TODO

NO

May 5, 2015

Contents

1 TODO **2**

2 Start report (NO) **2**

3 Profiling **2**

4 Easy solutions with omp critical **2**

5 Misc tests **2**

6 Reporting structure improvements **2**

7 Start testing on Povel **3**

8 Optimizing flags / compilers **3**

9 MPI? (if so on R-side, or?) **3**

10 Instructions **3**

10.1 Build & compile 3

10.1.1 Makevars 4

10.2 Project instructions from PDC 4

10.2.1 For some application and HPC architecture of your choice: 4

10.2.2 The project is not about: 4

10.2.3 So: 4

10.2.4 Examples: 4

10.2.5 After the course 4

10.2.6 Now - during lab-afternoons 5

10.3 Run valgrind from R 5
10.3.1 To explore and make callgrind readable: 5

1 TODO

2 Start report (NO)

3 Profiling

- profile R (AK)
- profile valgrind (NO)
- Baseline profiling (AK)

4 Easy solutions with omp critical

- omp critical -> omp atomic write/update? (AK)
- profile again

5 Misc tests

- change from dynamic to static
- if keep critical consider naming them (possible speedup) (AK)
- what does nowait do? speedup? (NO)

6 Reporting structure improvements

- Can summary$pt and summary$prev be merged? $prev and $pt are almost identical $pt allocates patient-time whereas $prev allocates number of individuals, all other columns are identical and their sizes seems to be very similar. !suggest a single structure with one more column.

> tmp <- callFhcrc(n=1e6)
user system elapsed
40.731 0.041 10.695
> dim(tmp$summary$pt)
122175 8
> dim(tmp$summary$prev)
116545 8

- $events has one more factor, $event - the event type and does not fit in that structure.
- Could the dynamic mapping be changed for a static sparse structure? According to the table below it would be 432000x8 matrix. The sparsity would result in overhead (and look bad if not post-processed) with small simulations, but the static allocation could save overhead for large simulations (when it is needed).

state	grade	dx	psa	cohort	age	number of combinations
3	3	3	2	(1980-1900)	100	432 000

- A consequence of using a static structure maybe that openMP's reduction() could be used. After a quick web search: it looks like it does not operate directly on arrays.

7 Start testing on Povel

8 Optimizing flags / compilers

9 MPI? (if so on R-side, or?)

10 Instructions

10.1 Build & compile

Howto install and compile the R-package:

```
https://github.com/mclements/microsimulation
shell:  git clone https://github.com/mclements/microsimulation.git
R: install.packages("BH")
R: install.packages("Rcpp")
shell:  R CMD INSTALL path_to_microsimulation
```

10.1.1 Makevars

On Ferlin, the gcc compiler used for R is 4.6.0, which only has expereminetal support for C++11 standard. The flag to provide is -std=c++0x in lieu of -std=c++11.

10.2 Project instructions from PDC

10.2.1 For some application and HPC architecture of your choice:

- Develop efficient program for non-trivial problem
- Demonstrate and report how efficient it is.
- 4.5 ECTS = 3 weeks of work incl. report writing

10.2.2 The project is not about:

- Substantial development of new code.
- Scientific results obtained with code

10.2.3 So:

- Prioritize measurements and analysis/interpretation!
- Demonstrate use of tools (profiling, ...) , and simple performance model.
- NO TIME for development of new significant code.

10.2.4 Examples:

- Parallelize a code you know and/or work with; choose interesting part.
- Write a simple code for key algorithm of bigger solution process
- Write a simple code for a simple problem

10.2.5 After the course

- Start the work ASAP:
- Finish the work; Get in touch with tutor

- Submit report to tutor. The report will be graded and sent back with comments; you may have to complete some parts and hand in again. We need email and paper mail address!
- KTH students: LADOK
- Other students: Certificate will be sent to you

10.2.6 Now - during lab-afternoons

- Discuss with instructors & course participants, form groups of size G.
- Define project and choose tutor: Michael, Jonathan, Erwin, Stefano
- Write very short synopsis, check with supervisor
- Submit synopsis to summer-info@pdc.kth.se before end of HPC course

10.3 Run valgrind from R

Howto run valgrind from shell:

```
   R --vanilla -d "valgrind --tool=memcheck --track-origins=yes"
< ~/src/ki/microsimulation/doc/RunSim.R
   R --vanilla -d "valgrind --tool=callgrind" < ~/src/ki/microsimulation/doc/RunSim.R
```

10.3.1 To explore and make callgrind readable:

- https://github.com/jrfonseca/gprof2dot
  - `gprof2dot -f callgrind < callgrind.out.18739 | dot -Tpng > profile.png` plot callgrind output
  - `gprof2dot -f callgrind < callgrind.out.18503 | dot -Tpdf > profile.pdf` plot to pdf
  - `gprof2dot -z callFhcrc -f callgrind < callgrind.out.4596 | dot -Tpdf > profile.pdf` set root function
  - `gprof2dot -s -z callFhcrc -f callgrind < callgrind.out.9822 | dot -Tpdf > profile.pdf` skip arguments to functions
- kCachgrind