Using RethinkDB in Production for SageMathCloud

William Stein

University of Washington

SageMath, Inc.

https://cloud.sagemath.com/

November 17, 2015

What is Sage?

SageMath

- SageMath: big open source math software I started in 2004

SageMathCloud (SMC)

- **Launched:** 2013
- **Real-time editing like Google Docs:** Latex, IPython/Jupyter notebooks, Sage, Terminals, Teaching, etc.
- **Tech Stack:** RethinkDB, Linux, React.js, Node.js, SageMath/Python, CodeMirror, CoffeeScript
- **Users:** 4000+ daily active; nearly 1000 simultaneous
- **Production:** anger when it doesn't work – "my homework is gone!"
- **Open source:** 100% open source, GPL 3, etc.

Hi From Sage Days 70

RethinkDB and SMC

Switched from Cassandra to RethinkDB this summer.

SMC Uses RethinkDB Heavily...

- **Setup:**
  - 6 Google compute engine nodes (quad-core n1-standard-4)
  - About 23 tables storing about 5 million documents
  - Replication factor 3, sharding of 3
  - Storage in persistent (network-mounted) SSD
- 5K-10K simultaneous changefeeds.

Operations

- **Backups:** periodic dump of most tables to json on a compressed filesystem, snapshot via bup (=git+more), rsync to google cloud storage and encrypted off-site USB drives.

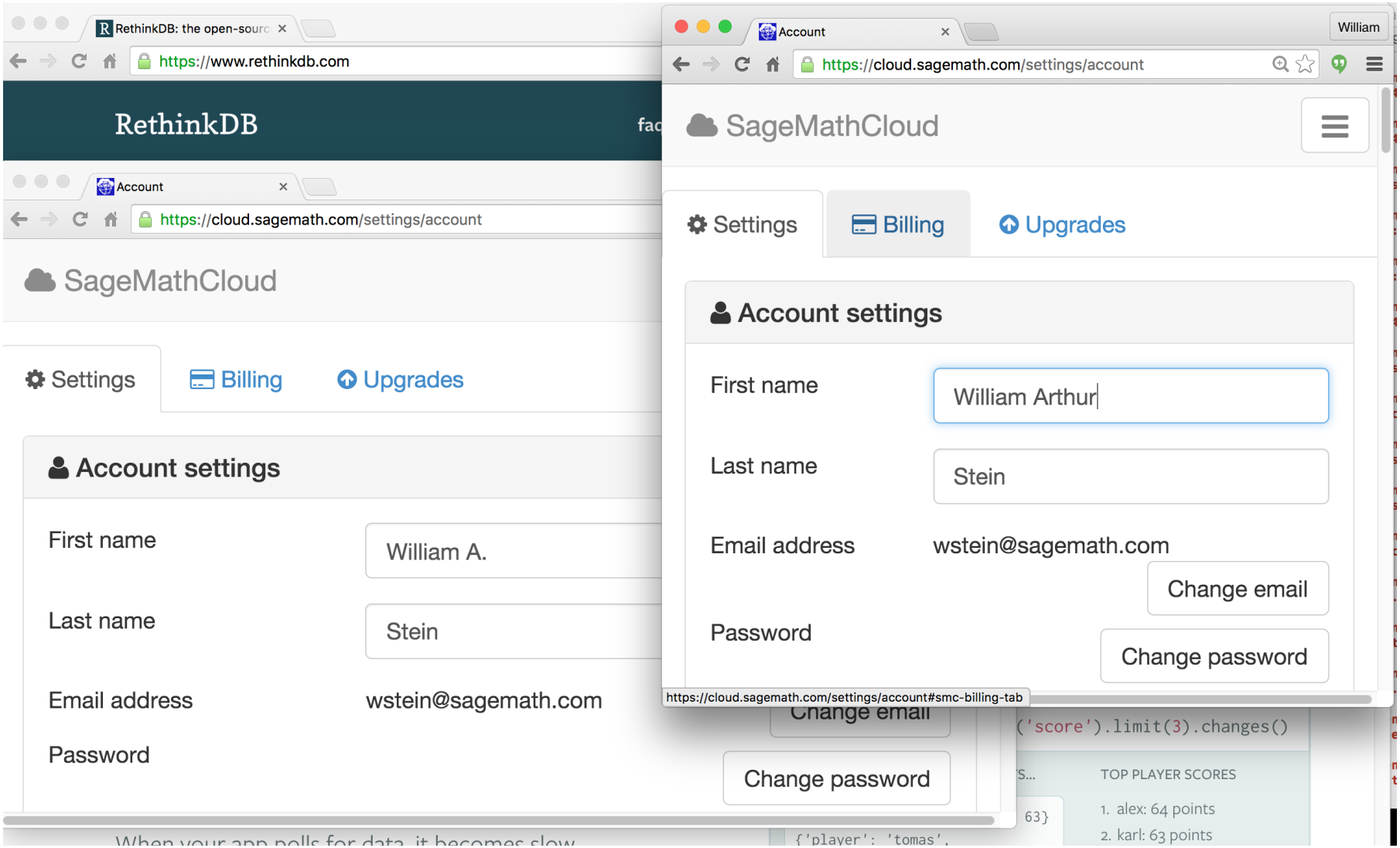
RethinkDB team **amazing** at addressing all issues I encountered.

