Proceedings

26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

PDP 2018

Cambridge, United Kingdom
21-23 March 2018

Edited by
Ivan Merelli, Pietro Liò and Igor Kotenko

CPS
Conference Publishing Services

26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

PDP 2018

Table of Contents

Message from General Chairs ..... **xix**
Message from Organizing Chairs ..... **xxi**
Conference Organization ..... **xxii**
Program Committee ..... **xxiii**
Reviewers ..... **xxviii**

Main Track

A Generic Learning Multi-agent-System Approach for Spatio-Temporal-, Thermal- and Energy-Aware Scheduling 1
Christina Herzog (University of Pau and Adour Country) and Jean-Marc Pierson (University of Toulouse)

A Parallel Implementation of WAND on GPUs 10
Roussian Gaioso (Federal University of Sao Carlos), Veronica Gil-Costa (CONICET-UNSL), Helio Guardia (Federal University of Sao Carlos), and Hermes Senger (Federal University of Sao Carlos)

A Portable Multidimensional Coarray for C++ 18
Felix Mößbauer (Ludwig-Maximilians-Universität München; Munich), Roger Kowalewski (Ludwig-Maximilians-Universität München; Munich), Tobias Fuchs (Ludwig-Maximilians-Universität München; Munich), and Karl Fürlinger (Ludwig-Maximilians-Universität München; Munich)

A Unified Programming Model for Time- and Data-Driven Embedded Applications 26
Gabriela Breaban (Eindhoven University of Technology), Sander Stuijk (Eindhoven University of Technology), and Kees Goossens (Eindhoven University of Technology)

An Improved One-to-All Broadcasting in Higher Dimensional Eisenstein-Jacobi Networks 34
Zaid Hussain (Kuwait University)

Collective I/O Performance on the Santos Dumont Supercomputer 45
André Ramos Carneiro (National Laboratory for Scientific Computing (LNCC)), Jean Luca Bez (Federal University of Rio Grande do Sul), Francieli Zanon Boito (Federal University of Rio Grande do Sul), Bruno Alves Fagundes (National Laboratory for Scientific Computing (LNCC)), Carla Osthoff (National Laboratory for Scientific Computing (LNCC)), and Philippe Olivier Alexandre Navaux (Federal University of Rio Grande do Sul)

Context-Aware Optimization for Energy-Efficient and QoS Wireless Body Area Networks with Human Dynamics 53
Da-Ren Chen (Department of Information Management National Taichung University of Science and Technology) and Ping-Feng Wang (Institute for Information Industry subsidized by the Ministry of Economy Affairs of R.O.C)

CSSMT: Compiler Based Software Simultaneous Multithreading (SMT) 60
Yuanfang Chen (University of Delaware), Qingchuan Shi (Advanced Micro Devices), and Xiaoming Li (University of Delaware)

Developing and Using a Geometric Multigrid, Unstructured Grid Mini-Application to Assess Many-Core Architectures 68
Andrew Owenson (University of Warwick), Steven Wright (University of Warwick), Richard Bunt (University of Warwick), Stephen Jarvis (University of Warwick), Yoon Ho (Rolls-Royce), and Matthew Street (Rolls-Royce)

Divisible Load Scheduling of Image Processing Applications on the Heterogeneous Star Network Using a new Genetic Algorithm 77
Sahar Nikbakht Aali (Iran University of Science and Technology), Hadi Shahriar Shahhosseini (Iran University of Science and Technology), and Nader Bagherzadeh (University of California)

Geo-Distributed BigData Processing for Maximizing Profit in Federated Clouds Environment 85
Thouraya Gouasmi (Faculty of Economics and Management of Sfax), Wajdi Louati (National School of Engineers of Sfax), and Ahmed Hadj Kacem (Faculty of Economics and Management of Sfax)

Improving Communication and Load Balancing with Thread Mapping in Manycore Systems 93
Eduardo H.M. Cruz (Federal University of Rio Grande do Sul), Matthias Diener (University of Illinois at Urbana-Champaign), Matheus S. Serpa (Federal University of Rio Grande do Sul), Philippe Olivier Alexandre Navaux (Federal University of Rio Grande do Sul), Laércio Pilla (Federal University of Santa Catarina), and Israel Koren (University of Massachusetts)

Integrating Learning, Optimization, and Prediction for Efficient Navigation of Swarms of Drones 101
Amin Majd (Abo Akademi University), Adnan Ashraf (Abo Akademi University), Elena Troubitsyna (Abo Akademi University), and Masoud Daneshtalab (Malardalen University)

Low Precision Deep Learning Training on Mobile Heterogeneous Platform 109
Olivier Valery (National Taiwan University), Pangfeng Liu (National Taiwan University), and Jan-Jan Wu (Academia Sinica)

Novel Application of Parallel Computing Techniques in Soft X-Rays Plasma Measurement Systems for the WEST Experimental Thermal Fusion Reactor 118
Rafa Krawczyk (Warsaw University of Technology), Pawe Linczuk (Warsaw University of Technology), Andrzej Wojeski (Warsaw University of Technology), Krzysztof Poniak (Warsaw University of Technology), Grzegorz Kasprowicz (Warsaw University of Technology), Wojciech Zabootny (Warsaw University of Technology), Micha Gaska (Warsaw University of Technology), Didier Mazon (CEA Cadarache), Axel Jardin (CEA), Tomasz Czarski (Institute of Plasma Physics and Laser Microfusion), Piotr Kolasiski (Warsaw University of Technology), Maryna Chernyshova (Institute of Plasma Physics and Laser Microfusion), Ewa Kowalska-Strzciwilk (Institute of Plasma Physics and Laser Microfusion), and Karol Malinowski (Institute of Plasma Physics and Laser Microfusion)

Parallel Simulation of Sinoatrial Node Cells Synchronization 126
Aurelio Nicolás Mata (Universidad Autónoma Metropolitana), Norma Pilar Castellanos Abrego (Universidad Autónoma Metropolitana), Graciela Roman Alonso (Universidad Autónoma Metropolitana), Miguel Alfonso Castro García (Universidad Autónoma Metropolitana), Gabriel López Garza (Universidad Autónoma Metropolitana), and José Rafael Godínez Fernández (Universidad Autónoma Metropolitana)

Parallelizable Strategy for the Estimation of the 3D Structure of Biological Macromolecules 134
Claudia Caudai (CNR), Monica Zoppè (CNR), Emanuele Salerno (CNR), Ivan Merelli (CNR), and Anna Tonazzini (CNR)

Quantifying the Interaction Between Structural Properties of Software and Hardware in the ARM Big.LITTLE Architecture 138
Srboljub Stepanovic (Åbo Akademi University), Georgios Georgakarakos (Åbo Akademi University), Simon Holmbacka (Åbo Akademi University), and Johan Lilius (Åbo Akademi University)

Reducing Message Latency and CPU Utilization in the CAF Actor Framework 145
Massimo Torquati (University of Pisa), Tullio Menga (ATS Advanced Technology Solutions S.p.A.), Tiziano De Matteis (University of Pisa), Daniele De Sensi (University of Pisa), and Gabriele Mencagli (University of Pisa)

Resizing of Heterogeneous Platforms and the Optimization of Parallel Applications 154
Moussa Beji (Faculty of Sciences of Tunis) and Sami Achour (Higher Institute of Applied Sciences and Technology of Sousse)

