



Porting Sage open source mathematics software to OpenSolaris

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

<http://www.sagemath.org/>

Dr. David Kirkby

david.kirkby@onetel.net

19th May 2010

Contents of the presentation.

My own background

Uses of mathematical software

Availability of mathematics software

Rationale for Sage

Solaris 10 issues (works 32-bit only)

OpenSolaris issues (not all solved)

Some quick examples of Sage usage

Conclusions.

About me.

I'm a Chartered Engineer (electrical background).

I've a Ph.D. in Medical Physics (electronics/optics/computing)

I've used SunOs/Solaris for about 23 years.

I own a nice new 3.333 GHz Sun Ultra 27 and several old power-hungry SPARCs.

Mathematics is not my strongest subject, but I have used Mathematica quite a bit – much more than I've used Sage.

Four simple examples – more to follow.

```
sage: factorial(30)
265252859812191058636308480000000
```

```
sage: is_prime(12)
False
```

An RF mixer, used in TV's, radios etc, multiplies two frequencies together.

```
sage: var('x,y')
(x, y)
sage: f = sin(x)*cos(y)
sage: f.trig_reduce()
-1/2*sin(-x + y) + 1/2*sin(x + y)
```

```
sage: solve(x^2 + 3*x + 2 == 0, x)
[x == -2, x == -1]
```

```
sage: diff(sin(x) + x^3,x)
3*x^2 + cos(x)
```

Uses of maths packages

Mathematics

Science

Engineering

Finance

Education

Art

Fractals

Ringtones for phones

Cloth designs ...

Some maths packages

	Win/Mac/Linux	Solaris	OpenSolaris
Mathematica (commercial)	£2035	£2545	£2545
Maple (commercial)	£1695	£1695	N/A
MATLAB (commercial)	£1500 (Some add-ons >£5000)	£1500 (To be discontinued)	N/A
MAGMA (commercial)	>£750	?	?
Mathematica (home)	£195	NA	NA
Mathematica (student)	£80	NA	NA
MAGMA (student, limits exist.)	£69	NA	NA
MATLAB (student)	£50 (approx)	N/A	N/A
SAGE	Free	Free	Port needed

The rationale of Sage

Conceived by Professor William Stein, at the Department of Mathematics, University of Washington.

It's impossible to do real research if you use a black-box program, with little or no knowledge of the internal algorithms.

Building programs like Mathematica from scratch would be a huge amount of work. (Hundreds of man-years).

There was **lots** of open-source software available solving specific problems, so there was no need to reinvent the wheel. Just combine them together.

Sage now has around 200 developers worldwide – mostly academics from maths departments. (I'm an exception)

Sage – what it does.

Combines many (around 100) different open-source packages. Some are written in C, C++, Fortran, Lisp, Python etc.

Use Python as a programming language so user only needs to know Python.

Provide interfaces to some external programs. Some are free, but not GPL v2+, others are commercial like MATLAB and Mathematica.

The potential for Sage in industry.

Sage is already quite popular at academic institutions, (many maths departments teach Sage), but Sage is not currently popular in industry.

Collaboration between individuals at locations remote from each other is possible due to the server design and web interface. (See later)

The fact Sage is free makes collaboration less of a financial barrier.

Building Sage (all platforms)

Download a huge tar file (around 300 MB) – the individual components are compressed, so there is little to be gained by compressing the tar file.

Type 'make' then wait a few hours (or days).

Each component is extracted from a compressed file, built, then intermediate files are removed. This reduces disk usage needed to build Sage to about 2 GB.

Everything is included.

Source code for many 'standard' packages are included – zlib, Python, readline, iconv, termcap. There are good reasons for this, but it causes problems packaging Sage – Debian for example will not accept this.

Source code for all specialist packages – gap, GSL, R, Singular, rubiks, ratpoints, PolyBoRi, PARI, MPIR, MPFR, pil, palp, Pynac etc etc (about 90 of them).

A binary Fortran compiler for OS X.

Ports of Sage to various platforms.

Sage was initially developed on Linux and most of Sage is still developed on Linux.

A port has been completed to Solaris 10 (SPARC), but only as a 32-bit application.

A 64-bit port to Solaris 10 (SPARC) should be easy, assuming we don't hit any gcc bugs.

