

**Praktikum Sekundärspeicheralgorithmen  
Algorithmik II**

Sanders, Dementiev, Schultes, Singler SS 06

<http://algo2.iti.uka.de/ioprakt06.php>

---

## Experiment 1

Deadline: 13:00 — 8 May 2006

### Exercise 1

Reading assignment:

- Read Chapter 1 from [MSS03] to make yourself familiar with the concept of I/O-efficient algorithms (also known as secondary memory algorithms).
- To become familiar with the design of the STXXL library read [DKS05].

### Exercise 2

STXXL installation. Download the library from <http://stxxl.sourceforge.net/>. Install the library into your home directory following the installation manual.

### Exercise 3

Your first STXXL program (Computing prefix sums of very long sequences).

**Definition:** For sequence  $\{a_0, a_1, a_2, \dots, a_i, \dots, a_{n-1}\}$  the prefix sum sequence is the sequence  $\{a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots, \sum_{j=0}^i a_j, \dots, \sum_{j=0}^{n-1} a_j\}$

Write a program using STXXL that computes the prefix sum of a sequence of random floating point numbers.

- Use the C++ type `double` to represent floating point numbers.
- To store input and output use `stxxl::vector`. The architecture and the methods of `stxxl::vector` are described in [http://algo2.iti.uka.de/dementiev/files/stxxl\\_tutorial.pdf](http://algo2.iti.uka.de/dementiev/files/stxxl_tutorial.pdf). The methods have almost the same syntax and semantics as the STL `std::vector`. See also the Doxy documentation <http://algo2.iti.uka.de/dementiev/stxxl/doxy/html/> and examples therein.
- For filling the input vector with random numbers use `stxxl::generate`.
- For computing the prefix sum values use `stxxl::for_each_m` which is described in STXXL manual.

Tuning:

- Your external memory STXXL file (configured in `.stxxl`) should be located on the pool computers in `/data/prakt/` directory.
- Run your program for inputs of length 128 MB, 256 MB, 512 MB, 1 GB, 2 GB, 4 GB. Plot (e. g. gnuplot, you can find a tutorial at <http://www-128.ibm.com/developerworks/library/l-gnuplot/?ca=dgr-lnxw04gnuplot>) the processing rate (items per second) for these inputs using your favorite program. The x-axis should have logarithmic scale.
- For one input size (for example 512 MB) try different block sizes for `stxxl::vector`: 64 KB, 128 KB, ..., 8 MB. Plot figures to find the best block size.
- Change the number of pages in `stxxl::vector`. Try the values 1, 2, 4. Check if the performance varies.

- Try to tune the number of buffer blocks used in `stxxl::for_each_m` and `stxxl::generate` functions. Plot figures to find the best values, again.

Write a short report that includes the figures you have plot. In your explanations the following points should be present:

- The role of the block size on the performance of the algorithm.
- For which access patterns `stxxl::vector` should have multiple pages.
- An estimation of the main memory requirements of your program (see STXXL manual for the memory consumption of `stxxl::vector`).

Send your source code and your report with figures to `dementiev@ira.uka.de` before the deadline. Also make an appointment with Roman Dementiev for the defense of your work.

## References

- [DKS05] R. Dementiev, L. Kettner, and P. Sanders. Stxxl: Standard Template Library for XXL Data Sets. Technical Report 18, Fakultät für Informatik, University of Karlsruhe, 2005. [http://i10www.ira.uka.de/dementiev/files/TRKA2005\\_18.pdf](http://i10www.ira.uka.de/dementiev/files/TRKA2005_18.pdf).
- [MSS03] U. Meyer, P. Sanders, and J. Sibeyn, editors. *Algorithms for Memory Hierarchies*, volume 2625 of *LNCS Tutorial*. Springer, 2003. <http://link.springer-ny.com/link/service/series/0558/tocs/t2625.htm>.