

State of SMC

William Stein, Sage EDU Days 6

6/16/2014

Contents

1	State of SageMathCloud	1
1.0.1	William Stein	1
1.0.2	June 2014 at Sage Edu Days 6	1
2	History	1
3	Features	2
4	Users	3
5	Weekly Returning Users	4
6	Users of Sagemath.org	5
7	SMC Goals and Motivation	5
8	What is SMC?	6
9	This Summer	7

```
%load talk.css
```

```
def slide(s):  
    md(s,hide=True)  
    print "\n"*10  
    coffeescript("""  
output = cell.opts.output  
if not output.hasClass('salvus-slide')  
    output.addClass('salvus-slide')  
    """)
```

```
https://cloud.sagemath.com/blobs/talk.css?uuid=5de08068-6131-412f-a0bb-cf40b87495dc
```

1 State of SageMathCloud

1.0.1 William Stein

1.0.2 June 2014 at Sage Edu Days 6

2 History

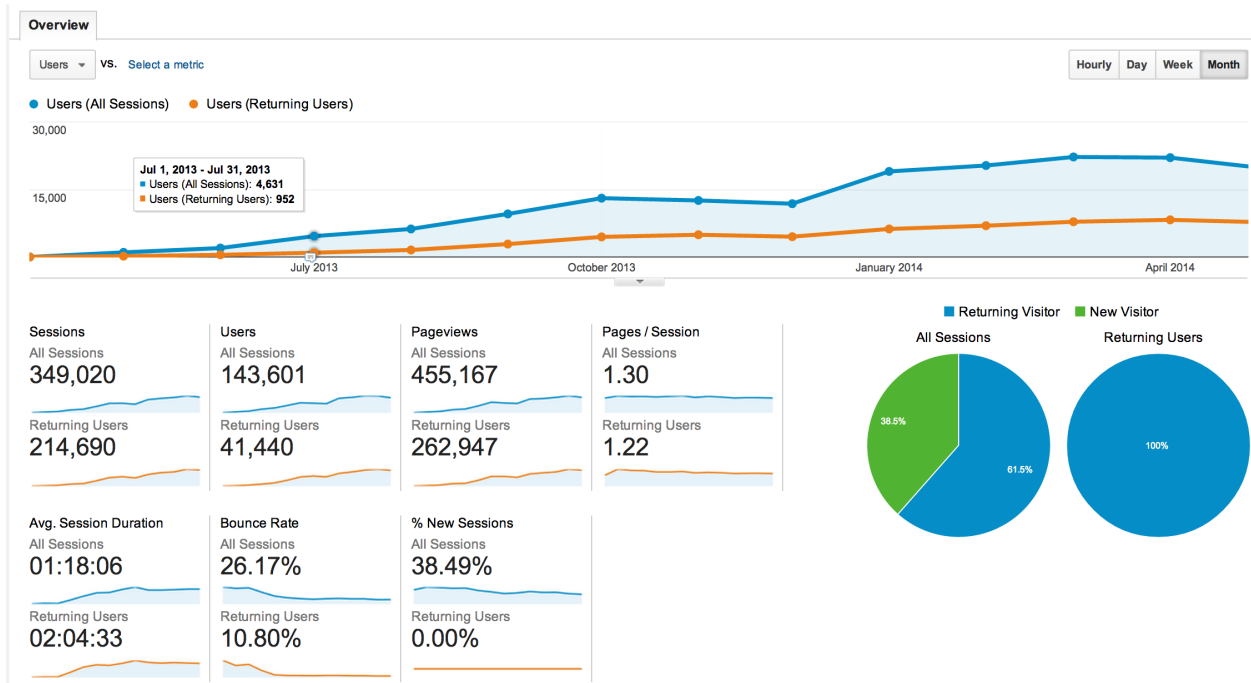
- May 2012: Started thinking about this project after a disappointing meeting at the Simons Foundation.
- 20122013: Wrote prototypes and read many books and articles about modern web development and building highly available, highly scalable distributed systems.
 - Amazed at how much web development has improved from 2008 to 2013!
 - Python is not the best tool for every programming task.
- Dec, Jan, Feb 2013: very early prototypes, which I showed some people at ICERM (Reproducible Research workshop), Sage Days in Hawaii, etc. Watched users closely, and modified design.
- March 2013: 2-week coding binge to write first version of SMC for use in teaching that quarter. Implemented differential synchronization from scratch, terminals, a completely new approach to Sage worksheets (in a single codemirror instance), etc. Weak backend architecture single points of failure. Commitment to dogfood henceforth!
- Spring 2013: *L^AT_EX* editing
- Summer 2013: Synchronized secure hosted IPython notebooks
- Nov 2013 - April 2014: A long impossible hellish deathmarch through backend storage architectures. Extremely stressful. Almost shut down the site.
- May - June 2014: Quality improvements. New functionality to get more organized: task lists.

