Dealing with Contributor overload

Holden Karau

@holdenkarau

Holden:

- My name is Holden Karau
- Prefered pronouns are she/her
- Developer Advocate at Google
- Apache Spark PMC, Beam contributor
- previously IBM, Alpine, Databricks, Google, Foursquare & Amazon
- co-author of Learning Spark & High Performance Spark
- Twitter: @holdenkarau
- Slide share http://www.slideshare.net/hkarau
- Code review livestreams: https://www.twitch.tv/holdenkarau / https://www.youtube.com/user/holdenkarau
- Spark Talk Videos http://bit.ly/holdenSparkVideos

Who is Boo?

@booprogrammer
Drawn by @impurepics

- Boo uses she/her pronouns (as I told the Texas house committee)
- Best doge
- Lot’s of experience barking at computers to make them go faster
- Author of “Learning to Bark” & “High Performance Barking”
- On twitter @BooProgrammer

Who do I think you all are?

Amanda

- Nice people*
- Care about open source
- Possibly work on a project bigger that got bigger than you thought
- Maybe overwhelmed with PRs/CRs/diffs & e-mails/DMs

Remember it's ok not to fix it all

jon jordan

- Many of us have a backlog of change requests to review
- Many of us have lots of messages we can’t answer
- Many of us wish we had more time to mentor folks
- This is normal and ok

What are we going to talk about?

Mark Ittleman

- What changes < 10 people to > 1k people is different
- It’s not your fault if you feel overwhelmed
- All these wonderful people want to help, but its still work
- Community structures (BDFL, ASF, etc.)
- Partial technical attempts at solving social problems
- Biased towards the problems of developers, etc.

Also:

- If I’m behind reviewing your PRs I’m sorry

Or another way of phrasing this:

- Is it on fire?
- Why is it on fire?
- Accepting the it is on fire and the marshmallows, are good, but maybe we should do something about this
- Oh wait the block is one fire
- What can we do to have it not all be on fire?
- Eh, it’s sort of smoldering but that’s ok

The fun of a small project:

flowbadger

- Simpler communication
- Generally more aligned goals (folks agree on project direction)
- It's easy to tell who knows what
- Tight knit community, easier to convert users to contributors
- Easier to keep track of users & contributors
- BDFL or consensus “just works”*

The fun of a large project:

- More people doing the work
- More impact/people thanking y'all
  - Even if your system is mostly used to sell ads, you can probably find someone doing some good with it and tell yourself that's good.
- Lots of ideas* & experience
- If theres $s you can have a pretty fun conference
- Easier getting folks (including self) paid to work on it

*Hopefully different ideas & experience. Ask yourself if your project is well diversified. If you can't answer that question maybe try and measure it.

So… has is it on fire?

Maybe it’s just bright….

Photo by Martin Snicer Photography in
Woy Woy Bay, Australia

So what changes?

Easily Measurable:

- User questions spike*
- Issue creation spike**
- Code change requests spike***
- More people everywhere

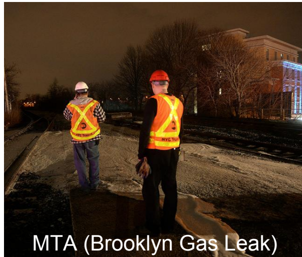
Less Measurable but also important:

- Unwritten assumptions are lost/not passed on
- Project goals could create silos and divide your community
- At some point your project will get a little more diverse*
- People will use your software to do unexpected things

Ok maybe it is on fire, but why?

Remember that classic pipeline?

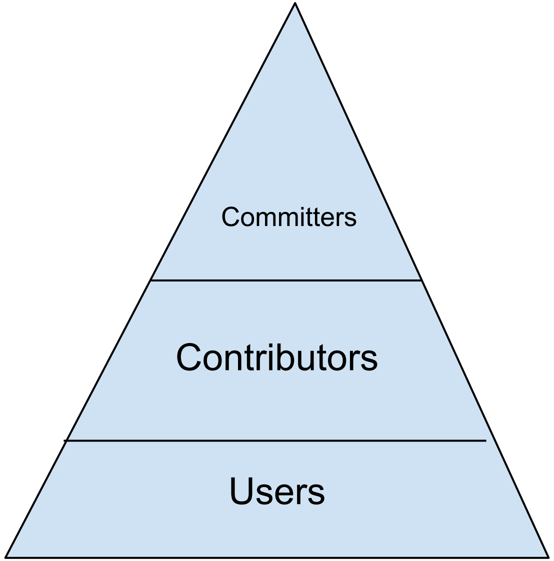
- No, not meltdown & specter
- Users -> Contributors -> Committers -> PMC
- Each stage takes time
    - What happens with a sudden influx? ruh-roh
- And most of us leaky pipelines
- And then add burnout...