Some work has been done on porting Sage to:

AIX (very little)

HP-UX (very little)

OpenSolaris – Almost finished. Subject of this talk.

FreeBSD

Windows (based on Cygwin)

Building on Solaris / OpenSolaris

One must use GNU 'make' and GNU 'tar'.

gcc >= 4.0.1 with Fortran support.

gcc configured to use Sun linker and assembler on Solaris 10 (SPARC).

gcc configured to use the Sun linker and GNU assembler on OpenSolaris (x64).

The first assembler and linker in the PATH must be those used by gcc – not a great idea, but blame me!

In order to build a binary distribution, the GNU version of 'cp' is needed. But this is not needed if building from source.

Problems porting to Solaris 10

Lack of suitable hardware.

My own Blade 2000 (2 x 1200 MHz, 8 GB RAM).

Sun T5240 donated by Sun. 2 x 1167 MHz, 16 cores, 128 threads and 32 GB RAM.

Sun Blade 2500 1.28 GHz.

Never made use of VirtualBox – probably silly.

Many Sage developers not keen to test on Solaris – a situation made worst by the lack of suitable hardware.

Current status

Sage builds fully on Solaris 10 (SPARC) as a 32-bit application and passes all self-tests. It is not currently officially supported on Solaris 10, though version 5.0 will be.

Sage does not yet build fully on OpenSolaris. There are some outstanding issues.

The build process on both Solaris and OpenSolaris is a bit tricky. In particular, linker/assembler issues.

Solaris specific issues.

Many Sage components make use of linker options which are not portable between Sun's linker and the GNU linker from 'binutils'. Options associated with shared libraries were particularly troublesome.

Sun linker	GNU linker
-G	-shared
-h	-soname
-z allextract	--whole-archive
-z defaultextract	--no-whole-archive

GNUisms.

Many components of Sage assume a 'GNU' Linux environment. They make assumptions about the options supported by 'cp', 'tar', 'make', 'ar', 'ld', etc which are not POSIX, and not supported by Solaris/OpenSolaris. I've tried to get people to use only POSIX options, but this has been a hard struggle.

Developers of parts used in Sage get notified of problems. They don't always respond, but some do and make their code more

'lcalc' tried to hide assembler warnings!

```
drkirkby@kestrel:~/L-1.23/src$ make
make libLfunction.so
make[1]: Entering directory `/export/home/drkirkby/L-1.23/src'
g++ -Wa,-W -O3 -Wno-deprecated -ffast-math -fPIC -I../include -c
Lglobals.cc
/usr/ccs/bin/as: error: unknown option 'W'
usage: /usr/ccs/bin/as [-V] [-Q{y,n}] [-q] [-s]
      [-S] [-K {pic,PIC}] [-o objfile] [-L] [-T]
      [-P [[-Yc,path] [-Ipath] [-Dname] [-Dname=def] [-Uname]]...]
      [-m [-Ym,path]] [-n] [-ul] [-xF]
      [-m32] [-m64]
      [-xarch={v7,v8,v8a,v8plus,v8plusa,v8plusb,v9,v9a,v9b,sparc,sparcvis,
sparcvis2,sparcfmaf,sparcima}]
      [-xcode={pic13,pic32}] file.s...
make[1]: *** [Lglobals.o] Error 1
```

Despite making him aware of the non-portability of the `-Wa,-W` option, the latest version of 'lcalc' still has this, so Sage has its own Makefile for 'lcalc'.

memset() bug on 'sun4v' CPUs.

```
#include <string.h>
#include <stdio.h>
int main (void)
{
    int i;
    int a = 1 , b = -1;
    char c = 1, d = 1;
    printf("The original values are:  c=%d d=%d\n", c, d);
    printf("a=%d, b=%d sum=%d \n",a,b,a+b);
    for (i = 0; i<= 8; ++i) {
        memset(&c, 2, a + b); /* a+b = 0, so memset should not modify 'c' */
        printf("After calling memset which should NOT change c, i=%d c=%d
d=%d\n", i, c, d);
        memset(&d, 2, a + b); /* Again, no modification should take place
of 'd' */
        printf("After calling memset which should NOT change d, i=%d c=%d
d=%d\n\n", i, c, d);
    }
    return 0;
}
```

Library failed if SunStudio was installed

One really annoying problem, which actually took several months to solve, was that the Sage library (written in Python, but compiled to C), would build on Solaris 10 if SunStudio was not installed, but would fail to build if SunStudio was installed. This was irrespective of whether SunStudio was in the path or not, so the only way to get around the issue was to log in as root and move SunStudio to a non-standard location.

```
# mv /opt/sunstudio12.1 /opt/sunstudio12.1.foobar
```

The problem was due to the SCons build system, which few understood. Finally we solved it.