3 Features

- All free open source math software is available, including Sage, R, Octave, Python, Cython, GAP, Macaulay2, Singular, Julia, etc.

- Collaboratively edit IPython notebooks and Sage worksheets. Realtime synchronization means that multiple people can edit the same document at once.
- Edit $LaTeX$ documents with inverse and forward search and Sage mode
- Write, compile, and run code in most programming languages
- Collaboratively use command line terminals
- Task lists to keep track of your work

15 minute demo



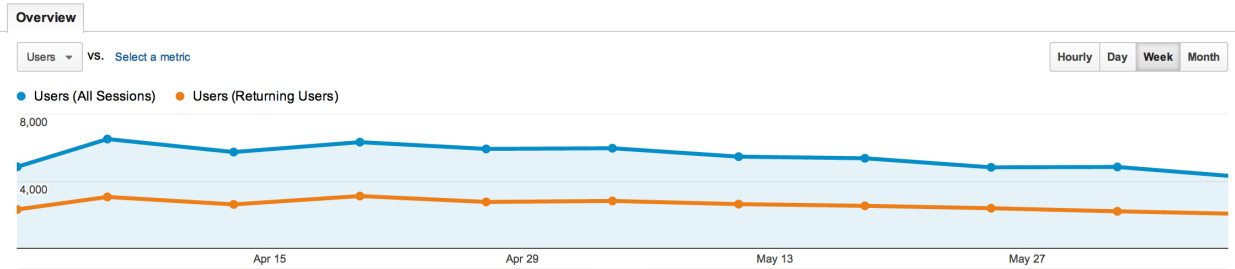
4 Users

- LongtermStats
- GoogleAnalytics:

