CERCIRAS Workshop-01: Abstracts

Editors: Padma Iyenghar and Gordana Rakic

September 1, 2021

Evaluating Performance and Reliability of Selective Redundant Multithreading for GPGPU Applications

Ercüment Kaya[1], Ömer Faruk Karadaş[2] and Işıl Öz[1]

[1]*Computer Engineering Department, Izmir Institute of Technology, Izmir, Turkey*
[2]*Electrical Electronics Engineering Department, Izmir Institute of Technology, Izmir, Turkey*

With the widespread use of GPU architectures in general-purpose computations, evaluating the soft error vulnerability of GPGPU programs and employing efficient fault tolerance techniques for more reliable execution becomes more prominent. Performing full redundancy, based on the redundant execution of the complete program, results in resource consumption and performance loss as well as energy inefficiency. Therefore, determining the most error-prone regions of the target program code and replicating only those parts maintains both high performance and acceptable error rates.

In this study, we propose a partial redundant multithreading mechanism based on the soft error vulnerability of GPGPU applications and perform a trade-off analysis between performance and reliability. Firstly, in our fault injection framework, we evaluate the most vulnerable code regions of the target applications by considering kernel functions. Then, based on the outcome of the fault injection experiments, we determine the kernel function to be replicated. According to the pragmas denoting the redundancy points in the source code, our compiler back-end generates the code that enables the redundant execution for the specified code region.

Our debugger-based regional fault injection tool generates fault injection points for each kernel function based on the information gathered during the profiling phase. It evaluates the silent data corruption (SDC) rates to get the soft error vulnerability of each kernel function of the target GPGPU program. Additionally, our LLVM-based compiler framework generates the target executable including redundant code sections for the specified kernel functions marked by the programmer, who utilizes the feedback from our fault injection analysis.

We perform an experimental study to reveal the efficiency of our approach for a set of GPGPU applications. Our fault injection experiments demonstrate that the code regions inside GPGPU programs exhibit different characteristics in terms of soft error vulnerability, pointing to a partial redundancy for both higher performance and reliability. Based on the recommendations of our soft error vulnerability analysis, we perform redundant executions that replicate only the most vulnerable parts of the target programs. We evaluate both the reliability and performance of the redundant execution scenarios by conducting fault injection experiments and measuring the execution time of the redundant program generated by our compiler-managed redundancy technique. Our results demonstrate that protecting only the most vulnerable kernel functions enables high reliability without hurting the performance significantly.

1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society (CERCIRAS)
ercumentkaya@iyte.edu.tr (E. Kaya); omerkaradas@std.iyte.edu.tr (Ö. F. Karadaş); isiloz@iyte.edu.tr (I. Öz)

CEUR Workshop Proceedings (CEUR-WS.org)

Analysing IoT Applications with DISSECT-CF-Fog in Simulated Fog and Cloud Environments

Andras Markus[1], Attila Kertesz[1]

[1]*Software Engineering Department, University of Szeged, Hungary*

Abstract

The Fourth Digital Revolution has created numerous challenges in the fields of Cloud Computing, Fog Computing and Internet of Things. These paradigms are often associated as the Cloud-to-Thing Continuum, where vast amounts of IoT data are analysed and stored in a multi-layered Fog-Cloud topology, in which geographically separated nodes are responsible for data processing. Such systems affect human life positively by offering various real time IoT applications. Nevertheless, by executing these applications, we have to consider the trade-offs of energy consumption, computational power usage, utilisation cost and other resource metrics. The emergence of simulation tools allows us to analyse such complex IoT-Fog-Cloud systems thoroughly, however the efficiency and comprehensive observation may require accurate and realistic models used in a simulation environment.

In this paper, we introduce the open-source DISSECT-CF-Fog simulator as a solution for investigating the most challenging problems arising from the management of these modern network systems. It provides fine-grained models with various parameters for IoT device and application behaviour, as well as data centre management with realistic network settings. The simulation tool is also capable of considering different algorithms for offloading of the computational nodes by taking into account energy consumption of the system entities, IoT device mobility, and pricing schemes of real providers and network properties as well. DISSECT-CF-Fog can also be a reasonable choice for large-scale experiments, especially where the number of active entities exceeds tens of thousands.

We also describe the usability of DISSECT-CF-Fog in a comprehensive manner by two of the typical IoT use cases. The first is a European-wide weather forecasting scenario, where IoT sensors (for instance humidity or temperature detectors) sample the local environment conditions and send the sensed data for further processing. To avoid the bottleneck effect of cloud resources and IoT application delays, they can be aided by resource constrained fog nodes. In the second scenario logistics is involved, where mobile devices are on the move continuously, and the system has to take into account the handover of devices, in order to foster service availability and reduce the overall latency of the data.

Our evaluation results show that the system characteristics have serious effects on the aforementioned trade-offs on operation costs, energy and resource utilisation. We provide greedy and more sophisticated algorithms for deeply analysing the behaviour of the use cases, and we point out that DISSECT-CF-Fog offers fundamental methods for fine-tuning the system parameters.