Scheduler Accelerator for TDMA Data Centers 162
Ioannis Patronas (National and Kapodistrian University of Athens), Nikolaos Gkatzios (National and Kapodistrian University of Athens), Vasileios Kitsakis (National and Kapodistrian University of Athens), Dionysios Reisis (National and Kapodistrian University of Athens), Konstantinos Christodoulopoulos (National Technical University of Athens), and Emmanouel Varvarigos (National Technical University of Athens)

SharP Data Constructs: Data Constructs to Enable Data-Centric Computing 170
Ferrol Aderholdt (Oak Ridge National Laboratory), Manjunath Gorentla Venkata (Oak Ridge National Laboratory), and Zachary Parchman (Tennessee Technological University)

Shrink or Substitute: Handling Process Failures in HPC Systems Using In-Situ Recovery 178
Rizwan A. Ashraf (Oak Ridge National Laboratory), Saurabh Hukerikar (Oak Ridge National Laboratory), and Christian Engelmann (Oak Ridge National Laboratory)

Social Auto-Scaling 186
Peter Smith (National College of Ireland), Horacio González-Vélez (National College of Ireland), and Simon Caton (National College of Ireland)

Solving Sparse Triangular Linear Systems in Modern GPUs: A Synchronization-Free Algorithm 196
Ernesto Dufrechou (Instituto de Computación - Universidad de la República) and Pablo Ezzatti (Instituto de Computación - Universidad de la República)

Stingray-HPC: A Scalable Parallel Seismic Raytracing System 204
Mohammad Alaul Haque Monil (University Of Oregon), Allen D. Malony (University Of Oregon), Doug Toomey (University Of Oregon), and Kevin Huck (University Of Oregon)

TMbarrier: Speculative Barriers Using Hardware Transactional Memory 214
Manuel Pedrero (Dept. Computer Architecture), Eladio Gutierrez (Dept. Computer Architecture), and Oscar Plata (Dept. Computer Architecture)

A Dynamic Multi-Core Multicast Approach for Delay and Delay Variation Multicast Routing 222
Hovhannes A. Harutyunyan (Concordia University) and Meghrig Terzian (Concordia University)

A Hybrid Parallel Implementation for the Maximum Flow Problem 229
Marco Aurélio Stefanes (UFMS) and Luiz Fernando Alvino (IFMS)

A New Execution Model for Improving Performance and Flexibility of CAPE 234
Van Long Tran (SAMOVAR), Éric Renault (SAMOVAR), Xuan Huyen Do (College of Science), and Viet Hai Ha (College of Education)

A Safe and User-Friendly Graphical Programming Model for Parallel Stream Processing 239
Stefan Sydow (Technische Universität Berlin), Mohannad Nabelsee (Technische Universität Berlin), Helge Parzyjegla (Universität Rostock), and Paula Herber (Technische Universität Berlin)

Accelerating Blockchain Search of Full Nodes Using GPUs 244
Shin Morishima (Keio University) and Hiroki Matsutani (Keio University)

Characterizing Memory-Latency Sensitivity of Sparse Matrix Kernels 249
Noboru Tanabe (Tokyo Institute of Technology) and Toshio Endo (Tokyo Institute of Technology)

Data-Layout Reorganization for an Efficient Intra-Node Assembly of a Spectral Finite-Element Method 255
Gauthier Sornet (BRGM and LIFO from Orléans University), Sylvain Jubertie (LIFO from Orléans University), Fabrice Dupros (BRGM), Florent De Martin (BRGM), Philippe Thierry (Intel), and Sébastien Limet (LIFO from Orléans University)

Distributed Heuristics for Optimizing Cohesive Groups: A Support for Clinical Patient Engagement in Social Network Analysis 259
Italo Zoppis (University of Milano Bicocca. Dep. Computer Science), Riccardo Dondi (University of Bergamo; Department of Letters), Davide Coppetti (ITC "B. Belotti"), Alessandro Beltramo (university of Milano Bicocca. Dep. Computer Science), and Giancarlo Mauri (University of Milano Bicocca. Dep Computer Science)

Efficient and Fast Approximate Consensus with Epidemic Failure Detection at Extreme Scale 267
Amogh Katti (University of Minnesota) and David J Lilja (University of Minnesota)

Evaluating and Optimizing OpenCL Base64 Data Unpacking Kernel with FPGA 273
Zheming Jin (Argonne National Laboratory), Iris Johnson (Northern Illinois University), and Hal Finkel (Argonne National Laboratory)

Evaluating the Effect of Multi-Tenancy Patterns in Containerized Cloud-Hosted Content Management System 278
Adekunbi A. Adewojo (University of Salford) and Julian M. Bass (University of Salford)

Evaluation of Time-Triggered Traffic in Time-Sensitive Networks Using the OPNET Simulation Framework 283
Maryam Pahlevan (University of Siegen) and Roman Obermaisser (University of Siegen)

Extending PluTo for Multiple Devices by Integrating OpenACC 288
Tim Süß (Johannes Gutenberg University Mainz), Tunahan Kaya (Johannes Gutenberg University Mainz), and Dustin Feld (Fraunhofer SCAI)

Heterogeneous Computing and Multi-Clustering Support Via Peer-To-Peer HPC 292
Bilal Fakih (LAAS-CNRS) and Didier El Baz (LAAS-CNRS)

Hybrid OpenMP-MPI Parallelism: Porting Experiments from Small to Large Clusters 297
Marco Ferretti (University of Pavia) and Luigi Santangelo (University of Pavia)

Improving Availability in Distributed Tuple Spaces Via Sharing Abstractions and Replication Strategies 302
Vitaly Buravlev (IMT School for Advanced Studies Lucca), Rocco De Nicola (IMT School for Advanced Studies Lucca), Alberto Lluch Lafuente (Technical University of Denmark), and Claudio Antares Mezzina (IMT School for Advanced Studies Lucca)

Increasing Efficiency in Parallel Programming Teaching 306
Marco Danelutto (Dept. Computer Science) and Massimo Torquati (Dept. Computer Science)

Lazy Allocation and Transfer Fusion Optimization for GPU-Based Heterogeneous Systems 311
Lu Li (Linköping University) and Christoph Kessler (Linköping University)

Local and Global Shared Memory for Task Based HPC Applications on Heterogeneous Platforms ... 316
Chao Liu (Northeastern University) and Miriam Leeser (Northeastern University)

Memory-Aware Tree Partitioning on Homogeneous Platforms 321
Changjiang Gou (LIP), Anne Benoit (LIP), and Loris Marchal (LIP)

MinVisited: A Message Routing Protocol for Delay Tolerant Network 325
Luis Veas-Castillo (Universidad de Santiago de Chile), Gabriel Ovando-Leon (Universidad de Santiago de Chile), Veronica Gil-Costa (Universidad Nacional de San Luis), and Mauricio Marin (CeBiB)

Optimizing Machine Learning Algorithms on Multi-Core and Many-Core Architectures Using Thread and Data Mapping 329
Matheus S. Serpa (Federal University of Rio Grande do Sul), Arthur M. Krause (Federal University of Rio Grande do Sul), Eduardo H.M. Cruz (Federal University of Rio Grande do Sul), Philippe Olivier Alexandre Navaux (Federal University of Rio Grande do Sul), Marcelo Pasin (University of Neuchâtel), and Pascal Felber (University of Neuchâtel)

