Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Exposé

Context-Adaptive Computational Offloading for Distributed Nature-Inspired Algorithms in Heterogeneous Edge Environments

---

Erstgutachter: Professor J. Edinger
Zweitgutachter: TBD

Abgabe: XX.2022

Table of Contents

List of Figures

List of Tables

List of Abbreviations

GA . . . . Genetic Algorithm

pGA . . . . parallel Genetic Algorithm

QoE . . . . Quality of Experience

TSP . . . . Traveling Salesman Problem

TVM . . . . Tasklet Virtual Machine

1 Introduction

In the last decades computing landscape was transformed tremendously by advancements in computing technology. Nowadays the usage of desktops and mainframes has been extended by many types of heterogeneous devices like smart phones, embedded systems, smart sensors, robotics, wearables etc. Powered by Moores Law these devices are able to do a lot of processing even though they are limited by design through battery life, network bandwidth, storage space and computing power [KLLB13]. However, computation power demands of software especially real-time applications like video processing of surveillance cameras or object recognition of robots can surpass the device limits regarding processing power or fulfilling certain time constraints [KLLB13]. To overcome device limitations, the concept of computational offloading was introduced, which migrates part of its computation [MHF13] onto the sheer infinite amount of resources provided by cloud computing [AFG+09]. In order to enhance the Quality of Experience (QoE) regarding latency-sensitive applications, computational offloading has moved to edge computing, bringing resources even closer to the end-user [LLJL19].

Edge-centric Computing however comes with its own set of challenges since the edge usually consists of heterogeneous human-controlled devices such as desktops, tablets, smart phones, etc. [GLME+15] Those devices mostly remain idle even though they could contribute as computing resources in a distributed environment [SEV+16]. Schäfer et al. [SEV+16] therefore proposed a distributed computing framework, namely the Tasklet system, to make use of computation resources in heterogeneous environments in a generic way. As a result hardware independent execution of computational offloading is achieved.

Beyond mentioned computational challenges in which computational offloading can be utilized there is a class of problems in complexity theory called NP-problems. An instance of this complexity class is the Traveling Salesman Problem (TSP) in which a Salesman wants to visit a certain number of cities while taking the shortest path and visiting every city only once [Ara16]. Since the TSP is of nondeterministic combinatorial nature, exact solutions in a reasonable time are not feasible. Therefore metaheuristics like the Genetic Algorithm (GA) can be used to find a correct solution in a timely manner which doesn't have to be the best solution though [Ara16]. A GA can also be parallelized by different approaches and hardware settings [HA20a]. In the context of computational offloading and heterogeneous devices new challenges for designing a parallel Genetic Algorithm (pGA) arise due to the nature of heterogeneity [SA15].

1.1 Objectives

This thesis objectives are laid out in several phases. First of all there is the design and implementation of a system architecture utilizing a pGA in order to solve the TSP. Furthermore the system should distribute its workload using the Tasklet System [SEP+16] in a heterogeneous edge environment. Secondly there needs to be a function derived from the Tasklet System to allow context-adaptive execution of workloads in a suitable manner during the experimentation phase. Lastly the obtained results need to be verified by analyzing the solution quality and execution time against a naive approach to distributing a pGA in a heterogeneous edge environment.

1.2 Structure

The thesis structure is divided into several chapters. After the Introduction in chapter one follows chapter 2 in which the theoretical foundation regarding Distributed Systems, the TSP and Genetic Algorithms is laid out. The third chapter covers related work followed by the fourth chapter in which the algorithmic design of a distributed pGA utilizing the Tasklet System is presented. The fifth chapter centers about the system design which will be implemented in the sixth chapter. Afterwards in the seventh chapter evaluation of the findings will be highlighted. Lastly on the eighth chapter a conclusion and outlook will be given.

2 Related Work

In a historical analysis Alba and Tomassini [AT02] show that the idea of enhancing Genetic Algorithms with parallelization not only to improve performance but to increase diversity and defer stagnation dates back into the 1960s. Nowadays parallelization of Genetic Algorithms regarding distribution models, hardware configurations, parameterization, genetic operators and applications are a thoroughly researched topic [HA20b]. Nevertheless homogenous distributed systems in the sense of hardware as well as software are the preferred choice in recent work on parallel metaheuristics [SA15]. Therefore heterogeneous (edge) environments present themselves as challenges in recent work on parallel genetic algorithms [HA20b].

