Parallel Discrete Event Simulation in Haskell

A functional approach

Jonathan Thaler
Thorsten Altenkirch
jonathan.thaler@nottingham.ac.uk
thorsten.altenkirch@nottingham.ac.uk
University of Nottingham
Nottingham, United Kingdom

ABSTRACT

TODO: submit to a journal

KEYWORDS

Agent-Based Simulation, Functional Reactive Programming, Property-Based Testing, Haskell

ACM Reference Format:
Jonathan Thaler and Thorsten Altenkirch. 2019. Parallel Discrete Event Simulation in Haskell: A functional approach. In *Proceedings of International Symposium on Implementation and Application of Functional Languages (IFL'18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

main message: optimistic PDES needs rollback of events which is particularly easily achieved in Haskell because: 1. it has persistent immutable data-structure resulting in a new version after an update where one still can hold the old version, 2. one can restrict side-effects to one which are guaranteed to be able to rollback and not having an effect on the real-world e.g. IO. Also using Haskell has the potential to come up with a DSL to declaratively implement DES models e.g. express a DES model in terms of sinks, sources, delays,... on a declarative level.

because DES can be the foundation for an event-driven ABS we benefit from looking at PDES as well to get inspiration for parallelisation of ABS

rollback should be easy with SFs as well. with MSFs more delicate because of side-effects

Contribution: first ones to describe haskells immutable data properties and STM to implement PDES

TODO: implement DES in Haskell TODO: implement an optimistic PDES in Haskell

logical processes (e.g. source, sink, delay,...) which are MSFs sending messages with a time-stamp to each other never process messages with time-stamp lower than previous one.

implement a DSL for specifying DES which gets then translated to MSFs for internal execution

rolling back in Haskell: keep the MSFs (continations) allows to revert encapsulated data AND behaviour (something quite important if used in ABS), further the lack of side effects guarantees that we can re-compute events

primary rolling back in Haskell can be achieved without anti-messages

global controller runs in STM

contribution: employing FRP, focus on Rollback and lack of side-effects, lock-free STM allows concurrency/parallelism but guarantees rollback. if we need IO, we can write IO actions to the global Controller (but how to roll them back?)

bezirgianis thesis: Testing of HDES lazy Evaluation is a joke because it doesnt relate it to the computations actually required for the simulation which would be much more interesting

what we are not doing: distributed PDES e.g. using cloud haskell, also this has been solved in aivika

2 RELATED WORK

In his masterthesis [1] the author investigated Haskells parallel and concurrency features to implement (amongst others) *HDES*, a lazy PDES framework. TODO: read this part

3 BACKGROUND

TODO: Parallel Discrete Event Simulation, Fujimoto [2] TODO: Parallel Discrete Event Simulation The Making Of A Field, Fujimoto [3] TODO: Parallel Discrete-Event Simulation Framework, Miranda [5] TODO: Parallel Discrete-Event Simulation Applications, Tropper [6] TODO: Parallel DiscreteâĂŘEvent Simulation, Liu [4] TODO: DEVS Book, Zeigler [7]

ACKNOWLEDGMENTS

The authors would like to thank

REFERENCES

[1] Nikolaos Bezirgiannis. 2013. *Improving Performance of Simulation Software Using Haskells Concurrency & Parallelism.* Ph.D. Dissertation. Utrecht University - Dept. of Information and Computing Sciences.

[2] Richard M. Fujimoto. 1990. Parallel Discrete Event Simulation. *Commun. ACM* 33, 10 (Oct. 1990), 30–53. https://doi.org/10.1145/84537.84545

[3] R. M. Fujimoto, R. Bagrodia, R. E. Bryant, K. M. Chandy, D. Jefferson, J. Misra, D. Nicol, and B. Unger. 2017. Parallel discrete event simulation: The making of a field. In *2017 Winter Simulation Conference (WSC).* 262–291. https://doi.org/10.1109/WSC.2017.8247793

[4] Jason Liu. 2011. Parallel Discrete-Event Simulation. In *Wiley Encyclopedia of Operations Research and Management Science.* American Cancer Society. https://doi.org/10.1002/9780470400531.eorms0639

[5] M. Mann, R. Miranda, and A. Alavarado. 2003. Parallel discrete-event simulation framework. In *23rd International Conference of the Chilean Computer Science Society, 2003. SCCC 2003. Proceedings.* 41–48. https://doi.org/10.1109/SCCC.2003.1245444

[6] Carl Tropper. 2002. Parallel Discrete-Event Simulation Applications. *J. Parallel Distrib. Comput.* 62, 3 (March 2002), 327–335. https://doi.org/10.1006/jpdc.2001.1794

IFL'18, August 2019, Lowell, MA, USA
2019. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[7] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press. Google-Books-ID: REzmYOQmHuQC.

Received May 2018