MTA (Brooklyn Gas Leak)

At the start the distribution is like:

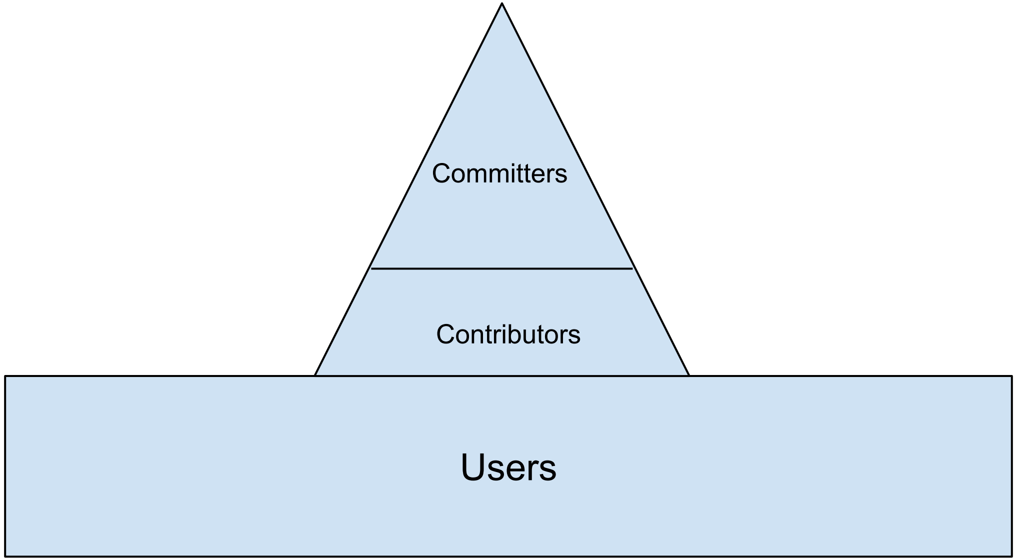
Committers
(aka you)

With slow growth the pipeline can keep up*



Kenta Hayashi

But hyper growth, can quickly go sour

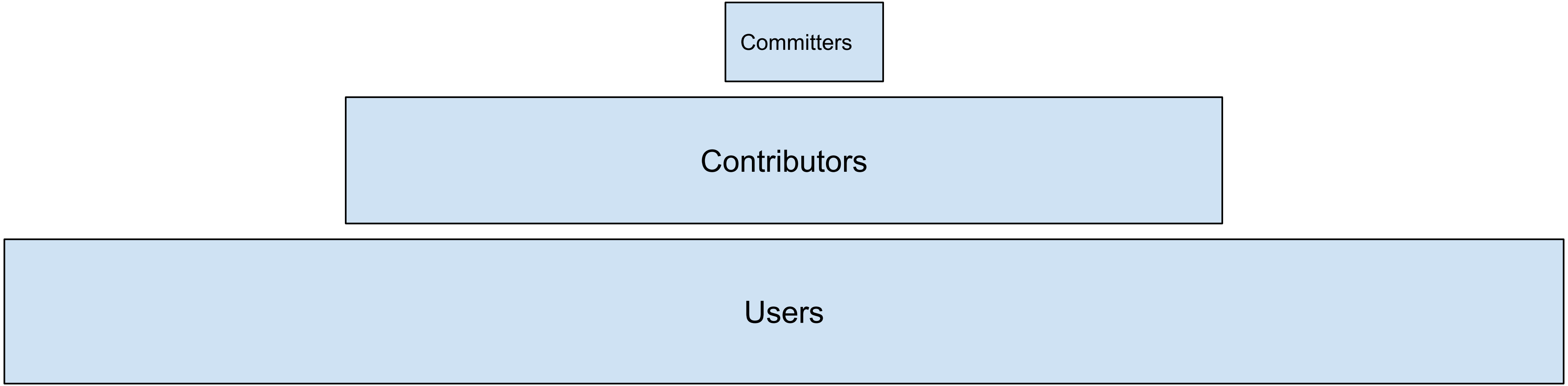


Committers

Contributors

Users

But hyper growth, can quickly go sour

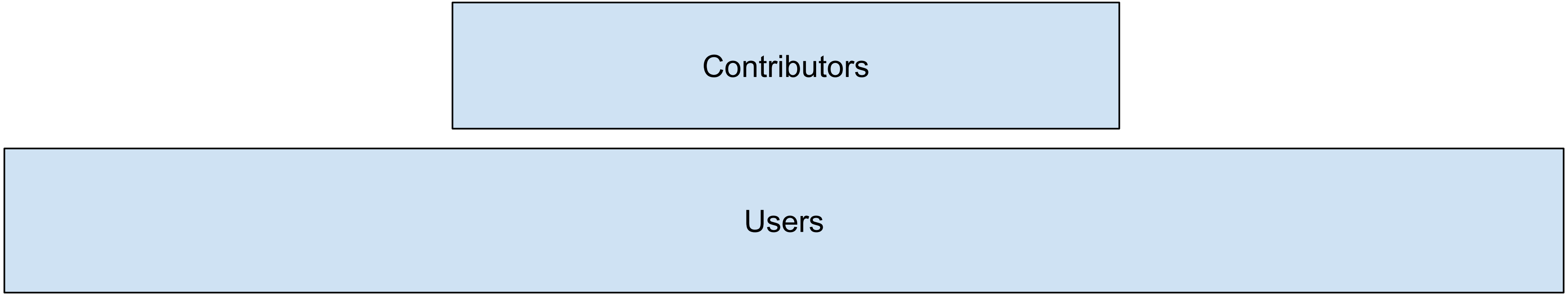


Committers

Contributors

Users

Add burn out and….



Contributors

Users

Add burn out and….