Audience Overview

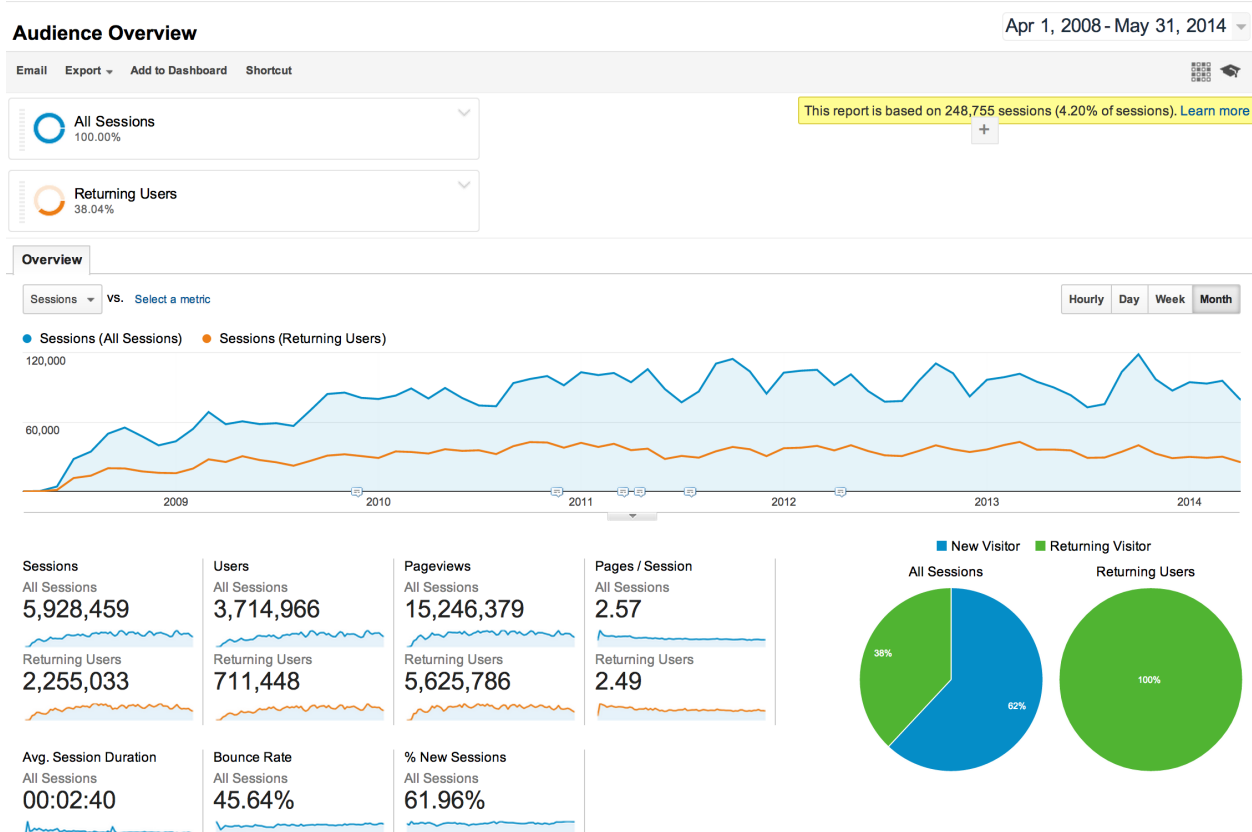
Apr 1, 2014 - Jun 14, 2014

Email Export Add to Dashboard Shortcut



5 Weekly Returning Users

- Im not happy with the growth rate.
- Number of weekly returning users is 2000-3000.
- Growth has stalled



6 Users of Sagemath.org

- I pointed out a year ago in my State of talk that 2010 was the last year in which Sage grew.
- There was zero growth in Sagemath.org users during the last year. Similar remarks about download numbers.

7 SMC Goals and Motivation

- Make open source mathematical software far more accessible:
 - Make using Sage, IPython, etc., the easiest possible choice for new users

- Make improving Sage as easy as contributing to Wikipedia (in my opinion contributing to Sage is only getting harder every year, not easier)
- Find a better funding source:
 - One million a year is the most optimistic possible funding from foundations (and massively exceeds what Sage gets)
 - Midsize company can bring orders of magnitude more revenue, and employ hundreds of people fulltime. (Wolfram Inc. has 600+ employees; Mathworks has about 2000.)

8 What is SMC?

- Hardware:
 - UW: 19 Dell R415 servers: 7 in Padelford and 12 in 4545 datacenter. Each has 16 cores, 64GB RAM. This cost \$650/month to host.
 - Google: over 30 virtual machines running Google Compute Engine, all over the world. This costs \$6000/month, but Google is making this free to us until November 2014.
- Software:
 - Database: A multi-datacenter Cassandra database (nodes in 4 different data centers)
 - Backend server: CoffeeScript + Node.js. Mostly closed source.
 - Frontend code: CoffeeScript/HTML/CSS. This will be open sourced soon.
 - Replication: Each project sits on one VM in each data center. When it is saved it gets replicated to all other data centers.
- Architectural Goals:
 - No SPOF: No single points of failure. Even when full data centers fail due to networking or other issues, you can still teach your course or do your homework. (With up to a few minutes of lost work.)
 - Collaboration: Everything should be simultaneously usable by multiple people at once.
 - Snapshots: All your files get snapshotted every few minutes, so you wont lose work.

9 This Summer

- Quality: The functionality of SMC is really, really hard to properly implement in a way that scales up to a huge number of users and has minimal bugs. Our top priority this month is overall system stability and quality.
- Course management functionality: Primary development goal for this summer.
 - A course project": linked to all student projects; provides realtime status, access, etc. to the work of all students in the course.
 - Assignment workflow:
 - * create and distribute assignments and problems
 - * automatic realtime grading
 - * collect assignments and make manual grading smoother
 - * a large crowdsourced pool of problems, with data about how they are used.
 - Simple integrated spreadsheet to keep track of students/grades.
 - Task list to keep track of assignments, and other tasks.
- Publishing: Make it possible to publish/share content from SMC.