Performance Evaluation of the Metadata-Driven MASi Research Data Management Repository Service 334
Richard Grunzke (Center for Information Services and High Performance Computing), Volker Hartmann (Institute for Data Processing and Electronics), Thomas Jejkal (Institute for Data Processing and Electronics), Helen Kollai (Monitoring of Settlement and Open Space Development), Christiane Dressler (Digitale Akademie), Julia Dolhoff (Digitale Akademie), Julia Stanek (Institut für Anorganische Chemie), Hendrik Herold (Monitoring of Settlement and Open Space Development), Alexander Hoffmann (Institut für Anorganische Chemie), Ralph Müller-Pfefferkorn (Center for Information Services and High Performance Computing), Torsten Schrade (Digitale Akademie), Sonja Herres-Pawlis (Institut für Anorganische Chemie), Gotthard Meinel (Monitoring of Settlement and Open Space Development), and Wolfgang E. Nagel (Center for Information Services and High Performance Computing)

Predicting the Price of Bitcoin Using Machine Learning 339
Sean McNally (National College of Ireland), Jason Roche (Dublin Business School), and Simon Caton (National College of Ireland)

SAWS: Simple and Adaptive Warp Scheduling for Improved Performance in Throughput Processors 344
Francisco Muñoz Martinez (Universidad de Murcia) and Manuel E. Acacio (Universidad de Murcia)

Scalable Mapping of Streaming Applications onto MPSoCs Using Optimistic Mixed Integer Linear Programming 348
Neela Gayen (Queensland University of Technology), Johannes Ax (Bielefeld University), Martin Flasskamp (Bielefeld University), Christian Klarhorst (Bielefeld University), Thorsten Jungeblut (Bielefeld University), Maolin Tang (Queensland University of Technology), and Wayne Kelly (Queensland University of Technology)

Utilizing Heterogeneous Memory Hierarchies in the PGAS Model 353
Roger Kowalewski (LMU Munich), Tobias Fuchs (LMU Munich), Karl Fürlinger (LMU Munich), and Tobias Guggemos (LMU Munich)

Value-Based Allocation of Docker Containers 358
Piotr Dziurzanski (University of York) and Leandro Soares Indrusiak (University of York)

Variable Batched DGEMM 363
Pedro Valero-Lara (BSC), Ivan Martínez-Pérez (BSC), Sergi Mateo (BSC), Raül Sirvent (BSC), Vicenç Beltran (BSC), Xavier Martorell (BSC), and Jesús Labarta (UPC)

GPU Computing and Many Integrated Core Computing

Accelerating the RICH Particle Detector Algorithm on Intel Xeon Phi 368
Christina Quast (CERN/TU Berlin), Angela Pohl (TU Berlin), Biagio Cosenza (TU Berlin), Juurlink Ben (TU Berlin), and Rainer Schwemmer (CERN)

Closed-Form Solutions for Dense Matrix-Matrix Multiplication on Heterogeneous Platforms Using Divisible Load Analysis 376
Gerassimos Barlas (American University of Sharjah) and Lamees El Hiny (American University of Sharjah)

Fast Blocking of Householder Reflectors on Graphics Processors 385
Andrés E. Tomás Dominguez (Depto. de Ingenieria y Ciencia de Computadores) and Enrique S. Quintana Orti (Depto. de Ingenieria y Ciencia de Computadores)

Studying Victim Caches in GPUs 394
Eric M.Taylor (Rose-Hulman Institute of Technology) and Daniel W.Chang (Rose-Hulman Institute of Technology)

Advances in High-Performance Bioinformatics and Biomedicine

Combining Parallel Genetic Algorithms and Machine Learning to Improve the Research of Optimal Vaccination Protocols 399
Marzio Pennisi (University of Catania), Giulia Russo (University of Catania), and Francesco Pappalardo (University of Catania)

Computing Empirical P-Values for Estimating Gene-Gene Interactions in Genome-Wide Association Studies: A Parallel Computing Approach 406
Valentina Giansant (CNR-ITB), Daniele D'Agostino (CNR-IMATI), Carlo Maj (IRCCS Istituto Centro San Giovanni di Dio Fatebenefratelli), Stefano Beretta (CNR-ITB), and Ivan Merelli (CNR-ITB)

GPU-Accelerated Differential Dependency Network Analysis 410
Gil Speyer (The Translational Genomics Research Institute), Juan Rodriguez (The Translational Genomics Research Institute), Tomas Bencomo (The Translational Genomics Research Institute), and Seungchan Kim (Prairie View A&M University)

ParallelHashClone: A Parallel Implementation of HashClone Suite for Clonality Assessment from NGS Data 415
Greta Romano (University of Turin), Elisa Genuardi (University of Turin), Raffaele Calogero (University of Turin), and Simone Ferrero (University of Turin)

ParallNormal: An Efficient Variant Calling Pipeline for Unmatched Sequencing Data 423
Laura Follia (University of Turin), Fabio Tordini (University of Turin), Simone Pernice (University of Turin), Greta Romano (University of Turin), Giulia Beatrice Piaggeschi (University of Turin), and Giulio Ferrero (University of Turin)

Security in Parallel, Distributed and Network-Based Computing

A Secure Distributed Framework for Agglomerative Hierarchical Clustering Construction 430
Mona Hamidi (Universita di Siena), Mina Sheikhalishahi (Consiglio Nazionale delle Ricerche (CNR)), and Fabio Martinelli (Consiglio Nazionale delle Ricerche (CNR))

An Approach for Securing Critical Applications in Untrusted Clouds 436
Luigi Coppolino (University of Naples Parthenope), Salvatore D'Antonio (University of Naples Parthenope), Giovanni Mazzeo (University of Naples Parthenope), Gaetano Papale (University of Naples Parthenope), Luigi Sgaglione (University of Naples Parthenope), and Ferdinando Campanile (Synclab srl)

Business Model of a Botnet 441
C.G.J. Putman (University of Twente), _ Abhishta (University of Twente), and Lambert J. M. Nieuwenhuis (University of Twente)

CARS: Context Aware Reputation Systems to Evaluate Vehicles' Behaviour 446
Gianpiero Costantino (IIT-CNR), Fabio Martinelli (IIT-CNR), Ilaria Matteucci (IIT-CNR), Antonia Bertolino (ISTI-CNR), Antonello Calabro (ISTI-CNR), and Eda Marchetti (ISTI-CNR)

Enabling Dynamic and Efficient Data Access Control in Cloud Computing Based on Attribute Certificate Management and CP-ABE 454
Somchart Fugkeaw (Thai Digital ID Co.) and Hiroyuki Sato (The University of Tokyo)

Parallelization of Security Event Correlation Based on Accounting of Event Type Links 462
Igor Kotenko (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)), Andrey Fedorchenko (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)), Igor Saenko (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)), and Alexey Kushnerevich (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS))

The Multi-Layer Graph Based Technique for Proactive Automatic Response Against Cyber Attacks 470
Elena Doynikova (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)) and Igor Kotenko (St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS))

Energy Efficient Management of Parallel Systems, Platforms, and Computations

DuoFS: A Hybrid Storage System Balancing Energy-Efficiency, Reliability, and Performance 478
Shu Yin (ShanghaiTech University), Bing Jiao (Hunan University), Xiaomin Zhu (National University of Defense Technology), Xiaojun Ruan (California State University), Si Chen (West Chester University), and Zhuo Tang (Hunan University)