Keywords

Simulation, Fog Computing, Internet of Things

Acknowledgments

The research leading to these results has received funding from the European COST programme

1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society, COST Action CA19135 (CERCIRAS), September 2, 2021, Novi Sad, Serbia

markusa@inf.u-szeged.hu (A. Markus); keratt@inf.u-szeged.hu (A. Kertesz)

CEUR Workshop Proceedings (CEUR-WS.org)

Towards Blockchain-based Smart Systems

Hamza Baniata[1], Dragi Kimovski[2], Radu Prodan[2] and Attila Kertesz[1]

[1] *Software Engineering Department, University of Szeged, Hungary*
[2] *Institute of Information Technology, University of Klagenfurt, Austria*

Abstract

Nowadays, we are witnessing an unprecedented pace of technological development in smart systems, incorporating sensing, actuation, and control functions. The current smart services and applications have the following properties and needs: (*i*) they are interconnected and need scalable, virtualized resources to run, store and process data, (*ii*) they are mobile and can potentially access and build on user data made available by smartphones and tablets, and (*iii*) they are getting smarter, so they may get access to user data provided by connected smart devices. As the number of smart devices in smart systems grows, the vast amount of data they produce requires high-performance computational and storage services for processing and analysis and other novel techniques and methods that enhance these services and their management. To support these needs, Cloud Computing services have been utilized for one decade by responding to large-scale systems' growing data management needs. They provide easy, on-demand, location-independent access and enable highly scalable management using virtualization. Meanwhile, the miniaturisation of electronic devices and improvements in battery lifetimes have led to small computational devices with communication capabilities giving birth to the Internet of Things paradigm.

Blockchain is the backbone technology for many Distributed Ledger and Distributed Computing applications, such as digital cryptocurrencies and digital smart contracts. Solutions integrated with Blockchains excel the provenance of high levels of security and trust and guarantee a fully immutable log of transactional history without the interference or control of a central authority. Blockchain applications have been proposed in a wide variety of environments such as distributed voting, eHealth, Mobile Computing, Internet of Vehicles, etc. We believe that integrating Blockchain technology with smart applications for managing data of mobile devices can further enhance the privacy and security requirements of current complex systems.

In this position paper, we discuss Blockchain-integration possibilities for smart systems to support the efficient, secure, and privacy-aware execution of smart applications. We envision a Blockchain simulation framework capable of analysing Blockchain-integration possibilities with fog/edge and cloud infrastructures at different layers of smart systems. The framework will be able to model and analyse the behavior of Blockchain networks in large-scale smart systems.

In the future, we plan to provide enhancements to COVID-19-related applications based on our vision, and use them to validate our proposal.

Keywords

Blockchain, Smart Systems, Cloud Computing, Internet of Things

---

1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society (CERCIRAS Cost Action CA19135), September 2, 2021, Novi Sad, Serbia

baniatah@inf.u-szeged.hu (H. Baniata); dragi.kimovski@aau.at (D. Kimovski); radu.prodan@aau.at (R. Prodan); keratt@inf.u-szeged.hu (A. Kertesz)

The TeamPlay Coordination Language for Energy-, Time- and Security-aware Engineering of Fault-tolerant Cyber-physical Systems

Clemens Grelck
c.grelck@uva.nl

Lukas Miedema
l.miedema@uva.nl

Benjamin Rouxel
b.rouxel@uva.nl

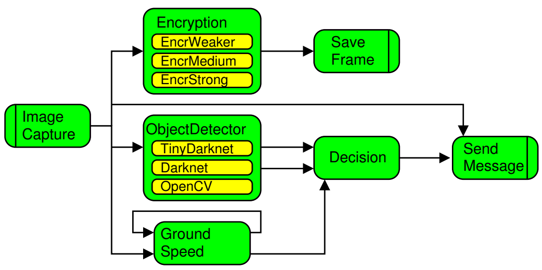
Julius Roeder
j.roeder@uva.nl

I. Introduction

Cyber-physical systems interconnect sensors and actuators with ever more complex computing functions running on increasingly parallel and heterogeneous hardware. Characteristic for cyber-physical systems is that non-functional properties such as time, energy, security but also resilience against hardware failure or cyber attack are as important for the orderly behaviour of a system as traditional functional correctness. Intertwining functional and non-functional properties, the latter often specific to some concrete execution machinery, creates a specific challenge for sustainable software engineering, or in other words is prone to create a software mess.

II. Coordination model and language

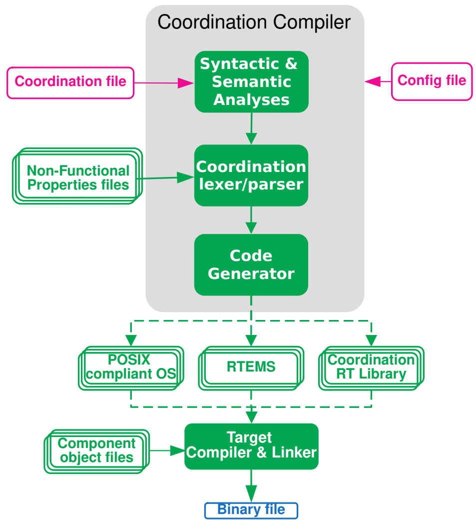
We propose the coordination language TeamPlay to introduce non-functional properties as first-class citizens into the software engineering process. The term *coordination* goes back to the seminal work of Gelernter and Carriero [2]. The TeamPlay language follows an *exogenous* [1] approach with full separation of concerns between intrinsic component behaviour and extrinsic component interaction. We initially proposed TeamPlay in [3]; this work introduces a revised syntax, the latest extensions and sketches out the implementation tool chain. We illustrate TeamPlay by the example of a maritime surveillance drone:



At the core of our approach is the organisation of an application as a set of software components that interact with each other exclusively via typed channels. Components are implemented using a language suitable for the purpose (typically C). Components are activated by the presence of data tokens on their input channels and produce new tokens on their output channels. Components may have multiple versions (e.g. Encryption and ObjectDetector above) that target different hardware units or solve the same problem with different time, energy, security trade-offs.

III. Tool chain

We have designed and built a complete implementation tool chain as illustrated below. At the core is our coordination compiler that takes a TeamPlay coordination programme and per-component as well as per hardware unit specific information from the *non-functional properties file*.