Why OpenSolaris port should be easy.

Since Sage builds on Solaris 10 SPARC, many Solaris issues are resolved.

I've personally got a fast OpenSolaris machine (Sun Ultra 27, quad core 3.333 GHz, 12 GB RAM) – thanks to some lightning which damaged my Sun Blade 2000.

VirtualBox should allow others to build/test at a reasonable speed. Convincing them to do so might be harder though.

ECL (Lisp interpreter) not building.

```
/ecl-9.10.2-20091105cvs.pl/src/src/c/arch/ffi_x86.d -> ffi_x86.c
gcc -DECLDIR="\ /export/home/drkirkby/sage-4.3.1/local/lib/ecl-9.10.2\" -I.
-I/export/home/drkirkby/sage-4.3.1/spkg/build/ecl-9.10.2-20091105cvs.pl/src/build
-I/export/home/drkirkby/sage-4.3.1/spkg/build/ecl-9.10.2-20091105cvs.pl/src/src/c -I../
ecl/gc -DECL_API -DECL_NO_LEGACY -I/export/home/drkirkby/sage-4.3.1/local/include
-O2 -m64 -g -Wall -fPIC -Dsun4sol2 -c -o ffi_x86.o ffi_x86.c

/var/tmp//ccvhai7u.s: Assembler messages:
/var/tmp//ccvhai7u.s:49: Error: suffix or operands invalid for `mov'
/var/tmp//ccvhai7u.s:51: Error: suffix or operands invalid for `mov'
/var/tmp//ccvhai7u.s:136: Error: suffix or operands invalid for `mov'
make[4]: *** [ffi_x86.o] Error 1
make[4]: Leaving directory
`/export/home/drkirkby/sage-4.3.1/spkg/build/ecl-9.10.2-20091105cvs.pl/src/build/c'
make[3]: *** [libeclmin.a] Error 2
make[3]: Leaving directory
`/export/home/drkirkby/sage-4.3.1/spkg/build/ecl-9.10.2-20091105cvs.pl/src/build'
make[2]: *** [all] Error 2
make[2]: Leaving directory
`/export/home/drkirkby/sage-4.3.1/spkg/build/ecl-9.10.2-20091105cvs.pl/src'
Failed to build ECL ... exiting
```

Cddlib builds 32-bit still (unknown)

```
ld: fatal: file /usr/local/lib/libgmp.so: wrong ELF class:
ELFCLASS32
ld: fatal: file processing errors. No output written to
.libs/scdd_gmp
collect2: ld returned 1 exit status
make[1]: *** [scdd_gmp] Error 1
make[1]: Leaving directory
`/export/home/jaap/sage_port/sage-4.3.2.alpha1/spkg/build/cd
dlib-094f.p5/src/src-gmp'
make: *** [all-recursive] Error 1
Error building cddlib
```

PARI reports the C compiler is broken.

```
Configured with: ../gcc-4.3.4/configure --prefix=/usr/local/gcc-4.3.4/  
--with-as=/usr/local/binutils-2.20/bin/as --with-ld=/usr/ccs/bin/ld  
--without-gnu-ld --enable-languages=c,c++,fortran
```

```
Thread model: posix
```

```
gcc version 4.3.4 (GCC)
```

```
*****
```

```
Configuring pari-2.3.3 (STABLE)
```

```
Checking echo to see how to suppress newlines...
```

```
...using \c
```

```
Looking for some tools first ...
```

```
...ld is /usr/ccs/bin/ld
```

```
...zcat is /usr/bin/zcat
```

```
...gzip is /usr/bin/gzip
```

```
...ranlib is /usr/ccs/bin/ranlib
```

```
...perl is /usr/bin/perl
```

```
...I could not find emacs.
```

```
*****
```

```
* C compiler does not work. PARI/GP requires an ANSI C compiler! *
```