Energy-Efficient Actor Execution for SDF Application on Heterogeneous Architectures 486
Hergys Rexha (Åbo Akademi University and Turku Centre for Computer Science), Sébastien Lafond (Åbo Akademi University and Turku Centre for Computer Science), and Karol Desnos (IETR)

Exploring Energy Efficiency Model Generalization on Multicore Embedded Platforms 494
Hergys Rexha (Abo Akademi University) and Sébastien Lafond (Abo Akademi University)

How do Loop Transformations Affect the Energy Consumption of Multi-Threaded Runge-Kutta Methods? 499
Thomas Rauber (University Bayreuth) and Gudula Rünger (Chemnitz University of Technology)

Performance and Energy Consumption Analysis of Coprocessors Using Different Programming Models 508
Robson Gonçalves (Universidade Federal do Pampa), Alessandro Girardi (Universidade Federal do Pampa), and Claudio Schepke (Universidade Federal do Pampa)

Cloud Computing on Infrastructure as a Service and Its Applications

An Adaptive Cloud-Based IoT Back-end Architecture and Its Applications 513
Attila Csaba Marosi (MTA SZTAKI), Attila Farkas (MTA SZTAKI), and Robert Lovas (MTA SZTAKI)

Cost-Effective Reconfiguration for Multi-Cloud Applications 521
Nikos Parlavantzas (Univ Rennes), Linh Manh Pham (Univ Rennes), Arnab Sinha (CEA Saclay), and Christine Morin (Univ Rennes)

Getmewhere: A Location-Based Privacy-Preserving Information Service 529
Giampaolo Bella (Università degli Studi di Catania), Francesco Marino (Scuola Superiore Sant'Anna), Gianpiero Costantino (Consiglio Nazionale delle Ricerche), and Fabio Martinelli (Consiglio Nazionale delle Ricerche)

GPU Enabled Serverless Computing Framework 533
Tae Joon Jun (Korea Advanced Institute of Science and Technology), Daeyoun Kang (Korea Advanced Institute of Science and Technology), Dohyeun Kim (Korea Advanced Institute of Science and Technology), and Daeyoung Kim (Korea Advanced Institute of Science and Technology)

Saving Energy for Cloud Applications in Mobile Devices Using Nearby Resources 541
Anas Toma (TU Dortmund University), Alexander Starinow (TU Dortmund University), Jan Eric Lenssen (TU Dortmund University), and Jian-Jia Chen (TU Dortmund University)

High Performance Computing in Modelling and Simulation

Analysis of the Impact Factors on Data Error Propagation in HPC Applications 546
Gladys Utrera (Universitat Politècnica de Catalunya. BarcelonaTECH), Marisa Gil (Universitat Politècnica de Catalunya. BarcelonaTECH), and Xavier Martorell (Universitat Politècnica de Catalunya. BarcelonaTECH)

Cellular Automata Modelling of the Movement of People with Disabilities during Building Evacuation 550
Panagiota Kontou (Democritus University of Thrace), Ioakeim G. Georgoulas (Democritus University of Thrace), Giuseppe A. Trunfio (Democritus University of Thrace), and Georgios Ch. Sirakoulis (University of Sassari)

Clustering Goes Big: CLUBS-P, an Algorithm for Unsupervised Clustering Around Centroids Tailored For Big Data Applications 558
Michele Ianni (DIMES-UNICAL), Elio Masciari (ICAR-CNR), Giuseppe Massimo Mazzeo (UCLA), and Carlo Zaniolo (UCLA)

Exploiting Task-Based Parallelism for Parallel Discrete Event Simulation 562
Yizhuo Wang (Beijing Institute of Technology), Zhiwei Gao (Beijing Institute of Technology), Weixing Ji (Beijing Institute of Technology), Han Zhang (Beijing Simulation Center), and Duzheng Qing (Beijing Simulation Center)

High Performance Computation for the Multi-Parameterized Edit Distance 567
Francesco Cauteruccio (University of Calabria), Davide Consalvo (University of Calabria), and Giorgio Terracina (University of Calabria)

Low-Effort Task Distribution of Stencil Computation on Heterogeneous Multi-GPUs: Simulating Graphene Superlattices 575
Manuel Rodrigues (Instituto de Telecomunicações), David Fernandes (Instituto de Telecomunicações), Mário Silveirinha (Instituto de Telecomunicações), and Gabriel Falcão (Instituto de Telecomunicações)

Multi-BSP vs. BSP: A Case of Study for Dell AMD Multicores 579
Guillermo Trabes (Universidad Nacional de San Luis), Veronica Gil-Costa (Universidad Nacional de San Luis), Marcela Printista (Universidad Nacional de San Luis), and Mauricio Marin (Universidad de Santiago de Chile)

Structured Grid-Based Parallel Simulation of a Simple DEM Model on Heterogeneous Systems 588
Alessio De Rango (University of Calabria), Pietro Napoli (University of Calabria), Donato D'Ambrosio (University of Calabria), William Spataro (University of Calabria), Alberto Di Renzo (University of Calabria), and Francesco Di Maio (University of Calabria)

Task Parallelism in the WRF Model Through Computation Offloading to Many-Core Devices 596
Rodrigo Bayá (Facultad de Ingeniería), Claudio Porrini (Facultad de Ingeniería), Martín Pedemonte (Facultad de Ingeniería), and Pablo Ezzatti (Facultad de Ingeniería)

On-Chip Parallel and Network-Based Systems

Memory-Aware Genetic Algorithms for Task Mapping on Hard Real-Time Networks-on-Chip 601
Lloid Robert Still (University of York) and Leandro Soares Indrusiak (University of York)

Performability Analysis of Mesh-Based NoCs Using Markov Reward Model 609
Jie Hou (University of Stuttgart) and Martin Radetzki (University of Stuttgart)

RVNoC: A Framework for Generating RISC-V NoC-Based MPSoC 617
Mahmoud A. Elmohr (Alexandria University), Ahmed S. Eissa (Alexandria University), Moamen Ibrahim (University of Oulu), Mostafa Khamis (Mentor Graphics), Sameh El-Ashry (Ain Shams University), Ahmed Shalaby (Benha University), Mohamed AbdElsalam (Mentor Graphics), and M. Watheq El-Kharashi (Ain Shams Universit)

Simulation-Based Evaluation Strategy for Task Mapping Approaches in WNoC Platforms 622
Luis Germán García Morales (University of Antioquia), José Edinson Aedo Cobo (University of Antioquia), and Nader Bagherzadeh (University of California)

Storage Architectures and Data Transfer Systems for Bigdata and Exascale Computing

Endolith: A Blockchain-Based Framework to Enhance Data Retention in Cloud Storages 627
Thomas Renner (Technische Universität Berlin), Johannes Müller (Technische Universität Berlin), and Odej Kao (Technische Universität Berlin)

Low-Power Storage Bricks and Bioinformatics on Systems-On-Chip 635
Lucia Morganti (CNAF - Italian Institute for Nuclear Physics), Daniele Cesini (CNAF - Italian Institute for Nuclear Physics), Elena Corni (CNAF - Italian Institute for Nuclear Physics), Luca Lama (CNAF - Italian Institute for Nuclear Physics), Carmelo Pellegrino (CNAF - Italian Institute for Nuclear Physics), Ivan Merelli (Institute for Biomedical Technologies - National Research Council of Italy), and Daniele D'Agostino (Institute for Applied Mathematics and Information Technologies "E. Magenes'')