Our coordination compiler generates code for a variety of runtime and operating systems and links this with the external and independently compiled component implementations to an executable binary.

Acknowledgments

This work is supported and partly funded by the European Union Horizon-2020 research and innovation programme under grant agreement No. 779882 (TeamPlay project) and No. 871259 (ADMORPH project).

References

[1] Arbab, F.: Composition of interacting computations. In: Interactive Computation, pp. 277–321. Springer (2006)
[2] Gelernter, D., Carriero, N.: Coordination languages and their significance. Communications of the ACM **35**(2), 97–107 (1992)
[3] Roeder, J., Rouxel, B., Altmeyer, S., Grelck, C.: Towards energy-, time- and security-aware multi-core coordination. In: Coordination Models and Languages, 22nd International Conference. Springer, LNCS 12134 (2020)

Strategy Switching: Smart Fault-tolerance for Resource-constrained Real-time Applications

Lukas Miedema
l.miedema@uva.nl

Benjamin Rouxel
b.rouxel@uva.nl

Clemens Grelck
c.grelck@uva.nl

I. Introduction

As transistor density increases and gate voltage decreases, the rate of *transient faults* also known as *single event upsets* (SEUs) goes up. Software-based fault-tolerance against single event upsets is attractive as it is able to protect workloads on commercial-off-the-shelf (COTS) hardware. However, such techniques incur a high price in terms of additional compute resources. For example, instrumenting a binary using SWIFT-R [1] increases its execution time by 99%.

Resource-constrained real-time systems may not have sufficient processing resources to allow all tasks to run under fault-tolerance regime at all times. This raises the research question as how to best make use of available resources for best possible fault tolerance. Our approach is based on the assumption that certain tasks can withstand one incorrect result or deadline miss, but not multiple consecutive ones [2].

We propose a new approach for such resource-constrained systems. Our approach minimizes the effective unmitigated fault-rate by selecting which tasks are to be run under fault-tolerance. Our approach recognizes that the effective criticality of a task may change over time due to earlier faults or lack thereof, and as such runs different tasks under fault-tolerance at different times.

II. Task and fault model

Task model: We assume a periodic set of tasks τ with a single period P and deadline D such that $D \leq P$ (no pipe-lining). Furthermore, we support precedence relationships between the tasks in our task model. For each task it is known whether it produces output or not, and whether it can tolerate one deadline miss or not. Finally, we assume the system is sufficiently resource-constrained that only a strict subset of τ can be run with fault-tolerance measures. This task model is in line with the TeamPlay coordination language for cyber-physical systems [3], which forms the overarching context for this work.

Fault model: The inter-arrival time between SEUs is exponentially distributed. We do not assume fault detection: only when the task runs under a fault-tolerance scheme, can a fault be detected and mitigated. When a task does not run with fault-tolerance, it is not known whether it succeeded or not. Finally, we assume that when some task fails (i.e. produces incorrect data), all successor tasks fail as well due to operating on incorrect data.

Fault detection and mitigation: We assume the presence and implementation of some fault-tolerance scheme implementing both fault detection and fault mitigation (e.g. spatial/temporal triple modular redundancy, and that any task can be run under that scheme. Our model allows the fault mitigation itself to fail (e.g. due to successive SEUs during both replicas of a task under triple modular redundancy), but assumes that it is known when fault mitigation fails.

III. Strategy Switching

We foresee both an online and an offline strategy switching state machine:

Online: The online strategy switching component selects a strategy s ahead of every (periodic) execution of the task set. The strategy s dictates which tasks run under fault-tolerance scheme (τ_{ft}) and which not ($\tau - \tau_{ft}$). After executing all tasks, the online component uses information from the execution of the task set to select the matching result r. This result reflects the state of the tasks, including if they succeeded, failed, or if it is unknown. This result is then used to select the best next strategy for the next iteration of the task set.

Offline: The full set of strategies $s \in S$ is computed ahead of time, as well as the transition relation Δ from any given result r to the next best successor strategy $\Delta(r) = s$.

We currently develop a variety of algorithms to compute S and Δ, such that our choice of Δ provides the lowest steady-state rate of unmitigated faults.

Acknowledgments

This work is supported and partly funded by the European Union Horizon-2020 research and innovation programme under grant agreement No. 871259 (ADMORPH project).

References

[1] J. Chang, G. A. Reis, and D. I. August, "Automatic instruction-level software-only recovery," in *International Conference on Dependable Systems and Networks (DSN'06)*, IEEE, 2006, pp. 83–92.

[2] N. Vreman, A. Cervin, and M. Maggio, "Stability and performance analysis of control systems subject to bursts of deadline misses," in *33rd Euromicro Conference on Real-Time Systems (ECRTS 2021)*, 2021.

[3] J. Roeder, B. Rouxel, S. Altmeyer, and C. Grelck, "Towards energy-, time- and security-aware multi-core coordination," in *Coordination Models and Languages*, Springer, LNCS 12134, 2020.

How Green is WebAssembly?

João Macedo[1,2], Rui Pereira[2], Rui Abreu[3] and João Saraiva[1,2]

[1]Department of Informatics, University of Minho, Braga, Portugal
[2]HASLab/INESC TEC, Braga, Portugal
[3]FEUP, University of Porto, Porto, Portugal

Abstract

The world wide web has dramatically evolved in recent years. Webpages are dynamic, expressed by programs written in regular programming languages. Thus, web browsers are almost operating systems, having to interpret/compile such programs and execute them within the browser itself.