Sad Users

Even without hyper growth: sadness

Chris Luczkow

- It can be like boiling a frog (please don’t)
- If your user -> contributor pipeline stalls
  - question overload
- If your contributor -> committer pipeline stalls
  - Oh god too many prs! The goggles do nothing
- If committer -> “management” pipeline stalls
  - Who should we make a committer? idk? (then see above)

- Insert joke about CPU pipelining here (see news)

Ok the marshmallows are delicious

But, maybe we need to do something about this fire

Question overload? - Mitigation

Johnny Jet

- You don’t have to answer everyone. This can be hard.
- Stackoverflow - it isn’t perfect but it have interesting gamification
- Are your answers easily searchable?
  - Can you summarize FAQs to the docs
- Ask for help (think like peer support for Spark users :p)
- Filters - some questions aren’t worth answering or will be answered by others

spark (22,331)

system ml commits (… ▼

Question overload? - Longer term

pjmorse

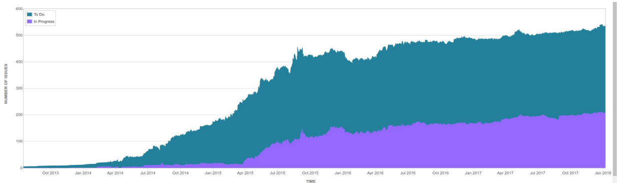
- Improve discovery for existing answers
  - Knowledge base + “did you mean” (often in corporate idk OSS)?
  - Or even just indexed list (eg. google groups) rather than IRC/Slack
- Take time and look for patterns:
  - Do you error messages make sense? Are there common ones you can improve. Potentially great starter issues!
  - Maybe old version has common bug? Add a bot to ask to upgrade?*
- Find people who like training/teaching/writing: trick them
  - Writing has terrible direct $s ROI, but maybe ok indirect ROI
  - Don’t do this yourself if you’re already overloaded (do review)

Issue overload?

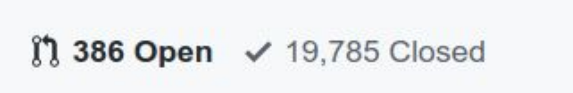
Again:

- You don’t have to answer everyone. This can be hard.

- Seeing lots of duplicates?
  - Similar questions as with user list, also consider find duplicates tooling
- Seeing lots of “help me”? Is issue form easier than question?
- Can you make it easier for folks to fix own issues?
- Lots of things that can’t/won’t be fixed: auto close?



Oh wait that’s more...



386 Open ✓ 19,785 Closed

vins Strauhmanis

How do people react to contributor spike?

Generally a mix 3 broad approaches:

- “Raise the bar” (e.g. make it harder for contributors)
  - e.g. We have too many pull requests, add more restraints or hoops
  - See some of the discussions around the Go review system
- Make it easier for contributors
  - e.g. We have too many pull requests, let make this simpler
- I have a social problem I bet perl* can solve this

*For modern folks replace perl with Python or the language of the day (one day it will be Perl again…)

What does “raising the bar” look like

phphoto2010

- Rejecting small changes as “trivial” or “irrelevant”
- Picking non-standard systems to make it harder**

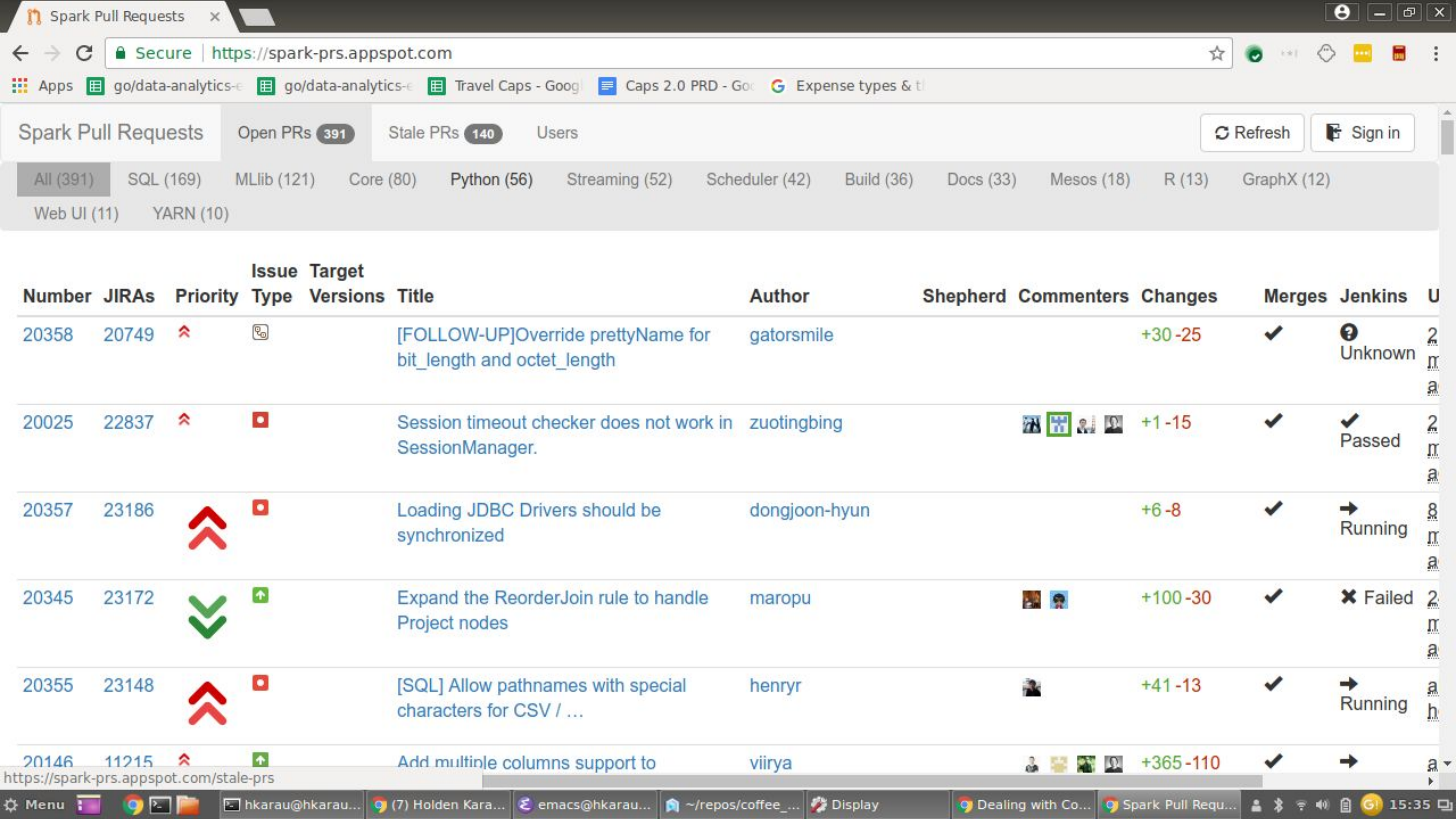
Downside:

- Reduces the accessibility of the on-ramp for contributors
- Only contributors willing to jump through hoops remain
- More posts/questions on how to contribute
- Money makes this not work: if it’s my job I like food.
- Worsens issue/question overload

What can we solve with perl*?

- Bots, bots, bots!
- Decide on a style guide & automate linting, run in CI
  - Stop arguing about spacing
- Make it faster to merge (e.g. close issue automatically)
- Improve PR + reviewer notice (e.g. mention-bot or spark-pr-dashboard, broken down by area)

*Fine Python.



Spark Pull Requests ×

Secure | https://spark-prs.appspot.com

Apps | go/data-analytics-e | go/data-analytics-e | Travel Caps - Googl | Caps 2.0 PRD - Go | Expense types & t

Spark Pull Requests | Open PRs 391 | Stale PRs 140 | Users | Refresh | Sign in

All (391) | SQL (169) | MLlib (121) | Core (80) | Python (56) | Streaming (52) | Scheduler (42) | Build (36) | Docs (33) | Mesos (18) | R (13) | GraphX (12) | Web UI (11) | YARN (10)

Number	JIRAs	Priority	Issue Type	Target Versions	Title	Author	Shepherd	Commenters	Changes	Merges	Jenkins	U
20358	20749	⌃	⧉		[FOLLOW-UP]Override prettyName for bit_length and octet_length	gatorsmile			+30 -25	✓	? Unknown	2 m a
20025	22837	⌃	🟥		Session timeout checker does not work in SessionManager.	zuotingbing			+1 -15	✓	✓ Passed	2 m a
20357	23186	⇈	🟥		Loading JDBC Drivers should be synchronized	dongjoon-hyun			+6 -8	✓	→ Running	8 m a
20345	23172	⇊	⬆		Expand the ReorderJoin rule to handle Project nodes	maropu			+100 -30	✓	✖ Failed	2 m a
20355	23148	⇈	🟥		[SQL] Allow pathnames with special characters for CSV / …	henryr			+41 -13	✓	→ Running	a h
20146	11215	⌃	⬆		Add multiple columns support to	viirya			+365 -110	✓	→	a

https://spark-prs.appspot.com/stale-prs

Menu | hkarau@hkarau... | (7) Holden Kara... | emacs@hkarau... | ~/repos/coffee_... | Display | Dealing with Co... | Spark Pull Requ... | 15:35

Community > Code*

5chw4r7z

- If we had bots that could handle the review we could just code this away, but we don’t
- Scaling your existing reviewers can only go so far, you probably need to look at community fixes.