GNU TLS – an undefined reference.

```
gcc -std=gnu99 -g -O2 -D_REENTRANT -D_THREAD_SAFE -I/home/wstein/screen/fulvia/sage-4.4/local/include -g -O2 -D_REENTRANT -D_THREAD_SAFE -Wno-pointer-sign -o .libs/gnutls-serv serv-gaa.o serv.o common.o select.o -L/home/wstein/screen/fulvia/sage-4.4/spkg/build/gnutls-2.2.1.p5/src/lib/.libs ../lib/.libs/libgnutls.so -L/home/wstein/screen/fulvia/sage-4.4/local/lib ../libextra/.libs/libgnutls-extra.so /home/wstein/screen/fulvia/sage-4.4/local/lib/libopencdk.so /home/wstein/screen/fulvia/sage-4.4/spkg/build/gnutls-2.2.1.p5/src/lib/.libs/libgnutls.so -lz ../gl/.libs/libgnu.a /home/wstein/screen/fulvia/sage-4.4/local/lib/libgcrypt.so /home/wstein/screen/fulvia/sage-4.4/local/lib/libgpg-error.so -lsocket -R/home/wstein/screen/fulvia/sage-4.4/local/lib
serv.o: In function `peer_print_info':
/
home/wstein/screen/fulvia/sage-4.4/spkg/build/gnutls-2.2.1.p5/src/src/serv.c:489: undefined reference to `gnutls_x509_cert_print'
common.o: In function `print_x509_info':
/
home/wstein/screen/fulvia/sage-4.4/spkg/build/gnutls-2.2.1.p5/src/src/common.c:151: undefined reference to `gnutls_x509_cert_check_hostname'
../libextra/.libs/libgnutls-extra.so: undefined reference to `_gnutls_hostname_compare'
collect2: ld returned 1 exit status
```

SYMPOW 64-bit and x86 issues.

SYMPOW is used to compute special values of symmetric power elliptic curve L-functions. (I do not have a clue what L-functions are).

There are at least two problems with SYMPOW.

SYMPOW reports my Ultra 27 is not x86 so does not use `fpu.c`. Clearly this is incorrect, as it's an Intel Xeon processor. SYMPOW fails to add the `-m64` option to build 64-bit code, so will produce 32-bit code. Numerous parts of Sage need to be told to build 64-bit, when the environment variable `SAGE64` is set to "yes".

What I don't like about Sage.

Sage is primarily developed by mathematicians. Some input from computer scientists, professional software developers, system admins, could improve Sage immensely in my opinion.

'Stable' releases are too frequent

Often critical bug fixes needed to be hurried out.

Version numbers x.y.z mean very little.

Insufficient testing before a release.

Sage is not as polished as Mathematica/Maple/MATLAB.

Few engineering examples – a chicken and egg situation, which will hopefully change when I start to use Sage on OpenSolaris.

Please help if you can.

Help solve the few remaining build issues for OpenSolaris.

Expertise in SPARC assembler would be useful for some components, where C is used.

Help package Sage for binary distribution.

Need an SMF start-up script – currently using `/etc/init.d/sage`

Use Sage and produce some example for non-mathematicians, like Mathematica has.

Links

Main Sage webpage

<http://www.sagemath.org/>

A public-access Sage server running on Linux

<http://www.sagenb.org/>

A public-access Sage server running on a Sun T5240 (Solaris 10 5/09)

<http://t2nb.math.washington.edu:8000>

sage-solaris@googlegroups.com

sage-devel@googlegroups.com

Conclusions.

Sage is an open-source GPL v2+ maths package, which is currently primarily developed by mathematicians. It's free, open-source and web-based interface give it the potential to be used more in industry.

Sage currently build relatively easily on Linux, with a bit of care on Solaris 10 (SPARC) in 32-bit mode, but not on OpenSolaris on x64. No attempt has been made to build Sage on Solaris 10 on x86, or OpenSolaris on SPARC.

Solving the remaining OpenSolaris issues on x64 should not be a huge task. More developers would obviously be very helpful!