Although JavaScript is widely used to express dynamic web pages, it has several short comings and performance inefficiencies. To overcome such limitations, a new portable and size/load efficient language is being developed by major IT powerhouses: WebAssembly. In this paper we analyse the current status and present a preliminary study on the energy efficiency of WebAssembly. Our first results show that WebAssembly, while still in its infancy, already challenges JavaScript.

Keywords

WebAssembly, Software Energy Efficiency, Green Computing

1. Introduction

The advent of the mobile smartphone and its widespread usage, change our lifestyle: *everyone* uses a computer and/or smartphone to perform everyday tasks, such as reading news, chatting with friends and playing games. Most of these tasks are performed via web browsers: the most widely used software tool to access internet. While in the very beginning, browsers were used to navigate via static web (HTML) documents, soon dynamic features were included to make we pages more expressive. Already in the 90's, Netscape and Sun reported JavaScript (JS) as an "easy-to-use object scripting language designed for creating live online applications that link together objects and resources on both clients and servers". Thus, JS became the *de facto* client-side web scripting language.

Although JavaScript technology has improved by using advanced Virtual Machines (VM) offering both Just-In-Time (JIT) compilation and GPU support, JavaScript is also known to have poor performance [1, 2]. In a recent survey on the performance of 27 programming languages implementing the same 10 software problems (non Web specific) we showed that JavaScript is in position 15 in the reported ranking and it is 6.5 times slower and 4.45 times more energy greedy than the C language (the fastest and greenest in that ranking) [3, 4, 5]. Because JavaScript source code is on the web (downloaded together with the static part of the webpage), its security and efficiency is of major concern. Internet attacks are often performed by injecting malicious code into the JS component of the webpage being downloaded/executed [6].

To overcome the limitations of JS, software developers from the four major browsers designed WebAssembly (Wasm) as a fast, safe, portable low-level bytecode tailored for Web-based applications [2]. Like all bytecode formats, Wasm aims at be the compilation target for the Web, which like the code produced by a C compiler is highly optimized at code generation time. By being a binary format, Wasm is transmitted over the network much faster than JS and thus reducing load times. Moreover, Wasm is designed such that function bodies code is placed after all declarations, thus allowing browsers to minimize pageload latency by starting streaming compilation as soon as functions arrive over the wire.

As a consequence of this modern design architecture, WebAssembly already shows that when a C program is compiled to Wasm instead of JavaScript, it runs 34% faster on Google Chrome [2]. Although speed is a key aspect of an application, the widely use of powerful mobile devices is making the energy consumption another relevant aspect of software [7]. Unfortunately, there is no work on analysing the energy efficiency of browsers running Wasm applications. In this paper, we analyse the performance of Wasm, both in terms of its execution time and energy efficiency. We compare the performance of 10 Wasm applications to their JavaScript counterparts. Although WebAssembly is still in a very early phase, our preliminary results show that Wasm is faster and more energy-efficient most of the time. In fact, Wasm is consistently more energetically efficient and faster than JS when programs are executed with small inputs. Our results also show that with larger inputs, Wasm programs are still faster and greener but the gap is smaller.

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY
rui.pereira@di.uminho.pt (R. Pereira); rma@fe.up.pt (R. Abreu); saraiva@di.uminho.pt (J. Saraiva)
http://di.uminho.pt/~jas (J. Saraiva)
https://orcid.org/0000-0002-5686-7151 (J. Saraiva)

References

[1] M. Selakovic, M. Pradel, Performance issues and optimizations in javascript: An empirical study, in: Proceedings of the 38th International Conference on Software Engineering, ICSE '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 61–72.

[2] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, J. Bastien, Bringing the Web up to Speed with WebAssembly, in: Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Association for Computing Machinery, New York, NY, USA, 2017, p. 185–200.

[3] M. Couto, R. Pereira, F. Ribeiro, R. Rua, J. Saraiva, Towards a green ranking for programming languages, in: Proceedings of the 21st Brazilian Symposium on Programming Languages, SBLP 2017, Association for Computing Machinery, New York, NY, USA, 2017. (best paper award).

[4] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, J. Saraiva, Energy efficiency across programming languages: How do energy, time, and memory relate?, in: SLE 2017 - Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, co-located with SPLASH 2017, Association for Computing Machinery, Inc, New York, New York, USA, 2017, pp. 256–267.

[5] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, J. Saraiva, Ranking programming languages by energy efficiency, Science of Computer Programming 205 (2021) 102609.

[6] C. Yue, H. Wang, Characterizing insecure javascript practices on the web, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 961–970.

[7] G. Pinto, F. Castor, Energy efficiency: a new concern for application software developers, Communications of the ACM 60 (2017) 68–75.

Identification of selected resource-aware problems across scientific disciplines and applications

Pawel Czarnul

Gdansk University of Technology, 11/12 Narutowicza St., 80-233, Gdansk, Poland

Keywords
resource-aware problems, identification of resources, cross discipline problem analysis

The goal of the work is to perform preliminary identification of resource-aware problems across various disciplines considered in scientific literature by formulations such as integer linear programming (ILP), greedy algorithms, dynamic programming and genetic algorithms, outlining the following:

1. resources,
2. scientific disciplines (associated with profiles of journals the works appear in),
3. practical applications.

As an example, in computer science, resources typically include: execution time (performance), energy, memory/storage, ease of programming/development time. Problem formulations in these cases are typically associated with trade-offs, for example: performance vs energy [1, 2], performance vs security of a system [3], performance vs storage [4], performance/time vs memory [5, 6], performance vs ease of programming/development effort [7].