*One of the ASF slogans

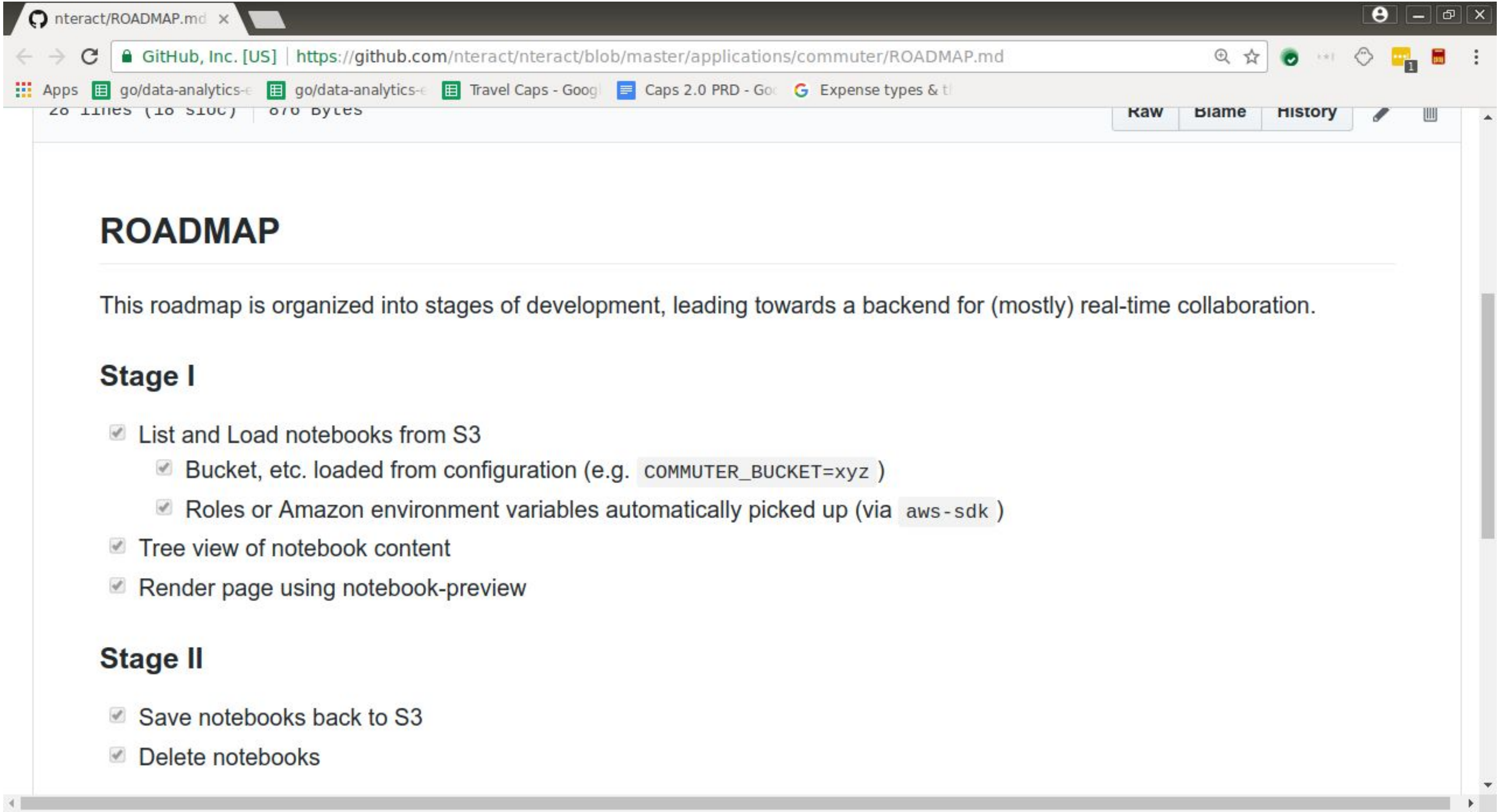
Decide what you *aren't* doing*

- Communicate what's in+out of scope clearly
  - Have these conversations in public so it isn't super arbitrary
  - Even if it's cool and related doesn't mean you have the bandwidth
  - Learn to say no nicely**, this can be hard. #dreamcrusher
- Consider extensibility -- let work live outside
- Project splitting - maaaaybe you don't need keep doing everything

e.x.:

- In Spark we added ML roadmaps + made pluggable
  - On the other hand we aren't so good at closing issues or PRs
- Spark+K8 support: not fully pluggable had 3+ forks :(
- Nteract keeps roadmaps in repo

*For now. It can change.



ROADMAP

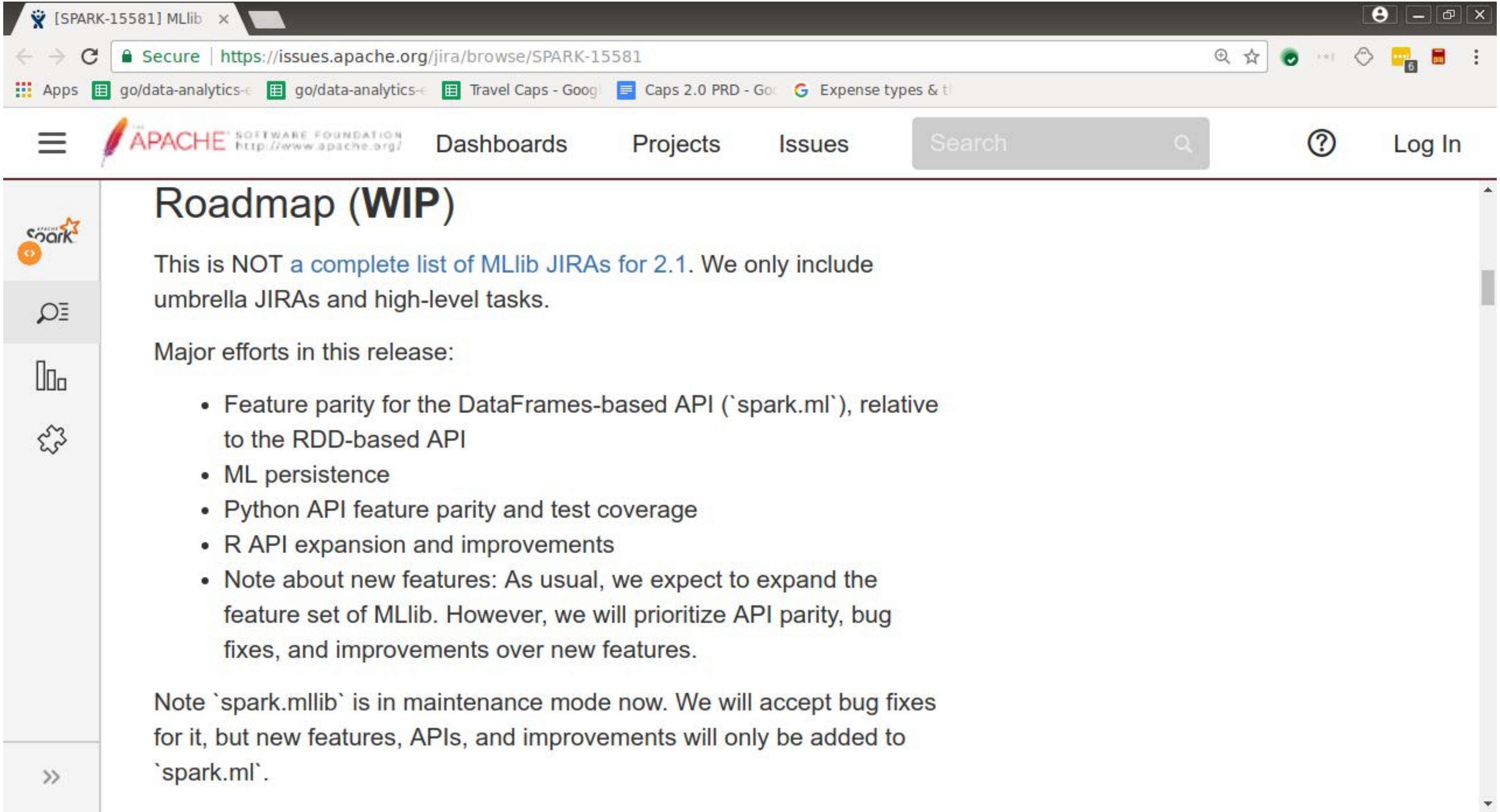
This roadmap is organized into stages of development, leading towards a backend for (mostly) real-time collaboration.

