UNIVERSITÀ DELLA SVIZZERA ITALIANA

Meeting - 14.03.2014

Simon Maurer

March 13, 2014

1 Problem Statement, Title and Abstract

The document "problemStatement.pdf" presents the complete problem statement of the master thesis. Remarks, suggestions and corrections are appreciated.

Concerning the title of the master thesis I have no better proposition as "Accelerator for Event-based Failure Prediction". I think this title should be improved but at the moment I cannot com up with something better. A good start would be to give the algorithm of Felix a catching name.

I was asked to write an abstract in order to upload something to the ALaRI web page. The final abstract I will write only at the very end of the work, in order to also being able to mention the achievements and results. For now I propose the following:

> In today's live it becomes increasingly important, that computer systems are dependable. The reason being, that computer systems are used more and more in areas where the failure of such a system can lead to catastrophic events.
>
> In the event of a system failure it is of course desirable to fix the system as soon as possible in order to minimize the downtime of the system (maximize the availability). This can be accomplished by using different types of recovery techniques, e.g. Check-pointing (create checkpoints to roll back/forward), system replication (switch to a redundant system), fail over (reboot). All these techniques require a certain amount of time to complete the recovery process, time that is very expensive. In order to minimize this time, techniques have been developed to anticipate upcoming failures.
>
> The work of this master thesis consists in designing a hardware accelerator for such a failure prediction algorithm. The algorithm in question has excellent prediction capabilities but needs a lot of computation power and hence computa-

tion time. The goal of the resulting accelerator will remedy this problem and offer an exact and fast solution for predicting failures.

2 Arguments Justifying the Work

The email of Felix left some doubts to whether the acceleration of the algorithm is useful. The following list will give some arguments to justify the work.

Too many parameters to be identified, estimated and set
: Considering an embedded system, this is usually not a problem because the parameters are defined during the design phase and will never be changed afterwards.

Limited performance scalability
: There are studies available claiming otherwise. The discussion of Neumanns work will provide some arguments against this statement.

Industry trends point towards cloud
: In embedded systems it will still be beneficial to predict failures of single nodes. It is however important to keep the power and computational footprint low. This will be one of the major challenges. On the other hand, I think it would also be possible to also use this algorithm to monitor a distributed system and predict failures. It is only a matter of getting defining the events to feed to the algorithm.

3 Discussion of Neumanns Work

By only considering the computation complexity of the forward algorithm, a theoretical speedup of factor N should be possible. This number is of course limited by the parallelization overhead, the available computation units and the data flow. In his work,

Following some thoughts about the implementation of Neumann:

- implementation of classical forward algorithm, not the adapted version proposed by Felix. The adaptation requires more memory access and more computation steps but does not change the complexity order.
- high number of states compared to the work of Felix
- other work achieved better results on older GPUs (but with ignoring scaling). I believe this has been achieved by using a matrix multiplication approach (memory optimization).
- limited parallelization but optimization potential in data flow and single operation optimizations (FPGA)
- FPGA may be a better solution if also power consumption is taken into account. However due to a lower clock speed the speedup may not improve or even be smaller.