Chong [CL99] performed as early as 1999 experiments on the communication performance of pGA on heterogeneous platforms over the internet and concluded the superiority of asynchronous over synchronous communication which has later been confirmed by Kang et al. [KLC01]. Not long after Alba et al. [ANT02] analyzed technical and practical issues induced by heterogeneous environments and showed that superlinear performance can also be achieved in these environments.

Bazterra et al. [BCFF05] proposed several metrics to measure the time, speedup and efficiency of parallel algorithms on heterogeneous computer systems. Furthermore they developed an adaptive pGA in a master-slave model which distributes the fitness evaluation of individuals according to the processing power of slave-nodes.

More recently Salto et al. [SA15] performed experiments on adapting distributed evolutionary algorithms to heterogeneous hardware in an asynchronous island model-based setting for solving the TSP. For this purpose they proposed the HAPA methodology to deal with heterogeneous hardware and distribute workloads evenly by performing benchmarks beforehand. As a result two algorithms were derived to regulate the number of generations evaluated per node on the one hand and to manage the migration frequency on the other hand, relative to the fastest benchmark respectively. Anyhow, heterogeneous nodes were able to communicate with each other which is not feasible in this thesis due to the underlying Tasklet System whose isolation model disallows interprocess communication. In addition uncertainty about the nodes participating as resource providers in the Tasklet System inhibits utilizing benchmarks to assign workloads relative to another node.

Valente De Oliveira et al. [VDOBD17] performed various experiments using asynchronous island model-based pGA to solve the TSP over heterogeneous grid enabled hardware. Although the focus of their work emphasized the impact of different migration topologies, their findings concluded that adding even slow nodes improved processing time while maintaining good fitness of the solutions. Nevertheless, this thesis aims not to research on different migration topologies but rather the challenges of heterogeneity in a computational offloading setting which differs from grid computing.

3 Solution Design

In order to meet the objectives of this thesis there needs to be a prototypical implementation to run a pGA on the Tasklet System. Therefore a system design draft with the components needed is displayed in Figure 3.1.

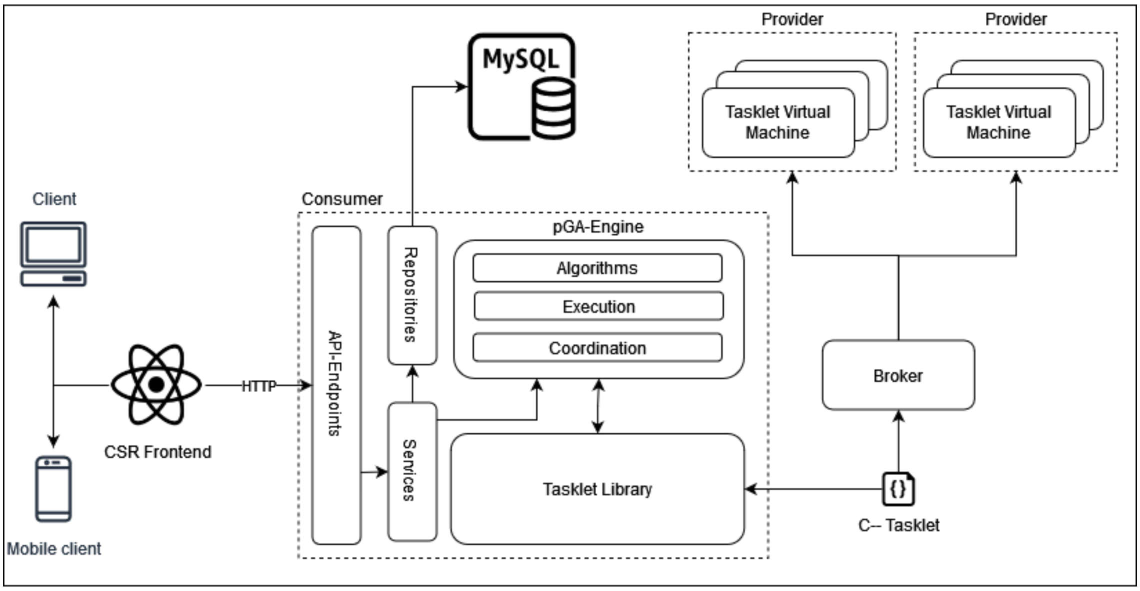

Figure 3.1: System Design

The system design composes of several main components:

- **Frontend:** The frontend is a client-side rendered web-interface which can be accessed by any client through a web browser to configure and start a pGA.
- **Server:** The server serves two purposes, it links all components together and it controls the execution of a pGA.
- **API-Endpoints:** The API-endpoints are used to enable communication between the frontend and server via HTTP-Connection.
- **Services:** The services provide general functionality which is called by the API-Endpoints.
- **Repositories:** The repositories are interfaces to access the database in order to create, read, update and delete entities.
- **Database:** The database holds information about TSP-data, jobs as well as metrics.

- **Tasklet Library:** The Tasklet Library provides access to common functionality of the Tasklet System and is used to generate C−− bytecode which is then via the Tasklet Broker served to a Tasklet Virtual Machine (TVM) of a Provider.
- **pGA-Engine:** The pGA-Engine serves as the core of the server. It instantiates algorithms, executes them and controls the coordination between the the TVM executions.

In the second step an experimentation phase is needed to make use of the context of a TVM in which a pGA is running. Under the presumption of running an island model-based pGA there is a number of n generations which are calculated on a TVM until the execution terminates and the result is sent back to the server for migration. In the context of heterogeneous execution environments the processing time needed to compute n generations of the pGA depends heavily on the processing power of the underlying hardware of a TVM. Therefore the idea is to utilize the benchmark values x of several TVMs to construct a function $f(x)$ in order to calculate the number of generations $n' > 0$ resulting in similar execution times $t(n) \approx t(n')$. The idea behind this approach is on the one hand to reduce total processing time, since slow TVMs can delay the termination of the algorithm and on the other hand improve migration quality from and to slow TVMs since they can contribute and benefit earlier to and of the migration process.

Lastly the findings need to be evaluated. Hence a lot of metrics need to be gathered during the experimentation and visualized for interpretation. Additionally the findings need to be compared to a pGA-version without context-adaptation. Metrics of special interest for the evaluation could be:

- **Speedup:** The speedup is one of the most interesting metrics to be analyzed. At best a substantial increase in performance of the context-adaptive pGA over the default pGA can be shown.
- **Hit rate:** The hit rate accounts for the number of times the pGA has produced the optimal result of a TSP-instance. Comparing these can give an insight into the quality of the algorithm.
- **Total execution time and average fitness:** The total execution time and average fitness are most significant in cases where the algorithm does not find an optimal solution. Therefore identifying shorter processing times without degradation of average fitness values of the solution is also relevant.

Other metrics for evaluation can be considered after more intense research on current literature.

Bibliography

[AFG^{+}09] ARMBRUST, Michael ; FOX, Armando ; GRIFFITH, Rean ; JOSEPH, Anthony D. ; KATZ, Randy H. ; KONWINSKI, Andrew ; LEE, Gunho ; PATTERSON, David A. ; RABKIN, Ariel ; STOICA, Ion ; ZAHARIA, Matei: Above the Clouds: A Berkeley View of Cloud Computing / EECS Department, University of California, Berkeley. Version: Feb 2009. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html. 2009 (UCB/EECS-2009-28). – Forschungsbericht

[ANT02] ALBA, Enrique ; NEBRO, Antonio J. ; TROYA, José M.: Heterogeneous Computing and Parallel Genetic Algorithms. In: *Journal of Parallel and Distributed Computing* 62 (2002), Nr. 9, 1362-1385. http://dx.doi.org/https://doi.org/10.1006/jpdc.2002.1851. – DOI https://doi.org/10.1006/jpdc.2002.1851. – ISSN 0743–7315

[Ara16] ARABI, Bander H.: Solving NP-complete Problems Using Genetic Algorithms. In: *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim)*, 2016, S. 43–48

[AT02] ALBA, E. ; TOMASSINI, M.: Parallelism and evolutionary algorithms. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), Nr. 5, S. 443–462. http://dx.doi.org/10.1109/TEVC.2002.800880. – DOI 10.1109/TEVC.2002.800880