Various applications considered in this analysis include, among others: allocating resources for fighting forest fires [8], emission minimization, fossil resource usage minimization, employment maximization [9], allocation of health care resources [10], reconfiguration and resource optimization in power distribution networks [11], site selection of a wind power plant [12], operation of a hospital emergency department, studying the impact staffing policies have on such key quality measures as patient length of stay (LoS), number of handoffs, staff utilization levels, and cost [13], decision-CPM network in order to obtain an overall optimum including time, cost, quality and safety in a road building project [14], resource allocation in communication [15, 16], clouds [17, 18], high performance computing systems [19, 1], management of natural resources [20], education [21] etc.

Outcome of this analysis allows to further outline problem formulations from the identified works and link analogous synthetic formulations and approaches used to solve the latter from the algorithmic point of view. This potentially allows to reuse approaches to take up problems already used in other disciplines and correspondingly identify base algorithms that form algorithmic foundations for resource-aware computing.

1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society
pczarnul@eti.pg.edu.pl (P. Czarnul)
http://pg.edu.pl/pawel.czarnul (P. Czarnul)
0000-0002-4918-9196 (P. Czarnul)

CEUR Workshop Proceedings (CEUR-WS.org)

References

[1] A. M. Coutinho Demetrios, D. De Sensi, A. F. Lorenzon, K. Georgiou, J. Nunez-Yanez, K. Eder, S. Xavier-de Souza, Performance and energy trade-offs for parallel applications on heterogeneous multi-processing systems, Energies 13 (2020). URL: https://www.mdpi.com/1996-1073/13/9/2409. doi:10.3390/en13092409.

[2] S. Diouani, H. Medromi, Trade-off between performance and energy management in autonomic and green data centers, in: Proceedings of the 2nd International Conference on Networking, Information Systems & Security, NISS19, Association for Computing Machinery, New York, NY, USA, 2019. URL: https://doi.org/10.1145/3320326.3320332. doi:10.1145/3320326.3320332.

[3] S. Müller, Security trade-offs in Cloud storage systems, Doctoral thesis, Technische Universität Berlin, Berlin, 2017. URL: http://dx.doi.org/10.14279/depositonce-6179. doi:10.14279/depositonce-6179.

[4] H. Jo, Y. Kim, H. Lee, Y. Lee, H. Han, S. Kang, On the trade-off between performance and storage efficiency of replication-based object storage, in: J. Chen, L. Yang (Eds.), Proceedings - 11th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2019, 19th IEEE International Conference on Computer and Information Technology, CIT 2019, 2019 International Workshop on Resource Brokering with Blockchain, RBchain 2019 and 2019 Asia-Pacific Services Computing Conference, APSCC 2019, International Conference on Cloud Computing Technology and Science, Institute of Electrical and Electronics Engineers (IEEE), United States, 2019, pp. 301–304. doi:10.1109/CloudCom.2019.00051, 11th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2019, 19th IEEE International Conference on Computer and Information Technology, CIT 2019, 2019 International Workshop on Resource Brokering with Blockchain, RBchain 2019 and 2019 Asia-Pacific Services Computing Conference, APSCC 2019 ; Conference date: 11-12-2019 Through 13-12-2019.

[5] J. M. Bermudo Mera, A. Karmakar, I. Verbauwhede, Time-memory trade-off in toom-cook multiplication: an application to module-lattice based cryptography, IACR Transactions on Cryptographic Hardware and Embedded Systems 2020 (2020) 222–244. URL: https://tches.iacr.org/index.php/TCHES/article/view/8550. doi:10.13154/tches.v2020.i2.222-244.

[6] G. Avoine, P. Junod, P. Oechslin, Characterization and improvement of time-memory trade-off based on perfect tables, ACM Trans. Inf. Syst. Secur. 11 (2008). URL: https://doi.org/10.1145/1380564.1380565. doi:10.1145/1380564.1380565.

[7] K. Karimi, The feasibility of using opencl instead of openmp for parallel cpu programming, 2015. arXiv:1503.06532.

[8] J. Rodríguez-Veiga, M. J. Ginzo-Villamayor, B. Casas-Méndez, An integer linear programming model to select and temporally allocate resources for fighting forest fires, Forests 9 (2018). URL: https://www.mdpi.com/1999-4907/9/10/583. doi:10.3390/f9100583.

[9] E. Özcan, S. Erol, A multi-objective mixed integer linear programming model for energy resource allocation problem: The case of turkey, Gazi University Journal of Science 27 (2014) 1157 – 1168.

[10] D. Epstein, Z. Chalabi, K. Claxton, M. Sculpher, Mathematical programming for the optimal