Stage I

- ☑ List and Load notebooks from S3
  - ☑ Bucket, etc. loaded from configuration (e.g. `COMMUTER_BUCKET=xyz`)
  - ☑ Roles or Amazon environment variables automatically picked up (via `aws-sdk`)
- ☑ Tree view of notebook content
- ☑ Render page using notebook-preview

Stage II

- ☑ Save notebooks back to S3
- ☑ Delete notebooks



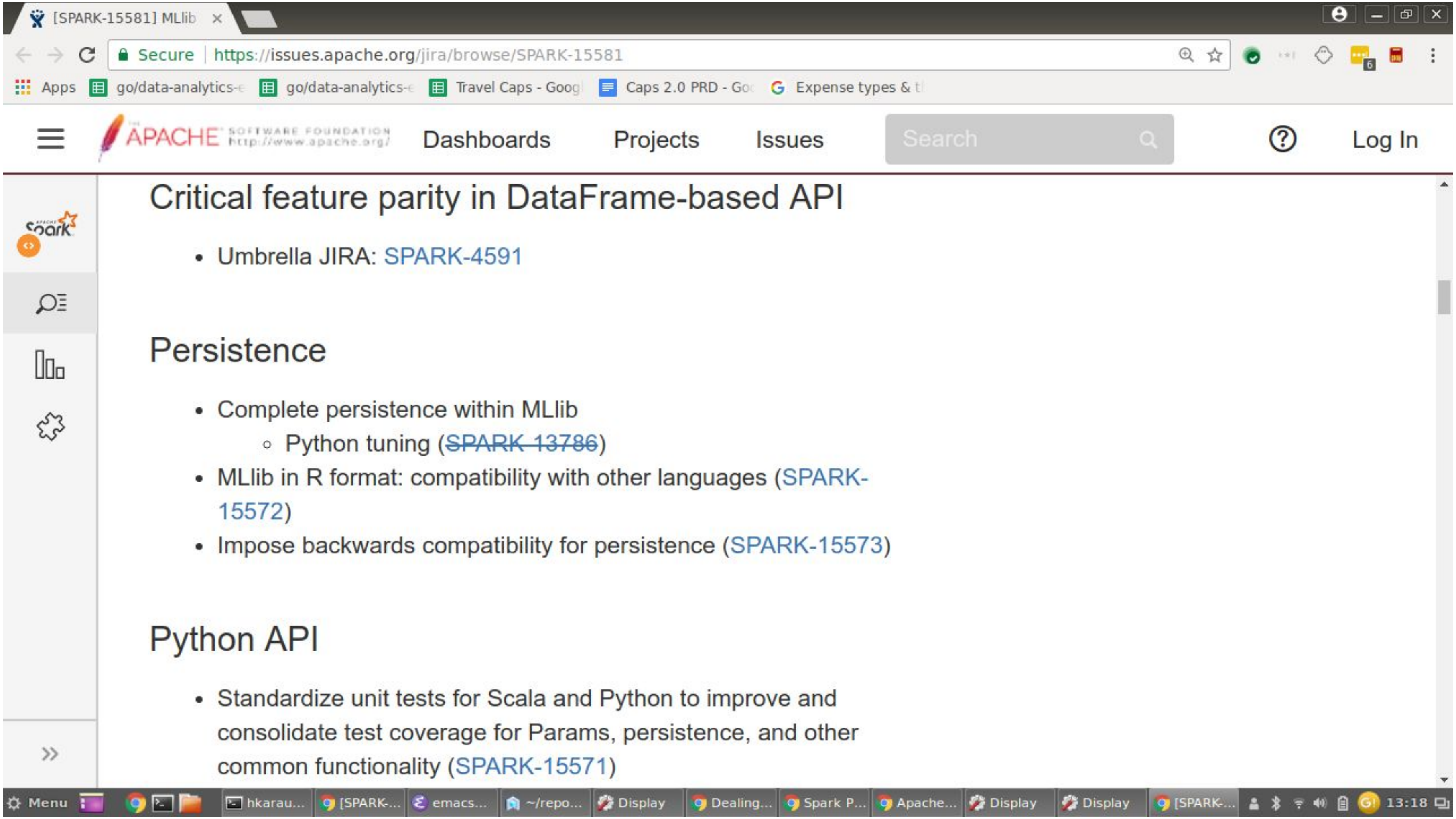
Roadmap (**WIP**)

