Valgrinding Sage For Fun and Profit

Michael Abshoff
University of Dortmund

Oct. 1st, 2007

The Problem

Valgrind is very cool, but not eady to use, specifically in the case of Python. Sage on top causes additional issues and requires some time to figure it all out.
Other additionally issues exist:

- reference counts of cython classes, i.e. the dictionary problem
- the python garbage collector doesn't "know" about objects in Cython
- the output from a simple "startup + quit" of Sage results in a 1.3 mb log. Before using valgrind the file was larged than 5 mb, so we have made a lot of progress, in Sage 2.8.5.1 there are no "definitely lost" left for startup + quit.
- Screwing up with Cython code causes all kinds of false leads - John Voight example

Overview

We will cover the following issues:

- An Introduction to Valgrind
- Compiling Python for Valgrind
- Compiling Sage spkgs with Debug Options
- Valgrind Integration into Sage
- Valgrind Tools - memcheck
- Valgrind Tools - massif
- Valgrind Tools - callgrind
- Valgrind Tools - cachegrind
- Valgrind Tools - omega
- Trouble in Paradise: valgrind vs. pari or GAP

An Introduction to Valgrind

- x86, x86-64, ppc 32, ppc 64 under Linux
- ppc 32 and ppc 64 on AIX
- ports to *BSD are on the way
- OSX 10.5, a.k.a. Leopard, will have valgrind port done by Apple (x86, x86-64, not sure about ppc)

Compiling Python for Valgrind

- enable “- -without-pymalloc“ and “- -pydebug“ in spkg-install and recompile python.spkg
- Wishlist: special option that compile all packages with -O0 and -g - easier to read stack traces
- python even with “- -without-pymalloc“ loads of errors
- “- -without-pymalloc“ makes python run about 4 to 5 times slower compared to the default.

Compiling Sage spkgs with Debug Options

- FIXME: it is on the wishlist, no specfic issues have been solved
- Singular: make omalloc use malloc instead of mmap?

Valgrind Integration into Sage

- We need a recent 3.3.0svn version, otherwise it segfaults when valgrinding python
- Sage's gmp with the Core Duo and also the Opteron patch requires a even more recent 3.3.0svn snapshot
- sage -[memcheck|massif|callgrind|cachegrind], -valgrind is an alias for memcheck
- works fine together with -testall and -t, clashes with -gdb
- currently defaults to sane options (IMHO), but should get automated via some environment variables

Valgrind Tools - memcheck

- check for definitely lost, possible lost and still reachable memory areas
- catches mismatched free/delete/delete[]
- has a limit of 1,000,000 errors - I have surpassed that number with a long computations - but there is a special option to count an unlimited number of errors
- roughly 50 times as slow, needs more memory - use your account on sage.math
- Gets slower when there are many errors, the current 3.3.0svn trunk has some optimizations
- log files end up $HOME/.sage

Valgrind Tools - massif

- profiles heap and/or stack
- slowdown factor?
- produces pretty postscript files
- plotting code was some wrapping issues when using more than 2^{31} or 2^{32} bytes - need to report bug to valgrind-devel
- postscript files end up in $CWD instead of $HOME/.sage - need to report buglet to valgrind-devel

Valgrind Tools - callgrind

- slowdown factor?
- counts instruction cycles spend in areas of the count
- with external tool can greatly help analyses where time is spend

Valgrind Tools - cachegrind

- slowdown factor?
- determines L1 and L2 cache hit - similar to callgrind - not particularly interesting to me

Valgrind Tools - omega

- experimental and out of tree, but extremly cool
- currently doesn't compile against 3.3.0svn trunk
- x86 and x86-64 only
- shows where memory is actually leaked, not where it is allocated

Trouble in Paradise: valgrind vs. pari or GAP

- when pari extends its heap valgrind throws an error
- GAP doesn't work under valgrind. Initial suspect was sbrk, but as it turned out in the end sbrk is not the problem - needs to be analyzed/fixed down the road

MoreTrouble in Paradise: valgrind vs. Singular

- Well, no real trouble, but since omalloc uses mmap:
- But: valgrind catches issues in libSingular