allocation of health care resources, 2005.
[11] A. Borghetti, Mixed Integer Linear Programming Models for Network Reconfiguration and Resource Optimization in Power Distribution Networks, John Wiley & Sons, Ltd, ????, pp. 43–88. doi:https://doi.org/10.1002/9781119116080.ch2.
[12] E. S. Ari, C. Gencer, Proposal of a novel mixed integer linear programming model for site selection of a wind power plant based on power maximization with use of mixed type wind turbines, Energy & Environment 31 (2020) 825–841. URL: https://doi.org/10.1177/0958305X19882394. doi:10.1177/0958305X19882394. arXiv:https://doi.org/10.1177/0958305X19882394.
[13] S. Y. Shin, Y. Brun, H. Balasubramanian, P. L. Henneman, L. J. Osterweil, Discrete-event simulation and integer linear programming for constraint-aware resource scheduling, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (2018) 1578–1593. doi:10.1109/TSMC.2017.2681623.
[14] J. R. San Cristóbal Mateo, An integer linear programming model including time, cost, quality, and safety, IEEE Access 7 (2019) 168307–168315. doi:10.1109/ACCESS.2019.2953185.
[15] S. Najeh, H. Besbes, A. Bouallegue, Greedy algorithm for dynamic resource allocation in downlink of ofdma system, in: 2005 2nd International Symposium on Wireless Communication Systems, 2005, pp. 475–479. doi:10.1109/ISWCS.2005.1547746.
[16] N. Ferdosian, M. Othman, B. M. Ali, K. Y. Lun, Greedy–knapsack algorithm for optimal downlink resource allocation in lte networks, Wireless Networks 22 (2015) 1427–1440. URL: http://dx.doi.org/10.1007/s11276-015-1042-9. doi:10.1007/s11276-015-1042-9.
[17] M. C. Calzarossa, L. Massari, G. Nebbione, M. L. Della Vedova, D. Tessera, Tuning genetic algorithms for resource provisioning and scheduling in uncertain cloud environments: Challenges and findings, in: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2019, pp. 174–180. doi:10.1109/EMPDP.2019.8671564.
[18] L. Liu, Q. Fan, Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment, IEEE Access 6 (2018) 24533–24542. doi:10.1109/ACCESS.2018.2830639.
[19] T. Boiński, P. Czarnul, Optimization of Data Assignment for Parallel Processing in a Hybrid Heterogeneous Environment Using Integer Linear Programming, The Computer Journal (2021). doi:10.1093/comjnl/bxaa187.
[20] J. O. Kennedy, Dynamic programming applications to agriculture and natural resources (????). URL: https://www.osti.gov/biblio/7151445.
[21] A. A. Popov, O. N. Lopateeva, A. K. Ovsyankin, M. M. Satsuk, Application of greedy algorithms for the formation of the educational schedule in the higher education, Journal of Physics: Conference Series 1691 (2020) 012066. URL: https://doi.org/10.1088/1742-6596/1691/1/012066. doi:10.1088/1742-6596/1691/1/012066.

Open-Source Research on Time-predictable Computer Architecture

Martin Schoeberl[1]

[1]*Department of Applied Mathematics and Computer Science, Technical University of Denmark*

Abstract

Real-time systems need time-predictable computers to be able to guarantee that computation can be performed within a given deadline. Therefore, these systems are resource aware systems, where time is the primary resource considered. This paper presents an open-source project on a time as a resource aware computer architecture.

Open-source software is currently the basis of many Internet services, e.g., an Apache web server running on top of Linux with a web application written in Java. Furthermore, for most programming languages in use today, there are a open-source compilers available.

However, hardware designs are seldom published in open-source. Furthermore, many artifacts developed in research, especially hardware designs, are not published in open source. The two main arguments formulated against publishing research in open source are: (1) "When I publish my source before the paper gets accepted, someone may steal my ideas" and (2) "My code is not pretty enough to publish it, I first will clean it up (which seldom happens)".

In this paper and in the presentation I will give counterarguments for those two issues. Furthermore, I will present the successful T-CREST research project, where almost all artifacts have been developed in open source from the day one on.

Keywords

real-time systems, time-predictable computer architecture, open-source

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

masca@imm.dtu.dk (M. Schoeberl)
http://www.imm.dtu.dk/~masca/ (M. Schoeberl)
0000-0003-2366-382X (M. Schoeberl)

Resource Awareness in Complex Industrial Systems – A Strategy for Software Updates

Petar Rajković [1], Dejan Aleksić [2], Dragan Janković [1]

[1] *University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, Niš, Serbia*
[2] *University of Niš, Faculty of Science and Mathematics, Department of Physics, Višegradska 33, Niš Serbia*

The complex industrial systems consist of many heterogeneous devices running different pieces of software in a layer-organized environment.
Starting from the sensors/actuator layers, through the edge layer, via SCADA and manufacturing execution systems (MES) to enterprise resource planning (ERP), all pieces of equipment run the software that needs to be updated from time to time. Furthermore, all these pieces of software communicate with each other using different protocols which makes the software update process a bit more complex.
The main objective of this paper is to highlight the common problems with software updates across multiple layers and to bring the set of recommendations and guidelines for the most effective and the cheapest software updates – from the resource awareness point of view.
The most critical points for resource management are the storage capacity (especially on the lower levels) and the data traffic through the connecting networks. I.e., MES systems could run in a shop floor environment on computers connected to the wireless network which could experience different disruptions as the result of the operating nearby machines generating high frequencies. When deploying a new version of the software to some device, an update package needs to be distributed via a network, stored to the destination device, and the old version needs to be backup in case of roll-back.
Depending on the type of devices, their number, and the layer of running software, the amount of transferred data (per device) could be anything between 1kB and 1GB. Considering regular data traffic, these additional bytes could create significant additional network load causing communication problems. Also, the devices, where the software is run, could have a limited storage capacity which would require backup on a remote location.
Furthermore, in the case of detected update errors, the deployment system needs to provide the possibility to roll back to the previous version of the software which requires additional time and network resources.
To reduce the impact of the mentioned problems on the system uptime, we tried to define the general approach that could be configured to use the combination of blue-green and canary deployment approaches in combination with both shared and local backups. In that way, we tried to establish the most effective deployment environment, applicable on each software layer.
In this paper we will present the following scenarios and the effects of different deployment configurations:

- Software update on one layer
- Communication protocol update between layers (leading to software update on both sides)
- Complex update including protocol and software changes in random layers
- Backup fails strategy
- Deployment fail/ Rollback strategy
- Rollbacks fail strategy

The results we achieve vary between different software layers and scenarios but applying proposed strategies we managed to reduce the data traffic in some cases roughly up to 30% and the downtime by 65%. Also, for some scenarios, we even increase the data traffic to reduce downtime. In the paper, detailed results will be presented.
As the next steps in the development, we will briefly present the digital twin of the industrial system that could be used to simulate different deployment and recovery strategies.

