Optimized Calculation of Timing for Parallel Beam Operation
at the FAIR Accelerator Complex

A. Schaller, J. Fitzek - GSI, Darmstadt, Germany
Prof. Dr. F. Wolf, Dr. D. Lorenz - TU Darmstadt, Germany

Abstract
For the new FAIR accelerator complex at GSI the settings management system LSA is used. It is developed in collaboration with CERN and will now be in execution strictly serial. Nowadays the performance gain of single core processors has nearly stagnated and multicore processors dominate the market. This evolution forces software projects to make use of the parallel hardware to increase their performance. In this thesis LSA is analyzed and parallelized using different parallelization patterns like task and loop parallelization. The most common case of user interaction is to change specific settings so that the accelerator performs at its best. For each changed setting, LSA needs to calculate all child settings of the parameter hierarchy. To maximize the speedup of the calculations, they are also optimized sequentially. The used data structures and algorithms are reviewed to ensure minimal resource usage and maximal compatibility with parallel execution. The overall goal of this thesis is to speed up the calculations so that the results can be shown in an user interface with nearly no noticeable latency.

Motivation
To allow the commissioning and operation of FAIR, the software used today has to be optimized. The Crying (YR), its local injector, acts as a test facility for the new control system and in special for the control systems central component, the settings management system LSA. For the last YR commissioning beam time, about 3 700 manual trims were calculated per week with 80 working hours, which is about one trim every 72 seconds. Since the YR is a very small accelerator ring, with a circumference of approximate 54 m, everything worked fine. The waiting time summates to about 18 minutes, however, the human reaction time is not much less. But when it comes to calculate the Heavy Ion Synchrotron 18 (SIS18) or SIS100, the calculations get very slow. To calculate 3 700 trims for the SIS18, with its approximate 216 m, an operator would have to wait for over 13 hours. The SIS100, with approximate 1 100 m, calculation would even take over 91 hours.

Speedup
The speedup represents a factor, that shows how two different algorithms perform on the same task. In the context of parallelization, the speedup is a factor that indicates how much the parallel algorithm is faster than the sequential one. It is given by

\[ \text{Speedup} = \frac{T(n)}{T(1)} \]

where \( T(n) \) is the total execution time on a system with \( n \) processing units.

Optimizations
Sequential
- use caching where possible
- use suitable data structures for the main use case
- reduce array copies when inserting (or deleting) multiple points to a function
- change algorithms with complexity \( O(n^2) \) to those with \( O(\log(n)) \) where possible
- don’t calculate a setting for all its parents but only once all its parents have been calculated

Parallel
- run static data preparation in parallel
- run calculation loops in parallel where possible
- use parameter hierarchy as a task graph

Optimization Results
Average execution times on the target platform (10 cores with HT, 64 GB RAM) where each scenario was run twice for warmup and five times for measurements. The parallel execution was measured with the default threadpool size of 19 plus the main thread.

Work Depth Model
The Work Depth Model, described by Blelloch, allows to compare the execution time of parallel algorithms. Especially when using trees for parallelization, like the parameter hierarchy in LSA, other comparison mechanism don’t fit to the problem. In the context of parallelizing LSA, the work \( W \) is expressed by the amount of settings to be calculated and the depth \( D \) is expressed by the depth of the parameter hierarchy. Using Equation 5 of Blelloch, a range for \( T \) can be calculated for a given number of processing units \( P \) as the time depends on the hardware.

\[ W = \frac{P^2}{P + D} \]

With respect to Equation 1 we can say that the speedup in the Work Depth Model is

\[ \text{Work Depth Model} = \frac{W}{W'} < \frac{P}{P'} \]

Work Depth Model for
- 4 cores: speedup is between 3.92 and 4.00
- 10 cores: speedup is between 9.49 and 10.00

The parallel speedup on the target platform with 10 cores has an efficiency of 9.092%, on the test platform with 4 cores the efficiency is 0.987 (see Equation 4), which nears is a so called perfect linear speedup where \( E = 1 \).

The following image shows the memory consumption on the target platform for scenario P2-3. The scenario was run twice for warmup and five times for the measurements.

For this chart, each scenario was executed 2 times for warmup and 5 times for measurements on a test platform (4 cores, 12 GB RAM) with and without Hyper Threading Technology (HT). The default threadpool size is \( n = 1 \), so for the setup without HT, the default threadpool size is 1 and with HT the default threadpool size is 7.