[BCFF05] BAZTERRA, Victor E. ; CUMA, Martin ; FERRARO, Marta B. ; FACELLI, Julio C.: A general framework to understand parallel performance in heterogeneous clusters: analysis of a new adaptive parallel genetic algorithm. In: *Journal of Parallel and Distributed Computing* 65 (2005), Nr. 1, 48-57. http://dx.doi.org/https://doi.org/10.1016/j.jpdc.2004.09.011. – DOI https://doi.org/10.1016/j.jpdc.2004.09.011. – ISSN 0743–7315

[CL99] CHONG, Fuey S. ; LANGDON, W. B.: Java Based Distributed Genetic Programming on the Internet. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 2*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999 (GECCO'99). – ISBN 1558606114, S. 1229

[GLME+15] Garcia Lopez, Pedro ; Montresor, Alberto ; Epema, Dick ; Datta, Anwitaman ; Higashino, Teruo ; Iamnitchi, Adriana ; Barcellos, Marinho ; Felber, Pascal ; Riviere, Etienne: Edge-Centric Computing: Vision and Challenges. In: *SIGCOMM Comput. Commun. Rev.* 45 (2015), sep, Nr. 5, 37–42. `http://dx.doi.org/10.1145/2831347.2831354`. – DOI 10.1145/2831347.2831354. – ISSN 0146–4833

[HA20a] Harada, Tomohiro ; Alba, Enrique: Parallel Genetic Algorithms: A Useful Survey. In: *ACM Computing Surveys* 53 (2020), 08, S. 1–39. `http://dx.doi.org/10.1145/3400031`. – DOI 10.1145/3400031

[HA20b] Harada, Tomohiro ; Alba, Enrique: Parallel Genetic Algorithms: A Useful Survey. In: *ACM Computing Surveys* 53 (2020), 08, S. 1–39. `http://dx.doi.org/10.1145/3400031`. – DOI 10.1145/3400031

[KLC01] Kang, Li-Shan ; Liu, Pu ; Chen, Yu-Ping: Asynchronous parallel evolutionary algorithm for function optimization. In: *Journal of Computer Research and Development* 38 (2001), Nr. 11, S. 1381–1386

[KLLB13] Kumar, Karthik ; Liu, Jibang ; Lu, Yung-Hsiang ; Bhargava, Bharat: A Survey of Computation Offloading for Mobile Systems. In: *Mob. Netw. Appl.* 18 (2013), feb, Nr. 1, 129–140. `http://dx.doi.org/10.1007/s11036-012-0368-0`. – DOI 10.1007/s11036–012–0368–0. – ISSN 1383–469X

[LLJL19] Lin, Li ; Liao, Xiaofei ; Jin, Hai ; Li, Peng: Computation Offloading Toward Edge Computing. In: *Proceedings of the IEEE* 107 (2019), Nr. 8, S. 1584–1607. `http://dx.doi.org/10.1109/JPROC.2019.2922285`. – DOI 10.1109/JPROC.2019.2922285

[MHF13] Mtibaa, Abderrahmen ; Harras, Khaled A. ; Fahim, Afnan: Towards Computational Offloading in Mobile Device Clouds. In: *2013 IEEE 5th International Conference on Cloud Computing Technology and Science* Bd. 1, 2013, S. 331–338

[SA15] *Kapitel* Adapting Distributed Evolutionary Algorithms to Heterogeneous Hardware. In: Salto, Carolina ; Alba, Enrique: *Transactions on Computational Collective Intelligence XIX*. Bd. 9380. 2015. – ISBN 978–3–662–49016–7, S. 103–125

[SEP+16] Schafer, Dominik ; Edinger, Janick ; Paluska, Justin M. ; VanSyckel, Sebastian ; Becker, Christian: Tasklets: "Better than Best-Effort" Computing. In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, 2016, S. 1–11

[SEV+16] SCHÄFER, Dominik ; EDINGER, Janick ; VANSYCKEL, Sebastian ; PALUSKA, Justin M. ; BECKER, Christian: Tasklets: Overcoming Heterogeneity in Distributed Computing Systems. In: *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2016, S. 156–161

[VDOBD17] VALENTE DE OLIVEIRA, José ; BALTAZAR, Sérgio ; DANIEL, Helder: On Asynchronous Parallelization of Order-Based GA over Grid-Enabled Heterogenous Commodity Hardware. In: *Soft Comput.* 21 (2017), nov, Nr. 21, 6351–6368. `http://dx.doi.org/10.1007/s00500-016-2190-2`. – DOI 10.1007/s00500–016–2190–2. – ISSN 1432–7643