., Li, K.Q.: A novel cooperative accelerated parallel two-list algorithm for solving the subset-sum problem on a hybrid CPU-GPU cluster. *J. Parallel Distrib. Comput.* **97**, 112–123 (2016)
5. Zhou, W., Cai, Z.X., et al.: A multi-GPU protein database search model with hybrid alignment manner on distributed GPU clusters. *Concurr. Comput.* **30**(8), 1–13 (2018)
6. Song, W., Zou, S.H., et al.: A CPU-GPU hybrid system of environment perception and 3D terrain reconstruction for unmanned ground vehicle. *J. Inf. Process. Syst.* **14**(6), 1445–1456 (2018)
7. Wang, H.Y., Guan, X.F., Wu, H.Y.: A cooperative parallel spatial interpolation algorithm for CPU/GPU heterogeneous environment. *Geomat. Inf. Sci. Wuhan Univ.* **42**(12), 1688–1695 (2017)
8. Vidal, P., Alba, E., Luna, F.: Solving optimization problems using a hybrid systolic search on GPU plus CPU. *Soft. Comput.* **21**, 3227–3245 (2017)
9. Mengjun, X., Kyoung-Don, K., Can, B.: Moim: a multi-GPU mapreduce framework. In: 16th International Conference on CSE, 1279–1286 (2013)
10. Guo, M.S., Zhang, Y., Liu, T.: Research advances and prospect of recognizing textual entailment and knowledge acquisition. *Chin. J. Comput.* **40**(4), 889–909 (2017)
11. Shan, J.H., Zhang, L., et al.: Extending timed abstract state machines for real-time embedded software. *Acta Sci. Nat. Univ. Pekin.* (2019). <https://doi.org/10.13209/j.0479-8023.2019.005>
12. Huang, S., Huang, J., et al.: The Hiben benchmark suite: characterization of the mapreduce-based data analysis. In: *IEEE International Conference on Data Engineering Workshops* vol. 74, pp. 41–51 (2010)
13. Osama, A.A., Muhammad, J.I., et al.: Analyzing power and energy efficiency of bitonic mergesort based on performance evaluation. *IEEE Access* **6**, 42757–42774 (2018)

Research on Multi-stage GPU Collaborative Model



Longxiang Zhang and Haifeng Wang

Abstract GPU heterogeneous cluster is extensively utilized in the field of data analysis and processing. Nevertheless, research and studies on collaborative activity model in computing elements of GPU heterogeneous clusters are still inadequate. To conduct research on GPU and multi-core CPU cooperative computing from a theoretical perspective, a multi-stage cooperative computing model (p-DCOT) is established. Bulk synchronous parallel (BSP) model is the core of p-DCOT. Cooperative computing is divided into three layers: data layer, computing layer and communication layer. Computing and communication are described and formalized by matrix. Lastly, representative computing examines the effectiveness of model and parameter analysis. The collaborative computing model finds out the collaborative computing in big data analysis and processing.

Keywords Collaborative model · Load balancing · Big data computing · GPU cluster

1 Introduction

Large-scale data analysis and processing changes the development of high-performance computing. The emergence of high-performance computing platforms has changed from computing platforms to professional computing systems, which processes the big data in different industries [1]. The thread mechanism of GPU is capable of processing data-intensive calculation. Heterogeneous cluster of GPU and multi-core CPU create new opportunities for big data analysis and processing, which enhance calculation performance and improve efficiency [2]. An increasing number of research and studies on the role of GPU heterogeneous cluster in big data analysis and processing are carried out. Due to dynamic data input and random variation of calculation loading, calculation pattern, optimal operation, load balancing, reliability

L. Zhang · H. Wang (✉)

School of Information Science& Engineering, Linyi Univesity, Shuangling Road, Linyi Shandong 276002, China

e-mail: gadfly7@126.com

© Springer Nature Singapore Pte Ltd. 2021

Q. Liu et al. (eds.), *Proceedings of the 9th International Conference on Computer Engineering and Networks*, Advances in Intelligent Systems and Computing 1143, https://doi.org/10.1007/978-981-15-3753-0_2

and expansibility of GPU have been negatively affected[3, 4]. For this reason, the systems encounter various limitations. Related theories on cooperative computing of big data analysis and processing are inadequate [5–7]. In this paper, theoretical models analyze the general calculation rules and extensive forms of MapReduce, thereby conducting optimization studies. The studies focus on cooperative computation models of different computation patterns and establish a new theoretical model at multiple stages and levels. Moreover, as the theoretical tool of discussing GPU and multi-core cooperative computing, primary model parameters are analyzed and validity of models is verified.

2 Problem Description

Assume GPU heterogeneous clusters include n nodes, and every node also involves CPU and GPU. If multi-core CPU has m cores, cybernetics cores are in interactive operation with GPU [8]. The remaining $m - 1$ are computing cores. Computing cores are divided into cooperative computing core (CCC) and standalone computing core (SCC). When it comes to big data processing and computing, cooperative computing cores and GPU divide data set, and complete cooperative and concurrent computing. Standalone computing cores are in charge of reduce or data aggregation within nodes. From this perspective, in GPU heterogeneous computing system, cooperative computing occurs among and within nodes. Therefore, computing and communicating process become more complicated.