O-TF and O-FTF, Optical Fault-Tolerant DCNs 639
Saeedeh Akbari Rokn Abadi (Sharif University of Technology) and Somayyeh Koohi (Sharif University of Technology)

pWebDAV: A Multi-Tier Storage System 643
Christos Filippidis (National and Kapodistrian University of Athens) and Yiannis Cotronis (National and Kapodistrian University of Athens)

Storage for Advanced Scientific Use-Cases and Beyond 651
Paul Millar (Deutsches Elektron-Synchrotron (DESY)), Olufemi Adeyemi (Deutsches Elektron-Synchrotron (DESY)), Gerd Behrmann (Nordic e-Infrastructure Collaboration (NeIC)), Patrick Fuhrmann (Deutsches Elektron-Synchrotron (DESY)), Vincent Garonne (Nordic e-Infrastructure Collaboration (NeIC)), Dmitry Litvinsev (Fermilab National Laboratory), Tigran Mkrtchyan (Deutsches Elektron-Synchrotron (DESY)), Albert Rossi (Fermilab National Laboratory), Marina Sahakyan (Deutsches Elektron-Synchrotron (DESY)), and Jürgen Starek (Deutsches Elektron-Synchrotron (DESY))

High Performance Computing for Neuroscience

Gaussian and Exponential Lateral Connectivity on Distributed Spiking Neural Network Simulation 658
Elena Pastorelli (INFN Sezione di Roma and PhD Program in Behavioural Neuroscience), Pier Stanislao Paolucci (INFN Sezione di Roma), Francesco Simula (INFN Sezione di Roma), Andrea Biagioni (INFN Sezione di Roma), Fabrizio Capuani (INFN Sezione di Roma), Paolo Cretaro (INFN Sezione di Roma), Giulia De Bonis (INFN Sezione di Roma), Francesca Lo Cicero (INFN Sezione di Roma), Alessandro Lonardo (INFN Sezione di Roma), Michele Martinelli (INFN Sezione di Roma), Luca Pontisso (INFN Sezione di Roma), Piero Vicini (INFN Sezione di Roma), and Roberto Ammendola (INFN Sezione di Tor Vergata and Electronic Engineering Dept.)

Implementation of Bayesian Inference In Distributed Neural Networks 666
Zhaofei Yu (School of Electronics Engineering and Computer Science), Tiejun Huang (School of Electronics Engineering and Computer Science), and Jian K. Liu (Institute for Theoretical Computer Science)

High Performance Computing in Astronomy and Astrophysics

ECHO-3DHPC: Relativistic Accretion Disks onto Black Holes 674
Matteo Bugli (Max Planck Institute for Astrophysics)

High-Resolution Numerical Relativity Simulations of Spinning Binary Neutron Star Mergers 682
Tim Dietrich (Max Planck Institute for Gravitational Physics (Albert Einstein Institute)), Sebastiano Bernuzzi (Universita di Parma), Bernd Bruegmann (University of Jena), and Wolfgang Tichy (Florida Atlantic University)

Parallel and Distributed High-Performance Computing Solutions in Systems Biology

Evaluation of Parallel Tempering to Accelerate Bayesian Parameter Estimation in Systems Biology 690
Sanjana Gupta (University of Pittsburgh School of Medicine), Liam Hainsworth (University of Pittsburgh School of Medicine), Justin Hogg (University of Pittsburgh School of Medicine), Robin Lee (University of Pittsburgh School of Medicine), and James Faeder (University of Pittsburgh School of Medicine)

GPU-Powered Multi-Swarm Parameter Estimation of Biological Systems: A Master-Slave Approach 698
Andrea Tangherloni (University of Milano-Bicocca), Leonardo Rundo (University of Milano-Bicocca), Simone Spolaor (University of Milano-Bicocca), Paolo Cazzaniga (University of Bergamo), and Marco S. Nobile (University of Milano-Bicocca)

Leveraging Compute Clusters for Large-Scale Parametric Screens of Reaction-Diffusion Systems 706
Md Shahriar Karim (Purdue University), Hans G. Othmer (University of Minnesota), and David M. Umulis (Purdue University)

Parallel Numerical Methods and Libraries for Heterogeneous Multi/Many-Cores

A High-Level C++ Approach to Manage Local Errors, Asynchrony and Faults in an MPI Application 714
Christian Engwer (University of Munster), Mirco Altenbernd (University of Stuttgart), Nils-Arne Dreier (University of Munster), and Dominik Göddeke (University of Stuttgart)

A Parallel Implementation of the Hestenes-Jacobi-One-Sides Method Using GPU-CUDA 722
Salvatore Cuomo (University of Naples Federico II), Livia Marcellino (University of Naples Parthenope), and Guglielmo Navarra (University of Naples Federico II)

CudaPre2D: A Straightforward Preprocessing Approach for Accelerating 2D Convex Hull Computations on the GPU 726
Gang Mei (China University of Geosciences Beijing) and Sixu Guo (China University of Geosciences Beijing)

Efficient NAS Benchmark Kernels with C++ Parallel Programming 733
Dalvan Griebler (Pontifical Catholic University of Rio Grande do Sul (PUCRS)), Junior Loff (Pontifical Catholic University of Rio Grande do Sul (PUCRS)), Gabriele Mencagli (University of Pisa), Marco Danelutto (University of Pisa), and Luiz Gustavo Fernandes (Pontifical Catholic University of Rio Grande do Sul (PUCRS))

Parallelizing and Optimizing LHCb-Kalman for Intel Xeon Phi KNL Processors 741
Plácido Fernández (CERN), David del Rio Astorga (University Carlos III of Madrid), Manuel F. Dolz (University Carlos III of Madrid), Javier Fernández (University Carlos III de Madrid), Omar Awile (CERN), and J. Daniel García (University Carlos III of Madrid)

Scaling Dense Linear Algebra on Multicore and Beyond: A Survey 751
Paolo Viviani (University of Turin; Noesis Solutions NV), Maurizio Drocco (University of Turin), and Marco Aldinucci (University of Turin)

Solving Multiple Tridiagonal Systems on a Multi-GPU Platform 759
Adrián Pérez Diéguez (University of A Coruña), Margarita Amor López (University of A Coruña), and Ramón Doallo Biempica (University of A Coruña)

Author Index **765**

A new execution model for improving performance and flexibility of CAPE

Van Long Tran*†, Éric Renault*, Xuan Huyen Do‡ and Viet Hai Ha§
*SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay
9 rue Charles Fourier - 91011 Évry Cedex, France
Email: { van_long.tran, eric.renault }@telecom-sudparis.eu
† Hue Industrial College, Hue, Vietnam
‡ College of Science, Hue University, Vietnam
§College of Education, Hue University, Vietnam

***Abstract*—Checkpointing-Aided Parallel Execution (CAPE) is a framework that is based on checkpointing technique and serves to automatically translates and execute OpenMP programs on distributed-memory architectures. Through some comparisons with MPI, CAPE has demonstrated high-performance and the potential for fully compatibility with OpenMP on distributed-memory systems. However, there is still a long way to improve the performance, flexibility, portability and capability. This paper presents the new execution model for CAPE that improves its performance and makes CAPE even more flexible.**

***Index Terms*—CAPE; Checkpointing; OpenMP; HPC; Distributed Computing; Parallel computing;**

I. Introduction

OpenMP [1] and MPI [2] are the two main standard parallel programming tools to develop parallel programs. While the former is a tool for shared-memory, the later has been developed for distributed-memory architectures. From the programmer's point of view, OpenMP is easier-to-use than MPI. Indeed, OpenMP requests the programmer to tag some pieces of code into sequential programs to be executed automatically in parallel, while MPI requires them to rewrite these programs and specifies clearly which pieces of code can be executed in parallel and explicit which data must be synchronized between processes. Therefore, many efforts have been made to port OpenMP on distributed-memory architectures. However, apart from CAPE [10], [11], no solution successfully met the two following requirements: 1) to be fully compliant with the OpenMP standard and 2) high performance. Most prominent approaches include the use of an SSI [3], SCASH [4], the use of the RC model [5], performing a source-to-source translation to a tool like MPI [6], [7] or Global Array [8], or Cluster OpenMP [9].