A Prototyping and Evaluation Framework for Research on Timing-analysable Memory Hierarchies for Embedded Multicore SoCs

Florian Haas, Sebastian Altmeyer

University of Augsburg, Germany

The performance of multicore processors is strongly desired in various domains of embedded systems to satisfy the increasing demand for computational power. Complex algorithms and software systems, e. g. in autonomous driving, would benefit from high-performance general-purpose shared-memory multicores. However, these processors do not meet the typical requirements on real-time and safety, and thus cannot be used without performance-degrading and laborious software mechanisms. Elaborate methods in such systems have been developed to further improve the average-case performance of the processor, for example the increasing depth of the memory hierarchy. These and the shared resources, like last-level caches, buses, and main memory, result in the ultimate challenge of calculating tight WCET bounds for the tasks in a time-critical system.

The general objective of research on this topic is to facilitate predictable performance, with minimal over-estimation of timing bounds, by reducing the sources of potential interferences on shared resources. Existing software-based approaches, e. g. performance counter monitors, or program modification during compilation, are limited, as they can either only detect excessing interferences, or are required to be applied to all tasks of the system. Thus, hardware mechanisms promise a better lever to control the behaviour of any task on the system, although they demand for a proper evaluation platform. To research potential improvements on shared resource accesses under timing constraints, a realistic model of a typical memory hierarchy is needed in the first place. Microarchitecture simulators with multicore configurations exist, but their processor-centric design does not support for a prototype implementation and a realistic evaluation. Further, the evaluation system needs to be capable of executing realistic benchmarks, for prototyping different ideas, as well as for a thorough evaluation of their impact on the performance.

This paper describes the requirements of a research-centric multicore SoC for embedded systems, and outlines the assembly of the individual parts into a synthesizable design for both simulation and prototyping on an FPGA. The framework is based on ChipYard, which supports design and evaluation of full-system hardware, using the Rocket Chip generator and its in-order RISC-V CPUs. The main benefit of ChipYard is the configurability and customizability of the involved modules. The interconnects could also be replaced with a NoC to research on manycore systems, or a combination of both with shared-memory clusters connected through a NoC. Based on the proposed framework, the research on elements of the memory hierarchy will be facilitated to improve the applicability of multicore processors in embedded systems.

haas@es-augsburg.de (F. Haas); altmeyer@es-augsburg.de (S. Altmeyer)

Resource-aware approach in the design of complex information systems as a problem of multicriteria optimization

Mikhail Tatur [1] , Natalia Novoselova [2] and Viachaslav Prorovski [1]

[1] *Belarusian State University of Informatics and Radioelectronics, Pietrusia Brouki, 6, Minsk 220013 Belarus*
[2] *United Institute of Informatics Problems, Surganova 6, Minsk, 220012, Belarus*

Abstract

One of the present tasks of the modern information society is the design of complex multi-core information systems, taking into account the rational use of system resources, namely, the development of a new resource-aware approach for the design of next generation information systems, including smart home systems, autonomous transport, cyber-physical systems, etc.

This paper considers the application of data mining technology to solving the problem of optimizing technical solutions taking into account the resource-aware approach, based on the method of multicriteria optimization using a genetic algorithm (GA). Evolutionary optimization techniques allow to cope with the multidimensionality, nonlinearity, and stochasticity of real data sets.

All types of used system resources are parameters of a technical solutions and can be considered as points in a multidimensional feature space, which allows finding optimal solutions according to some complex criterion.

As the formation of such a complex criterion is a rather difficult task, due to the specifics of technical solutions, it is of interest to use many criteria (parameters), namely, finding the Pareto front of optimal solutions - non-dominated solutions. In this case, the main criteria to be optimized may be the execution time of certain tasks, the energy consumed, the multiprocessor architecture used, etc.

As a rule, the criteria are interdependent, i.e. an increase in one of them can lead to a decrease in the value for the other; therefore, the use of multicriteria optimization makes it possible to search for a trade-off between them or to find the non-dominant solutions, where none of them is better than the other in all the considered parameters and therefore have the same importance. In this case, many solutions are allowed, each of which is acceptable in the absence of preliminary information about the importance of the criteria.

The main advantage of using evolutionary methods in multicriteria optimization is the ability to obtain many alternative solutions, as well as intermediate solutions that can be analyzed and the final decision can be made. Having a set of several non-dominated solutions obtained as a result of multicriteria optimization of the technical solutions taking into account the resource-aware approach, it is possible to choose a solution that is most preferable for a specific applied problem.

Keywords

resource-aware approach, multi-core information systems, data mining, multicriteria optimization

Proceedings Name, Month XX–XX, YYYY, City, Country
EMAIL: tatur@bsuir.by (A. 1); novos65@mail.com (A. 2); slawapro@gmail.com (A. 3)
ORCID: XXXX-XXXX-XXXX-XXXX (A. 1); 0000-0002-3825-8889 (A. 2); 0000-0002-6819-1611 (A. 3)

Safety Risk Analysis in Multi-Agent Scenario

Michael Kieviet[1], Padma Iyenghar[1,2]

[1]innotec GmbH, Erlenweg 12, 49324, Melle, Germany
[2]Software Engineering Research Group, University Of Osnabrueck, Germany

Keywords
Safety parameter, Multi-agent Systems, Risk analysis

The upcoming Fifth Generation (5G) and beyond systems are expected to achieve several performance objectives towards supporting various emerging applications/platforms, including the Internet of Things (IoT) [1]. Thus, these interconnected *things (e.g. machineries)* tend to become more and more mobile and flexible. This kind of flexibility offers the possibility to collaborate among the machines. In such a scenario, a set of individual machines partly work together to solve a job and after this, they diverge or rebuild in another constellation. This kind of flexibility has an impact on the process of (safety) risk analysis [2] and their resulting definition of safety functions. The meaning of *risk* in this context is the probability of a *harm* getting too dangerous in combination with the severity of harm.