This is NOT a complete list of MLlib JIRAs for 2.1. We only include umbrella JIRAs and high-level tasks.

Major efforts in this release:

- Feature parity for the DataFrames-based API (`spark.ml`), relative to the RDD-based API
- ML persistence
- Python API feature parity and test coverage
- R API expansion and improvements
- Note about new features: As usual, we expect to expand the feature set of MLlib. However, we will prioritize API parity, bug fixes, and improvements over new features.

Note `spark.mllib` is in maintenance mode now. We will accept bug fixes for it, but new features, APIs, and improvements will only be added to `spark.ml`.



Critical feature parity in DataFrame-based API

- Umbrella JIRA: SPARK-4591

Persistence

- Complete persistence within MLlib
  - Python tuning (~~SPARK-13786~~)
- MLlib in R format: compatibility with other languages (SPARK-15572)
- Impose backwards compatibility for persistence (SPARK-15573)

Python API

- Standardize unit tests for Scala and Python to improve and consolidate test coverage for Params, persistence, and other common functionality (SPARK-15571)

Increase committer productivity

- Better tools to merge changes
- Easier to review changes
- More tests to reduce worry about changes
    - Tests make it less scary/easier to trust new changes
- Make it *ok* to break things and fix in master
    - Everything could break something,
- Better tools for reviews

*In the same time, no just buying them red-bull.

Add more committers:

- Encourage people to be interested in being committers
    - Share decision making (e.g. if it didn’t happen on the list…)
- Evaluate how you pick committers
    - Consensus voting is a great way to eat at cheesecake factory
    - Are your standards unreasonable? Encourage folks to be really clear with -1s
- Make guides for committers
- Encourage more issues for new committers to handle
- Find people who are almost ready and mentor them
    - Encourage the high volume contributors to help out with review
        - Review mention bots can help here
    - Help them on the review
        - Livestream reviews can help people feel comfortable reviewing.
- Make it shiny

Mentoring folks to become committers

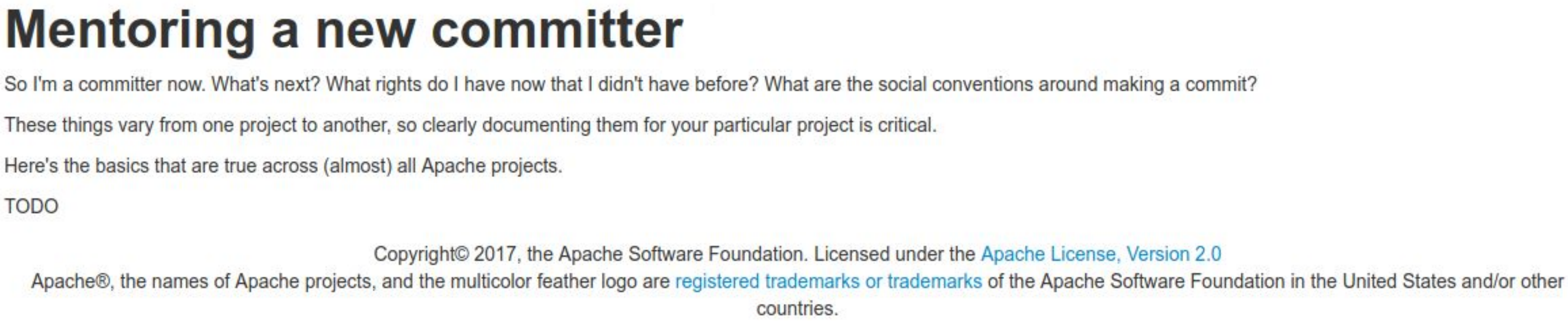
rsaxvc

- Encourage contributors to review each others code
  - Hey I saw your working and X and its related to Y, can y'all sync?
- Teaching people to review OSS PRs well
  - Very different skill with OSS than internal.
  - Much less shared background, we probably didn't all study at UTS
  - Encourage partial reviews / asking for second opinions
- Allow specialization to start, and optional growth
  - If you require folks to know everything before they start they might walk away
- Hard to scale, and hard to find people excited about.

And then onto: effective committers

- Encourage people to feel safe making changes
  - New committers can often be gun-shy
  - Follow the new committers and reach out if they don’t merge anything
- Make a guide for new folks to follow
  - It can help to point out what can be fixed easily and what can’t
  - We forgot SCM makes a lot of things less scary
- If you can make a safe space to ask questions
  - That nagging feeling “maybe they’ll realise how little I know and take away my bit” can be hard to overcome
  - Mailing lists can be scary (especially if it’s the same one used for serious business)

But a lot of this is about your project...



Mentoring a new committer

So I'm a committer now. What's next? What rights do I have now that I didn't have before? What are the social conventions around making a commit?