CAPE provides a set of prototypes and functions to automatically translate OpenMP programs for distributed memory architectures and make them ready for execution in parallel. The main idea of this solution is using incremental checkpoint techniques (ICKPT) [12], [13] to distribute the parallel jobs and their data to other processes (the fork phase of OpenMP), and collect the results after the execution of the jobs from all processors (the join phase of OpenMP). ICKPT is also used to deal with the exchange of shared data automatically.

Although CAPE is still under development, it has shown its ability to provide very efficient code. For instance, a comparison with MPI showed that CAPE is able to reach up to 90% of the MPI performance [14], [15]. This has to be balanced with the fact that CAPE for OpenMP requires the introduction of few `pragma` directives only in the sequential code, ie. no complex code from the user point of view, while writing an MPI code might require the user to completely refactorise the code. Moreover, as compared to other OpenMP for distributed-memory solutions, CAPE promises a fully compatibility with OpenMP [11], [14].

This paper introduces a new execution model that improves the performance and the flexibility of CAPE. The paper is organized as follows: the next section describes CAPE mechanisms, capabilities and restrictions. Then, section III presents the design and the implementation of the new execution model. Section IV shows analysis and the result of the experiments. Finally, section V concludes the paper and presents our future works to improve CAPE.

II. CAPE Principles

CAPE uses a set of templates to translate OpenMP source code into CAPE source code. Then, the CAPE source code is compiled using a traditional C/C++ compiler. At last, the binary code can be executed independently on any distributed-memory system supporting CAPE framework. The different steps of the CAPE compilation process for C/C++ OpenMP programs is shown in Fig. 1.

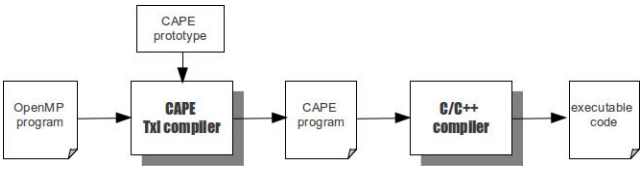

Fig. 1. Translation of OpenMP programs with CAPE.

A. Execution model

The CAPE execution model uses checkpoints to implement the OpenMP fork-join model. This mechanism is shown in

0-7695-6380-5/18/$31.00 ©2018 IEEE
DOI 10.1109/PDP2018.2018.00039

Fig. 2. To execute a CAPE program on a distributed-memory architecture, this program first runs on a set of nodes, each node executes as a process. Whenever the program meets a parallel section, the master node distributes the jobs among the slave processes using the Discontinuous Incremental Checkpoints (DICKPT) [13], [14] mechanism. Through this approach, the master node generates DICKPTs and sends them to the slave nodes, each slave node receiving a single checkpoint. After sending checkpoints, the master node waits for the results returned from the slaves. The next step is different depending upon the nature of the node: the slave nodes receive their checkpoint, inject it into their memory, execute their part of the job, and sent back the result to the master node by using another DICKPT; the master node waits for all returned results in form of DICKTs and then merge them before injection into its memory. At the end of the parallel region, the master sends the merged checkpoint to every slaves to synchronize the memory space of the whole program.

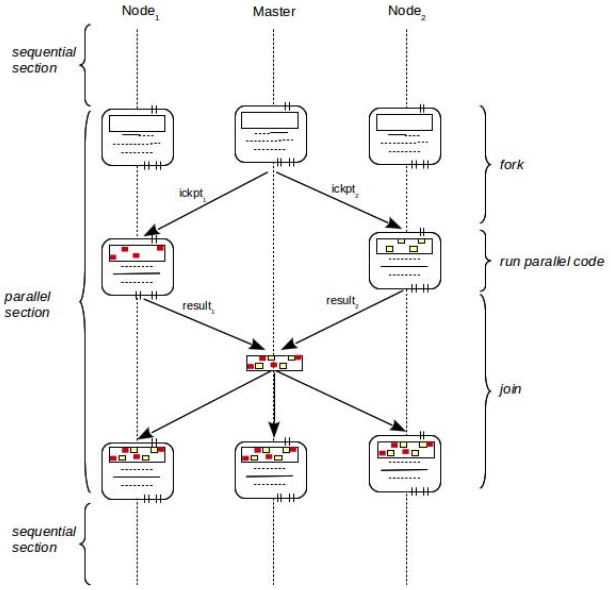

Fig. 2. CAPE execution model.

B. Translation from OpenMP to CAPE

In the CAPE framework, a set of functions has been defined and implemented to perform the tasks devoted to DICKPT, typically distributing checkpoints, sending/receiving checkpoints, extracting/injecting a checkpoint from/to the program's memory, etc. Besides, a set of templates has been defined in the CAPE compiler to perform the translation of the OpenMP program into the CAPE program automatically and makes it executable in the CAPE framework. At the present, nested loops and shared-data variable constructs are not supported yet. However, this is not real issue as this can be solved at the level of the source-to-source translation and does not require any modification in the CAPE philosophy. To this end, CAPE can only be applied to OpenMP programs matching the Bernstein's conditions.

Fig. 3 presents an example of code substitution for the specific case of the `parallel for` construct. This example is typical of those we implemented for the other constructs [7]. The automatically generated code is based on the following functions that are part of the CAPE framework:

```
# pragma omp parallel for
  for ( A ; B ; C )
      D ;
```

↓ automatically translated into ↓

```
if ( master ( ) )
   start ( )
   for ( A ; B ; C )
        create ( before )
        send ( before, slave_x )
   create ( final )
   stop ( )
   wait for ( after )
   inject ( after )
   if ( ! last parallel ( ) )
           merge ( final, after )
           broadcast ( final )
else
   receive ( before )
   inject ( before )
   start ( )
   D
   create ( after_i )
   stop ( )
   send ( after_i, master )
   if ( ! last parallel ( ) )
           receive ( final )
           inject ( final )
   else
           exit
```

Fig. 3. Template for the `parallel for` with incremental checkpoints.

C. Remarks

The good performance of CAPE as compared to those of MPI and the full compliance to the OpenMP specifications [11], [14] have made CAPE a good alternative to port OpenMP on distributed-memory architectures. So far, the implementation of CAPE is not complete, some disadvantages can be listed:

1) As shown in Fig. 2, the master node might act as a bottleneck while waiting for checkpoints from the salves,

merging checkpoints and/or sending back data to slaves for memory synchronization.

