Algorithm Engineering

Jens K. Mueller

jkm@informatik.uni-jena.de

Department of Mathematics and Computer Science
Friedrich Schiller University Jena

Monday 24th November, 2014

Sorting Algorithms

Mergesort

Mergesort

1. Half input
2. Recurse
3. Merge sorted halves

Algorithmic improvements

- Allocate additional memory once
- Improve merge
- Insertion sort on small sizes

Profiling

Profiling

Find code that is worth optimizing

Strategy

- Profile on real(istic) input
- Optimize identified hot spot(s)

Resources

- CPU
- Memory

Techniques

- Instrumentation
- Statistical
- Trapping on predefined events
  Java, .NET, ...

Profiling with gcc and gprof

Instrumentation

gcc's -pg and gprof

- Compile and link with `-pg`
- Run binary (creates `gmon.out`)
  ```
  $ ./binary
  ```
- Time per function
  ```
  $ gprof /path/to/binary /path/to/gmon.out
  ```
- Time per line
  ```
  $ gprof -l /path/to/binary /path/to/gmon.out
  ```

Be careful with inlining and compiler optimization in general while profiling.

Profiling with gperftools

Statistical

gperftools

- Link with `-lprofiler` and run
  `$ CPUPROFILE=/path/to/profile /path/to/binary`
  or
- `$ LD_PRELOAD=/path/to/libprofiler.so CPUPROFILE=/path/to/profile /path/to/binary`
- Samples per function
  `$ pprof --text /path/to/binary /path/to/profile`
- Samples per line
  `$ pprof --list=function_name /path/to/binary /path/to/profile`