Benchmarking Environment - Feature #1010

Tasks # 824 (In Progress): Measure CPU time via callgrind

Valgrind as intrinsic execution engine

06/20/2012 12:04 AM - M. Rolf

Status:	In Progress	**Start date:**	06/19/2012
Priority:	Normal	**Due date:**	
Assignee:	M. Rolf	**% Done:**	10%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Allow benchmark calls that execute valgrind in background. Something like

```
MyBenchmarkExecutable --engine cpu
Benchcase [mu_suite:my_case]
  Total (corrected) time:    1.3 s
  Estimated cost per operation:    1.7 us
  Operations per second:    5.9e5
```

vs.

```
MyBenchmarkExecutable --engine valgrind
Benchcase [mu_suite:my_case]
  Total (corrected) time:    1.3 GCycles
  Estimated cost per operation:    1.7 MCycles
  Operations per second:    5.9e5
```

The "--engine valgrind" must call valgrind in background, read and parse the logfile and transfer the results into the internal result data-structure.
Some ideas...

- Interally disable warmup and init-count (pointless when using valgrind)
- Automatically reduce repitition-count, e.g. by factor 100

History

#1 - 06/20/2012 01:16 PM - M. Rolf

- Parent task set to #824

#2 - 06/20/2012 02:49 PM - M. Rolf

Callgrind generates an aweful output format:
http://kcachegrind.sourceforge.net/html/CallgrindFormat.html

Better use callgrind_annotate to collect infos in a more comprehensive, yet still ASCII and new-line/blank-line separated, format:

```
callgrind_annotate --inclusive=yes --threshold=100 <callgrind.out>
```

This tool cannot write to files, so we would need to do some bash-pipe stuff. Not very platform-independent ^^

#3 - 06/20/2012 02:55 PM - M. Rolf

- % Done changed from 0 to 10

#4 - 06/20/2012 02:55 PM - M. Rolf

- Status changed from New to In Progress