2) To distribute jobs to slaves, the master node generates a number of checkpoints that depends upon the number of slave nodes and so that each slave node receives a checkpoint (see Fig. 5). This method can reach a high-level of optimization. However, it might not be enough flexible for some cases like 1) the number of slaves may not be identified at compile time, 2) the OpenMP source code should be modified to detect when the master generates the checkpoint and 3) the dynamic scheduling of OpenMP cannot be implemented using this method.
3) After distributing the jobs, the slave nodes execute the divided jobs while the master does not do anything until the reception of the resulting checkpoints from the slaves, which clearly wastes resources.

III. A NEW EXECUTION MODEL FOR CAPE

In order to improve the performance and the flexibility of CAPE, we designed a new execution model that extends the one presented in Sec. II-A. Fig. 4 illustrates the model that can be described as follows:

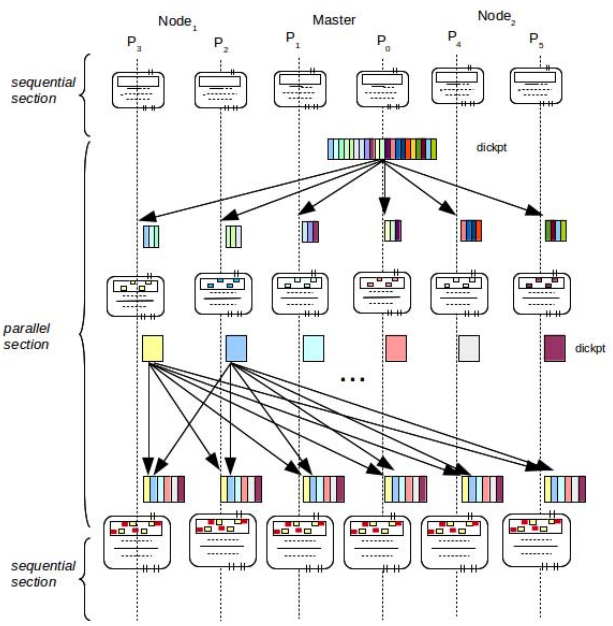

Fig. 4. The new execution model for CAPE.

1) At the beginning of the program, all nodes in the system execute the same sequential code.
2) When a parallel region is reached, the master process creates a set of incremental checkpoints. The number of incremental checkpoints depends upon the number of tasks in the parallel region. Each incremental checkpoint contains the state of the program to be used to resume its execution in an another process at the saved time.
3) The master process scatters the set of incremental checkpoints. Each node receives some of the checkpoints generated by the master process. This step is illustrated in Fig. 6.
4) The received incremental checkpoints are injected into the slave processes' memories.
5) The slave processes resume their execution.
6) Results on slave processes are extracted by identifying the modified regions and recorded in an incremental checkpoint.
7) Incremental checkpoints of each process is sent back to the master node. Incremental checkpoints are combined altogether to generate a single checkpoint. This step can be distributed among the processes if need be.
8) The final combined incremental checkpoint is injected in the master process' memory and the master process can resume its execution.

Changing the execution model implies changing the translation templates. Fig. 7 presents the template for the *#pragma omp parallel for* directive that adapts to the new execution model. The other OpenMP directives can be designed in a similar way. For this template, CAPE functions are as follows:

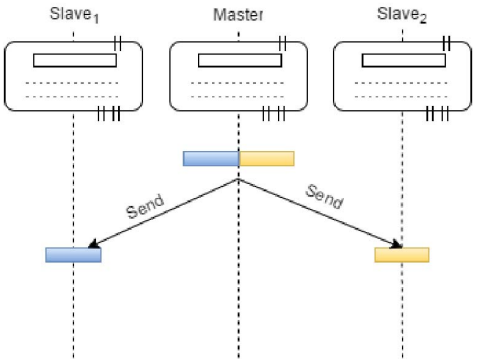

Fig. 5. Former scheduling method in CAPE

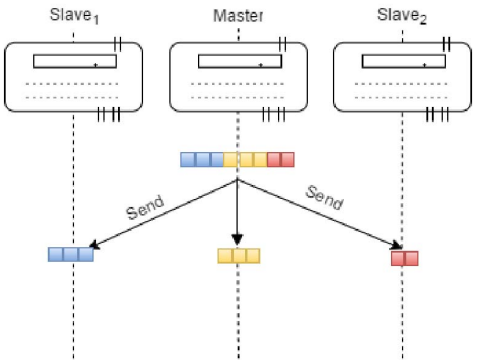

Fig. 6. Scheduling method with the new execution model

- $generate_dickpt(before_i)$ (line 3): at each loop iteration related to a task, the master process generates an incremental checkpoint.
- $scatter(before, \&recv_n, master)$ (line 4): the master process scatters the checkpoints to the available processes, including itself. Each process receives some of the checkpoints ($recv_n$).

```
# pragma omp parallel for
  for ( A ; B ; C )
      D ;
```

↓ automatically translated into ↓

```
1 if ( master ( ) )
2   for (A ; B ; C)
3     generate_dickpt (before_i)
4 scatter(before, &recv_n ,master)
5 inject(recv_n )
6 D
7 generate_dickpt (after_n)
8 allreduce(after_n,&after,[<ops>])
9 inject(after)
```

Fig. 7. Prototype for the *parallel for* for new model.

- $inject(recv_n)$ (line 5): each checkpoint is injected into the target process' memory.
- the execution is resumed on instruction D (line 6).
- $generate_dickpt(after_n)$ (line 7): each process generates an incremental checkpoint that saves the result of its execution.
- $allreduce(after_n, \&after, [< \ ops \ >])$ (line 8): the $after_n$ checkpoint of process n is sent to the other processes. Checkpoints are combined, computed and saved in a new $after$ checkpoint. This is performed using the Recursive Doubling algorithm [16] as illustrated in Fig. 8.

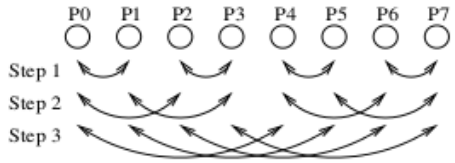

Fig. 8. Recursive doubling for allreduce.

- $inject(after)$ (line 9): incremental checkpoint $after$ is injected into the application's memory to synchronize the state of the program on all nodes.

IV. Analysis and Evaluation

Moving from a scheduling of CAPE processes based on the number of nodes (Fig. 5) to a scheduling based on the number of jobs (Fig. 6) makes CAPE more flexible at least for the three following reasons:

1) The number of available processes can be identified at runtime. The master node can distribute the jobs to all available processes.
2) All OpenMP scheduling mechanisms such as `static` and `dynamic` can be implemented on CAPE as the master node generates a number of checkpoints depending on the number of jobs. First step, one checkpoint can be sent to each slave node to execute a divided job. When the slave node finishes, it sends the next checkpoint that has not been executed yet.
3) All nodes of the system, including the master node, share the calculate tasks of the parallel section. Therefore, resource of master node is better exploited while comparing with the previous executing model of CAPE.

Considering that both initial and sequential codes of the program are executed in the same way in any processes of the system, only the execution time of the parallel regions has been considered. For the new execution model, the Recursive Doubling algorithm [16] is applied to communicate between all processes and the master node is counted in to executed divided jobs. This means that the execution time for CAPE is reduced with this new execution model and the resources are used more efficiently.