SMC Demo

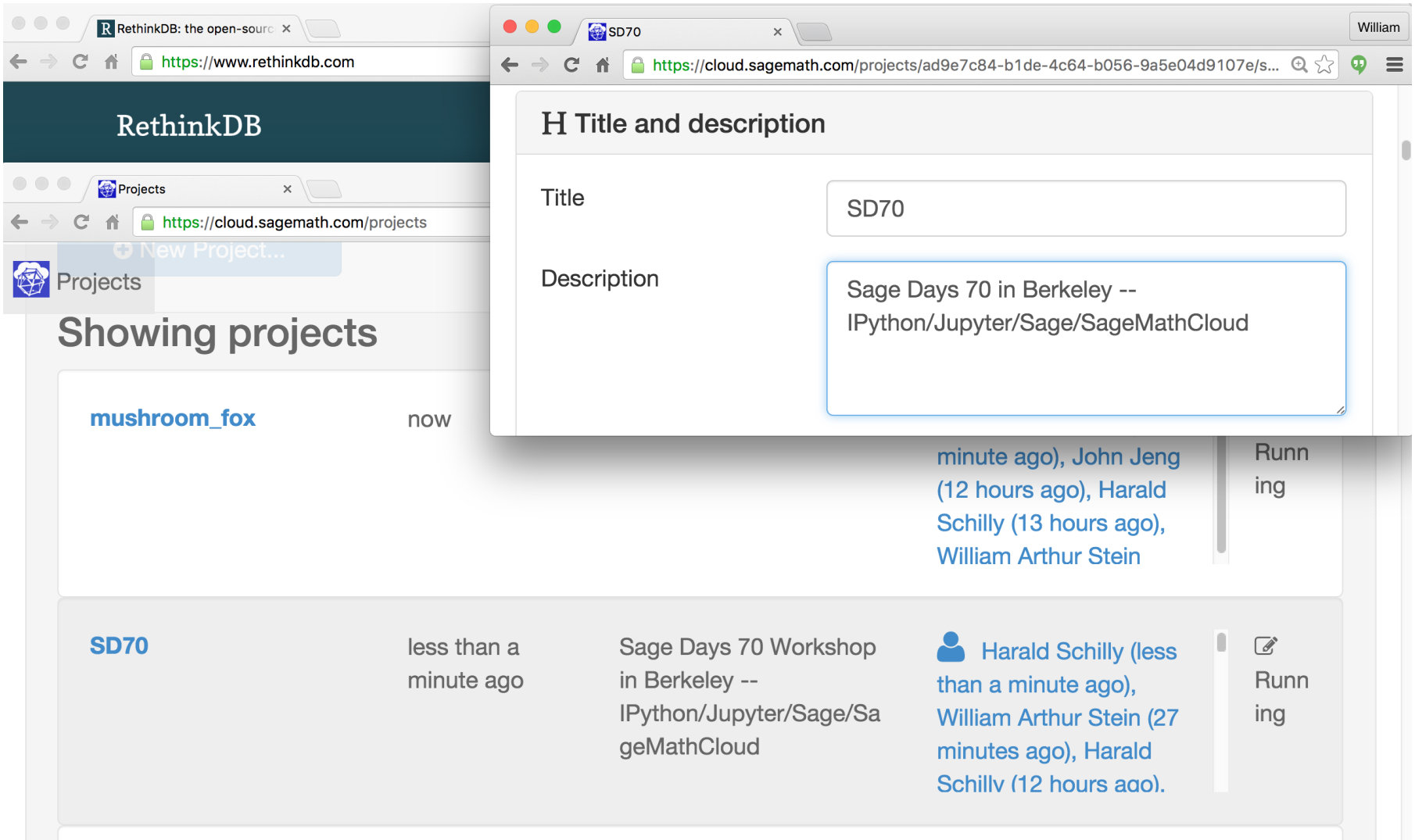
Show how SMC uses RethinkDB

1. Change name and see change in another browser.
2. Show changing project title and that appearing in another browser.
3. Draw a 3d plot in a sage worksheet
4. Open a Jupyter notebook – demo sync and history
5. No REST/API calls; instead, set entries in a table, back-end sees it, makes table change, all parts of all front-ends simultaneously see that (do a demo of project restart).