These things vary from one project to another, so clearly documenting them for your particular project is critical.

Here's the basics that are true across (almost) all Apache projects.

TODO

Copyright© 2017, the Apache Software Foundation. Licensed under the Apache License, Version 2.0
Apache®, the names of Apache projects, and the multicolor feather logo are registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Making it easier to contribute

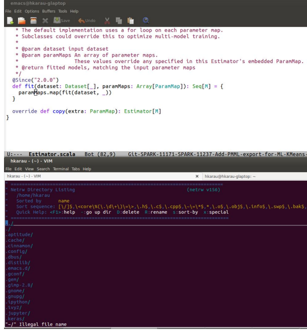
- Semi-counter intuitive solution to too many contributions
- Adds more simple small changes, mini-victory
- Introduce a half-step: empower new committers for simple issues
    - Helps move the pipeline along faster
- Gives more people to move forward
- Try to push people (initially) towards contributing in areas that easier for you to handle

Reduce overhead to contribute *well*

Tambako The Jaguar

- Set expectations clearly (CONTRIBUTING.md)
- Is your style guide easy to find? It should be
- PR templates (if possible)
- Tooling (send-pr, linters, etc.) -- ideally integrate in the process **before** committers see the changes

There are more decisions & they get harder to make



```
   * The default implementation uses a for loop on each parameter map.
   * Subclasses could override this to optimize multi-model training.
   *
   * @param dataset input dataset
   * @param paramMaps An array of parameter maps.
   *                  These values override any specified in this Estimator's embedded ParamMap.
   * @return fitted models, matching the input parameter maps
   */
  @Since("2.0.0")
  def fit(dataset: Dataset[_], paramMaps: Array[ParamMap]): Seq[M] = {
    paramMaps.map(fit(dataset, _))
  }

  override def copy(extra: ParamMap): Estimator[M]
}

U:---  Estimator.scala   Bot (82,9)    Git-SPARK-11171-SPARK-11237-Add-PMML-export-for-ML-KMeans-
```

```
" ============================================================================
" Netrw Directory Listing                                        (netrw v156)
"   /home/hkarau
"   Sorted by      name
"   Sort sequence: [\/]$,\<core\%(\.\d\+\)\=\>,\.h$,\.c$,\.cpp$,\~\=\*$,*,\.o$,\.obj$,\.info$,\.swp$,\.bak$,
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:special
" ==============================================================================
../
./
.aptitude/
.cache/
.cinnamon/
.config/
.dbus/
.distlib/
.emacs.d/
.gconf/
.gem/
.gimp-2.8/
.gnome/
.gnupg/
.ipython/
.ivy2/
.jupyter/
.keras/
"~/" Illegal file name
```

- Consensus voting
  - Consensus* voting can stall
  - A way for companies that have different interest to work together
  - Style used by ASF (additional requirements)
  - Feeling of loss of control
- BDFL: Scale difficulty, may upset more folks
  - Even then delegation is necessary
  - May be more likely to fork?
- Democracy
  - Feeling of loss of control

Primarily technical things explode too

Cats by moonwhiskers

- Codebases tend to get bigger
- Build times get longer
- More and more tests get added for the bug the community finds
- The technical problems can feel simple in comparison

Technical problems -> technical fixes

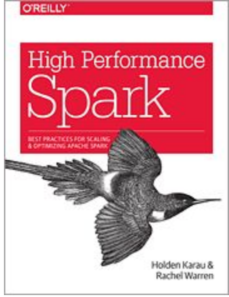
- Componentize/refactor
  - Faster build times
  - Allows people develop separately to some degree
  - Also makes it easier if you decide to split up into separate projects
- Parallelize testing
- Quick tests + validation/integration testing
- More hosted testing/validation infra
  - This can be rough, trusting folks with existing infra is scary
  - Consider parallel testing infra where you let the “new kids” work

Wrapping up: It’s ok not to be perfect

紫流

- Many of us have a backlog of change requests to review
  - My areas of Spark have > 100 open PRs, many of which I’ll never touch
- Many of us have lots of messages we can’t answer
- Many of us wish we had more time to mentor folks
- It’s ok to only do some, it doesn’t mean you’re bad at this.

I also have a book...


O'REILLY
High Performance
Spark
BEST PRACTICES FOR SCALING & OPTIMIZING APACHE SPARK
Holden Karau & Rachel Warren

High Performance Spark, it’s available today & the gift of the season.

Unrelated to this talk, but if you have a corporate credit card (and or care about distributed systems)….

http://bit.ly/hkHighPerfSpark

And some upcoming talks:

- June
  - Live streams (today & tomorrow) - follow me on twitch & YouTube
  - Office hours tomorrow ~ 1630 to 1730 somewhere close by
  - Scala Days NYC - Missed out on Scala Days EU? Come to NYC!
- July
  - Possible PyData Meetup in Amsterdam (tentative)
  - Curry on Amsterdam
  - OSCON Portland
- August
  - JupyterCon NYC
- September
  - Strata NYC
  - Strangeloop STL

k thnx bye!

If you <3 Spark testing & want to fill out survey: http://bit.ly/holdenTestingSpark

Want to e-mail me? Promise not to be creepy? Ok: holden@pigscanfly.ca

Want to tell me (and or my boss) how I’m doing? http://bit.ly/holdenTalkFeedback

Cat wave photo by Quinn Dombrowski