In order to measure the impact of the new execution model on the performance, some experiments were conducted. These experiments were performed on 4-node and 16-node clusters. Each node includes an Intel core i3-2100, a dual-core 4-thread CPU running at 3.10 GHz and 2 GB of RAM. These computers are connected using a 100 Mb/s Ethernet network. To avoid external influence as much as possible, the entire system was dedicated to the tests during all the performance measurement campaign.

The program used as the basis for these experiments is the classic matrix-matrix multiplication. The sizes of the matrix are increased from 1600×1600 to 9600×9600. Each program executed at least 10 times to measure the total execution times and a confidence interval of at least 98% has been always achieved for the measures. Data reported here are the means of the 10 measures.

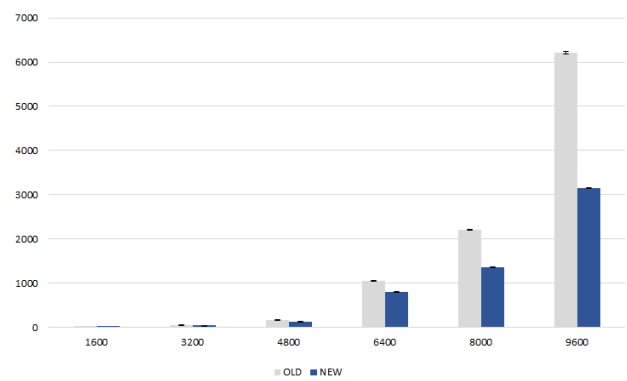

Fig. 9. Total execution time (in seconds) of two models on 4-node clusters.

Fig. 9 and Fig. 10 present the total execution time of CAPE on 4-node and 16-node clusters respectively. As can be seen from these figures, the execution time of both models are shown, the gray color (OLD) is represented for the previous execution model, and the blue one (NEW) is represented for the new execution model. The horizontal axis shows the size

of the matrix and vertical axis shows the execution time in seconds.

For the 4-node cluster as compared with the previous model, the execution time of the new model is reduced significantly and the reduction is inversely proportional to the size of the matrix. The larger the size of matrix, the higher the improvement. This is due to the fact that there are only three nodes executing the divided jobs on the previous execution model. The master just divides and distributes jobs to slave, and then waits for the returned results. It does not participate in computational tasks. However,the new execution model is in contrast, after divided jobs, the master node receives and executes a part of divided jobs. Therefore the computation time (t_c) with this model is much lower than itself on previous model, especially, on the cluster which includes only 4 nodes.

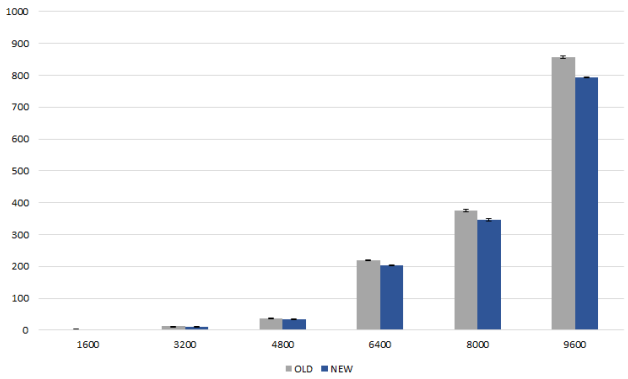

Fig. 10. Total execution time (in seconds) of two models on 16-node clusters.

For the 16-node cluster, the result in the Fig. 10 shows the same trend, the new execution model is better results than the previous one. However, the distance of execution time of the two models is closer. It maintains a saving time of around 10%. This is because the larger the number of nodes, the less time to compute the divided jobs. Therefore, the total saving time of t_c in this case is not too much.

Indeed, both the analysis and the experiments on clusters composed of 4 nodes and 16 nodes as comparing the previous execution model and the new execution model show that the resources of the system are used more efficiently and the execution time is significantly reduced (decreased at least by 10%). This shows that the new execution model is a good direction to pursue the development of CAPE in the future.

V. Conclusion and Future works

In this paper, we analysed the drawbacks of the previous execution model of CAPE and proposed a new one to solve them. The new scheduling method and a new algorithm for collecting results have been developed and presented to improve the flexibility of CAPE. In addition, both theoretical analysis and experimentations showed that the execution time and the risk of bottlenecks are significantly reduced.

For the future works, we will keep on developing CAPE using this new execution model. We will use the operations on checkpoints to implement OpenMP's shared-data environment variables.

References

[1] OpenMP ARB, "OpenMP application program interface version 4.0," 2013.

[2] MPI Forum, "Message passing interface forum."

[3] C. Morin, R. Lottiaux, G. Vallée, P. Gallard, G. Utard, R. Badrinath, and L. Rilling, "Kerrighed: a single system image cluster operating system for high performance computing," in *Euro-Par 2003 Parallel Processing*. Springer, 2003, pp. 1291–1294.

[4] M. Sato, H. Harada, A. Hasegawa, and Y. Ishikawa, "Cluster-enabled OpenMP: An OpenMP compiler for the SCASH software distributed shared memory system," *Scientific Programming*, vol. 9, no. 2, 3, pp. 123–130, 2001.

[5] S. Karlsson, S.-W. Lee, and M. Brorsson, "A fully compliant OpenMP implementation on software distributed shared memory," in *High Performance ComputingHiPC 2002*. Springer, 2002, pp. 195–206.

[6] A. Basumallik and R. Eigenmann, "Towards automatic translation of OpenMP to MPI," in *Proceedings of the 19th annual international conference on Supercomputing*. ACM, 2005, pp. 189–198.

[7] A. J. Dorta, J. M. Badía, E. S. Quintana, and F. de Sande, "Implementing OpenMP for clusters on top of MPI," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Springer, 2005, pp. 148–155.

[8] L. Huang, B. Chapman, and Z. Liu, "Towards a more efficient implementation of OpenMP for clusters via translation to global arrays," *Parallel Computing*, vol. 31, no. 10, pp. 1114–1139, 2005.

[9] J. P. Hoeflinger, "Extending OpenMP to clusters," *White Paper, Intel Corporation*, 2006.

[10] É. Renault, "Distributed implementation of OpenMP based on checkpointing aided parallel execution," in *A Practical Programming Model for the Multi-Core Era*. Springer, 2007, pp. 195–206.

[11] V. H. Ha and E. Renault, "Improving performance of CAPE using discontinuous incremental checkpointing," in *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*. IEEE, 2011, pp. 802–807.

[12] J. S. Plank, M. Beck, G. Kingsley, and K. Li, *Libckpt: Transparent checkpointing under unix*. Computer Science Department, 1994.

[13] V. H. Ha and É. Renault, "Discontinuous incremental: A new approach towards extremely lightweight checkpoints," in *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*. IEEE, 2011, pp. 227–232.

[14] V. H. Ha and E. Renault, "Design and performance analysis of CAPE based on discontinuous incremental checkpoints," in *2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2011.

[15] V. L. Tran, E. Renault, and V. H. Ha, "Analysis and evaluation of the performance of CAPE," in *IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress. IEEE International Symposium on*. IEEE, 2016, pp. 620–627.

[16] R. Thakur, R. Rabenseifner, and W. Gropp, "Optimization of collective communication operations in MPICH," *International Journal of High Performance Computing Applications*, vol. 19, no. 1, pp. 49–66, 2005.