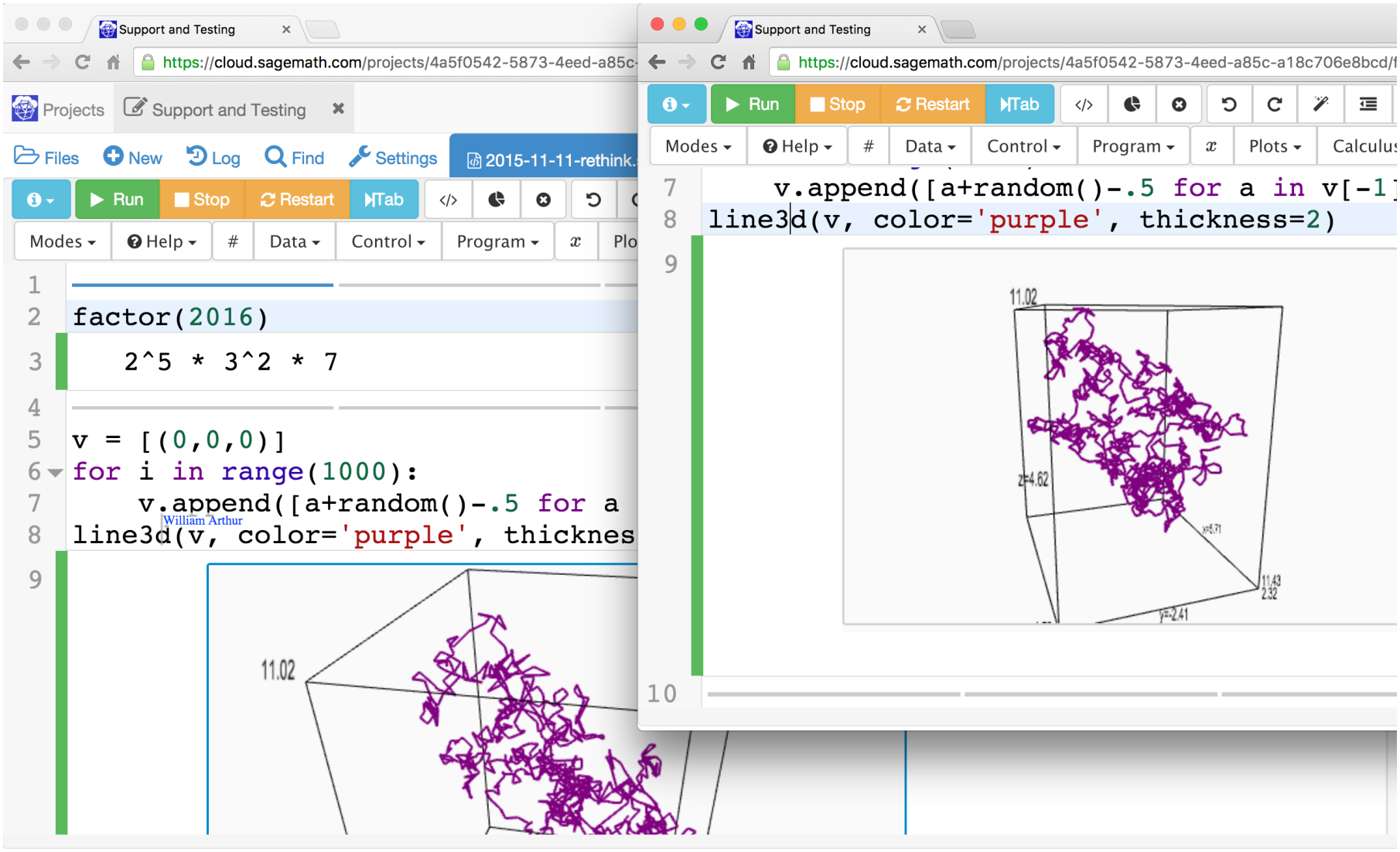
SMC Demo: Change username



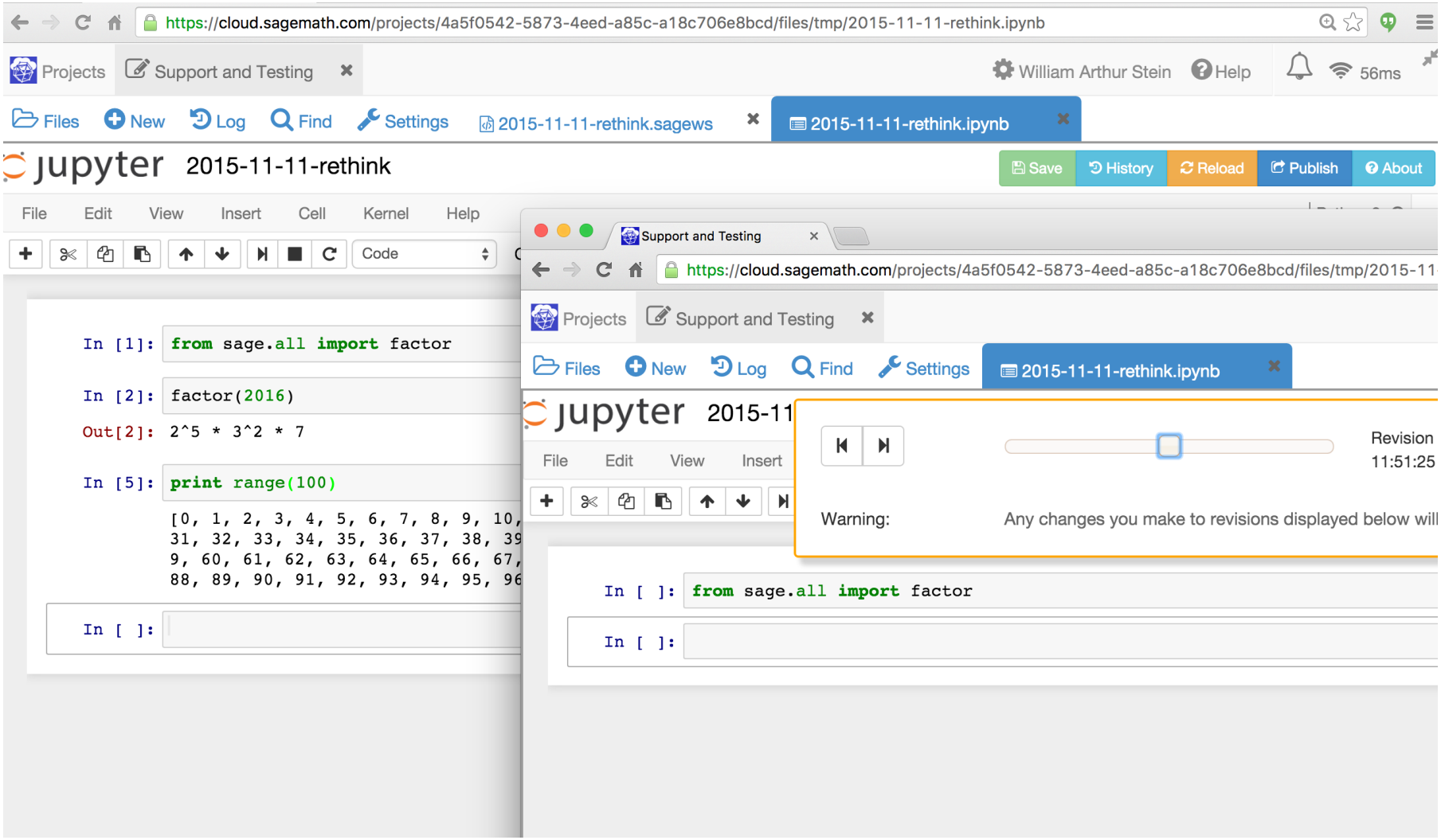
SMC Demo: Change Project Title



SMC Demo: 3D Plot



SMC Demo: Jupyter Notebook



How SageMathCloud uses Changefeeds

Motivation

- Make front-end development easier
- Simplify code connecting the front-end to back-end (one declaration instead of messages flying all over)

Inspiration

- Facebook's GraphQL – but simpler

Goal

- Have declarative client-side queries and database schema
- Instant notifications about changes.

Building a GraphQL-like API on RethinkDB and Node.js

(do not look at this)

Browser (or iOS/Android at some point) client query:

- JSON object that describes what result should look like; null's get filled in. `{table:{foo:bar, stuff:null}}` gets one record in table where `foo="bar"` and `{table:[{foo:bar, stuff:null}]}` gets them all.
- If `changes=true`, then any time RethinkDB table changes, client gets updates, and anytime client makes changes, they get pushed to back-end to RethinkDB.
- Tables can be "virtual", and not correspond to actual RethinkDB tables. e.g., different permissions, or involving multiple tables (so joins, technically; they also have a killfeed).
- Show `schema.coffee`.
- Text editing: describe algorithm based on the above, which isn't deployed yet.