3 Cooperative Computing Model

3.1 Overview of Cooperative Computing Model

As is shown in Fig. 1, a big data computing task is completed in p stages. Each stage is divided into data layer, computing layer and communication layer. On data layer, cybernetics cores of CPU complete distributed mapping. On computer layer, cybernetics cores of CPU, computing cores and GPU conduct concurrent operation. On communication layer, data aggregation occurs among distributed nodes. Intermediate results of computing layer are transmitted and aggregated, in order to prepare data for the next stage of computing.

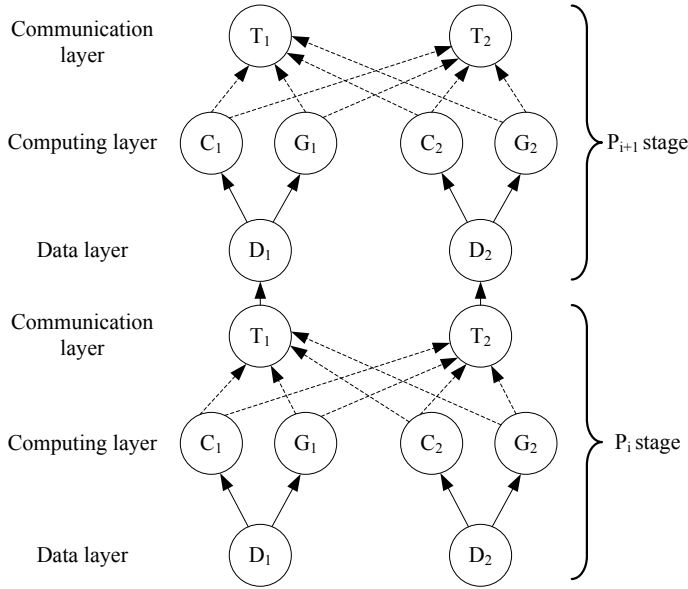


Fig. 1 Cooperative computing process of heterogeneous system

3.2 Formalized Description

Cooperative calculation model is described by matrix. Formalized expression of p-DCOT model is shown as follows:

$$p\text{-DCOT} = D((C + O)T)^p = D(C_1 + O_1)T_1 \dots D(C_p + O_p)T_p \quad (1)$$

Equation (1) shows the p phase of iteration. The expansion equation of each computing phase is shown as follows. Formalization of the i stage:

$$\begin{aligned}
 D(C_i + O_i)T_i &= \beta [D_1 \ D_2 \ \dots \ D_n] \\
 &\times \left(\delta_i W_g \begin{vmatrix} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & c_{ni} \end{vmatrix} + (1 - \delta_i) W_c \begin{vmatrix} o_1 & 0 & 0 & 0 \\ 0 & o_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & o_{ni} \end{vmatrix} \right) \\
 &\times \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,ni+1} \\ t_{2,1} & t_{2,2} & \dots & t_{2,ni+1} \\ \dots & \dots & \dots & \dots \\ t_{ni,1} & t_{ni,2} & \dots & t_{ni,ni+1} \end{bmatrix} \quad (2)
 \end{aligned}$$

Equation (2) demonstrates: cooperation of n_i computing nodes in the i phase of calculation. The diagonal elements of diagonal matrix C_1 to C_{ni} indicate the concurrence computing of GPU nodes. The diagonal elements of diagonal matrix O_1 to O_{ni} represent the concurrence computing of GPU cooperative computing cores. Matrix addition shows the convergence of GPU computing and CPU multi-core computing. Parameters δ_i represent loading and allocation factors of GPU and CPU in the nodes. This parameter determines the data size for GPU and CPU to process in each node. Parameter β_i is the loading and allocation factor that maps computing tasks to different machines. Weight matrix W_g and W_c represent the parameter matrix of GPU and CPU performance differences. Average matrix T represents the point-to-point communication status among the nodes. Element t_{ij} represents the telecommunication relationship between node n_i of the i phase and node n_j of $i + 1$.

3.3 Parameter Analysis

1. GPU and CPU loading ratio δ in single computing node

Assume that L is the data loading for single node. In single nodes, there are R cooperative computing cores and GPU. Peak performance of each cooperative computing core is P_{cpu} (assume they are isomorphic computing models. Cooperative computing cores have the same peak performances). Peak performance of GPU is P_{GPU} . Obtain the loading ratio of GPU and CPU δ in the same locking range. Equation (3) shows the completion time of whole computational nodes:

$$\begin{cases} T_{\text{total}} = \max(T_{\text{gpu}}, T_{\text{cpu}}) \\ T_{\text{cpu}} = \frac{L \times \delta}{R \times P_{\text{cpu}}} \\ T_{\text{gpu}} = \frac{L(1-\delta)}{P_{\text{gpu}}} \end{cases} \quad (3)$$

In order to achieve the minimum completion time T_{total} for the whole computing nodes, cooperative computing cores and GPU computing time should be equal to each other, which is shown in Eq. (4):

$$\frac{L \times \delta}{R \times P_{\text{cpu}}} = \frac{L(1-\delta)}{P_{\text{gpu}}} \quad (4)$$

Equation (4) determines the loading ratio δ of GPU and CPU

$$\delta = \frac{R}{P_{\text{gpu}}/P_{\text{cpu}} + R} \quad (5)$$

In practical application, accelerate cooperative computing cores of GPU and CPU to replace peak performance ratio, i.e., $s = P_{\text{gpu}}/P_{\text{cpu}}$. Next, Eq. (5) is derived from approximation of Eq. (6).