Several methods are available to have a systematic approach for estimating the risk, such as Risk Priority Numbers (RPN), Risk graph and As Low As Reasonably Practicable (ALARP). Depending on the risk level, it is necessary to define the mitigation measures as (a) inherent safe construction (b) risk reduction by safeguarding or implementation of complementary protective measures and (c) risk mitigation measures by safety management or behaviour rules.

The idea behind this work is to introduce a method based on the multi-agent system theory by deriving the requirements from complex system to a simple safety function and modelling the behaviour of the number of single functional safety function back into the complex system under the circumstances by super positioning of functional safety functions. This concept will help to identify safety requirements for individual safety agents and supports the system safety engineering concept for dynamic complex systems. This paper introduces the safety risk analysis method mentioned above with an application example for a negotiation process of safety functions in unmanned transport vehicles (i.e., agents).

References

[1] S. K. Sharma, I. Woungang, A. Anpalagan, S. Chatzinotas, Toward Tactile Internet in Beyond 5G Era: Recent Advances, Current Issues, and Future Directions, IEEE Access 8 (2020).

[2] IEC 61508-1:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010. URL: https://webstore.iec.ch/publication/5515.

1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society (CERCIRAS)
michael.kieviet@safelabs.de (M. Kieviet); piyengha@uos.de (P. Iyenghar)

Considerations on combining Vestal's mixed-criticality task model and the predictable execution model (PREM) for real-time systems

Ishfaq Hussain[1], Muhammad Ali Awan[1], Konstantinos Bletsas[1], Pedro F. Souto[2] and Eduardo Tovar[1]

[1] *CISTER Research Centre and ISEP/IPP, Porto, Portugal*
[2] *University of Porto, FEUP-Faculty of Engineering and CISTER Research Centre, Porto, Portugal*

Abstract
In the design of critical real-time embedded systems, predictability in timing behavior and in system resource usage is necessary. Vestal's mixed-criticality task model and the Predictable Execution Model (PREM) help achieve these objectives in different ways. Under Vestal's model, multiple worst-case execution time (WCET) estimates are considered for each task, with corresponding degree of confidence, and associated with a different criticality level. The schedulability analysis can derive the appropriate timing safety guarantees for each task without using more conservative estimates than needed, thereby avoiding overengineering. The adaptive variant of Vestal's model also allows for system modes, with some tasks idled at mode change and more conservative WCET estimates thereafter assumed for remaining tasks. Meanwhile, the 2-state PREM model, via compiler support, first fetches from memory (into the cache) all the locations that a task will access, and only subsequently proceeds with computation. This removes a lot of the uncertainty in WCET estimation stemming from the cache state and memory access delays, leading to better predictability and tighter WCET estimates. Vestal's model and the PREM model, however, were independently conceived, and never combined. In this work, we explore different possibilities about how these two models could be combined. We focus on the semantics of multiple (static or probabilistic) per-task estimates of processor computation time and number of memory accesses, how these can be derived, the associated compiler and O/S support required, and the implications for timing analysis.

Keywords
Real time systems, mixed criticality model, Predictable execution model,

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY
hussa@isep.ipp.pt (I. Hussain); awa@isep.ipp.pt (M. A. Awan); Koble@isep.ipp.pt (K. Bletsas); pfs@fe.up.pt (P. F. Souto); emt@isep.ipp.pt (E. Tovar)
http://www.cister.isep.ipp.pt/people/ishfaq__hussain// (I. Hussain)
0000-0002-4470-1744 (I. Hussain)

An approach to Estimating Energy Consumption of web-based IT systems

Maja H. Kirkeby[1], John Gallagher[1] and Bent Thomsen[2]

[1]Roskilde University, Roskilde, Denmark
[2]Aalborg University, Aalborg, Denmark

In this short paper we motivate and present an approach to estimating the entire energy consumption of a web-based software system. The purpose of the estimation is to be able to document – and in the future also predict – the impact of changes to the platform such as energy-reducing initiatives or a growth in the number of users. In the future, we hope that such energy estimates will play an important role in energy certified software development.

There has been previous research on energy consumption of software systems by analysing and optimising the source code of application software. For distributed, communicating systems such as web-based systems, it is crucial also to consider the relationship between software running on different devices in order to understand the energy consumption of the whole system. Studies on the energy consumption of web services have found that there are three major energy consumers: user-devices (42%), data transmission via networking (27%) and servers perhaps in data centres (31%). This indicates that we have to take all three major energy consumers, and their interdependencies, into consideration when evaluating the impact of energy reducing initiatives.

Our research will be driven by a major case study on Edora A/S's Work Force Planner (WFP), which has a fairly typical web service architecture. WFP is a live, self-service web-based booking and resource management system with 500,000 daily interactions used by many local authorities in Denmark such as job centres, unemployment insurance funds and public authorities. We aim to build an energy model for WFP incorporating parameters influencing energy consumption, including both top-down features such as measured electricity usage, user behaviour and network traffic as well as bottom-up features obtained by analysing the source code running on servers and client devices. Potential modelling techniques include semantics-based code analysis and abstraction, and models such as priced timed automata supported by the advanced formal modelling tool UPPAAL.

Cerciras Workshop, September 2–3, 2021, Novi Sad, Serbia