Instrumentation data in production

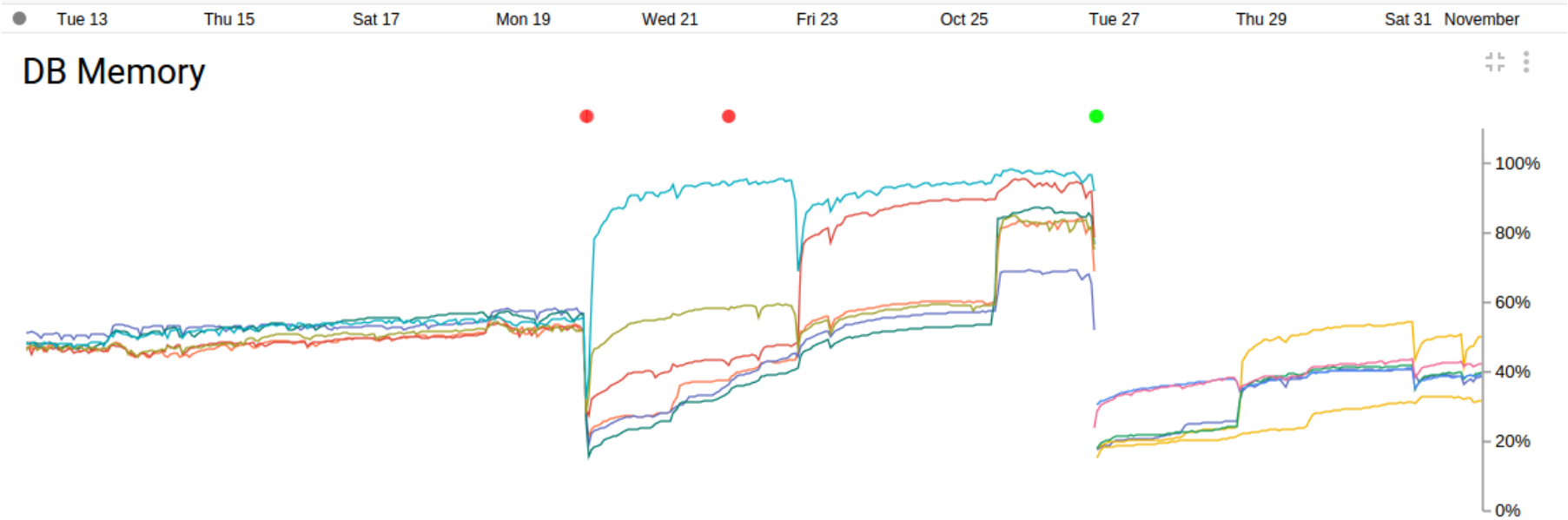
Example: Server Overload

About 3 week of data for November 2015 across 6 nodes. At one point (with 6 n1-highcpu-2s), we hit a threshold (with around 850 simultaneous users) and the backend collapsed.

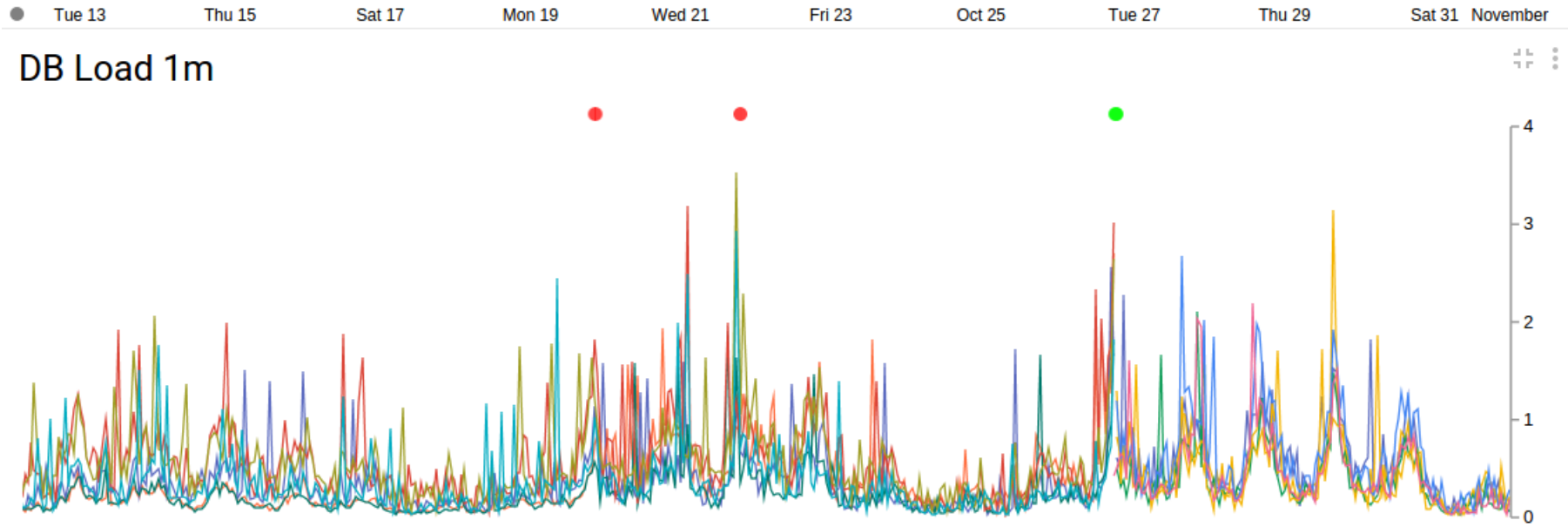
Solution

A new node had to be added (Tue 27th).

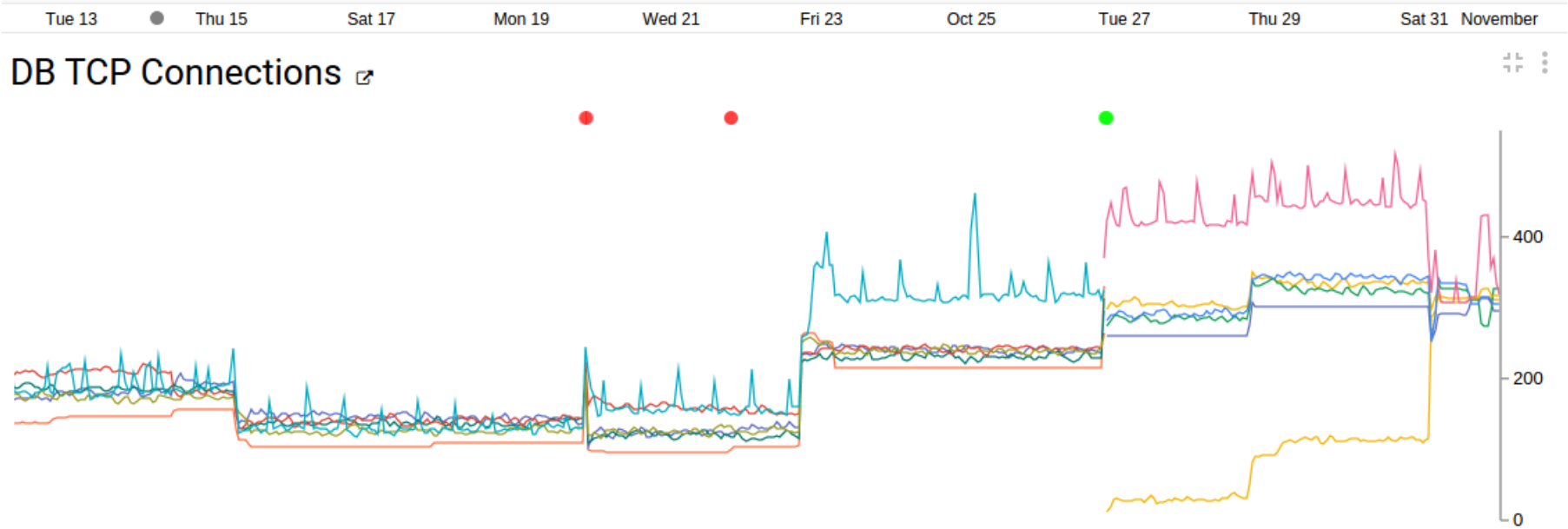
Memory usage across database nodes



CPU Load (1 min) across database nodes



TCP connections across database nodes



Thanks!

Sign up today!

https://cloud.sagemath.com/