BigPanDA monitor

Sergey Padolski
BNL

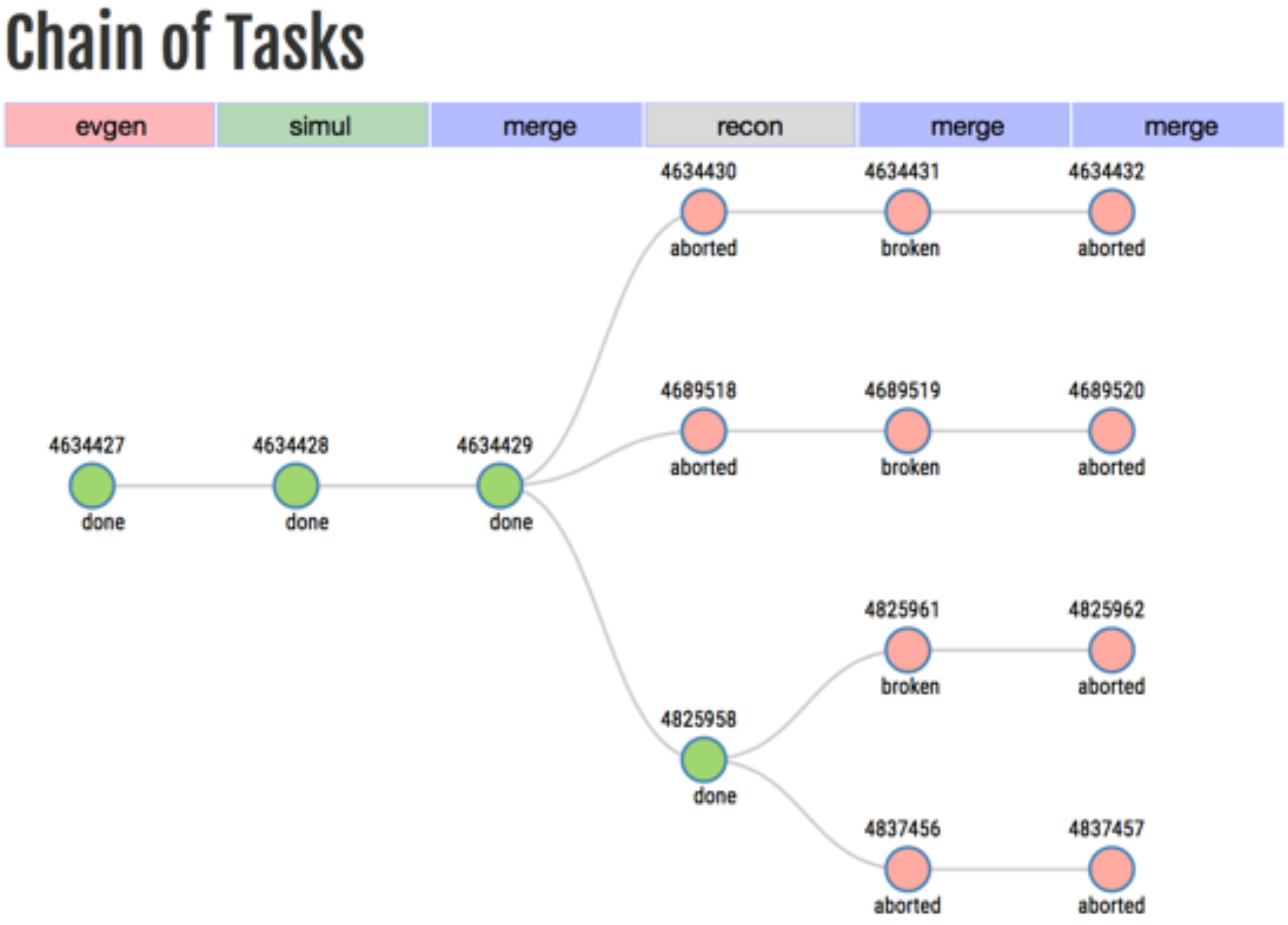
Status

- Recently: More data, more plots, more pages, more fixes

- Feedback: monitor is great, but performance is a problem. + More links to offline analytics is needed.

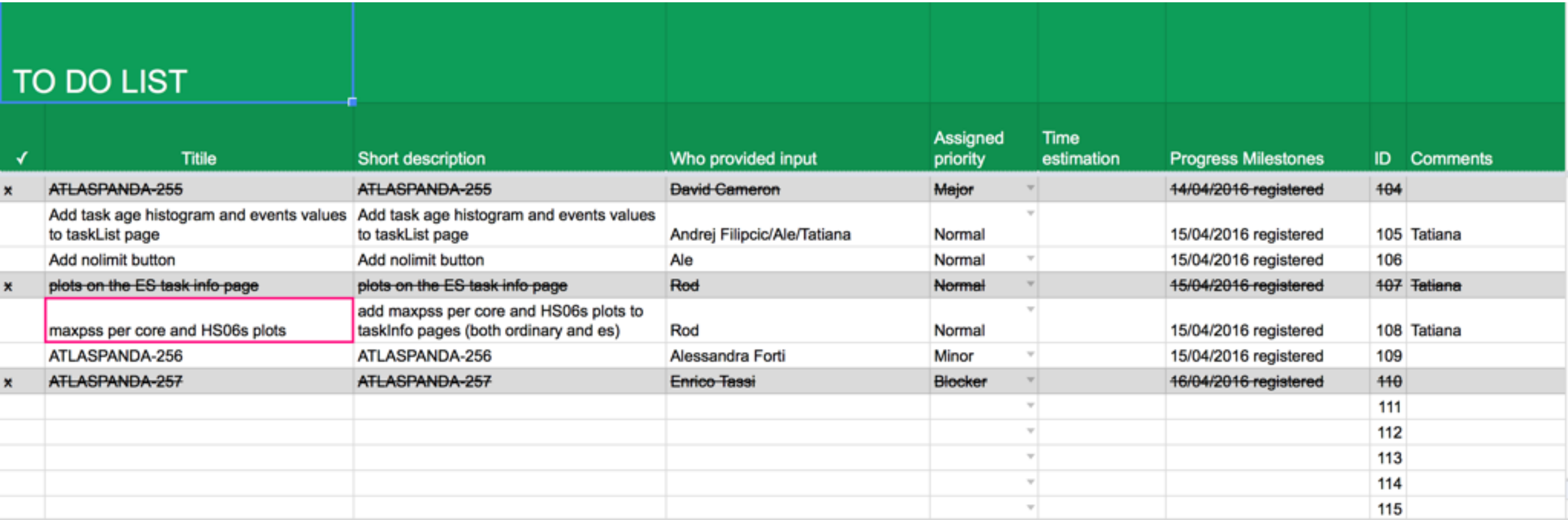
- Near future: performance reworking, numbers revision.

Status



New feature from Kurchatov Institute

Task List

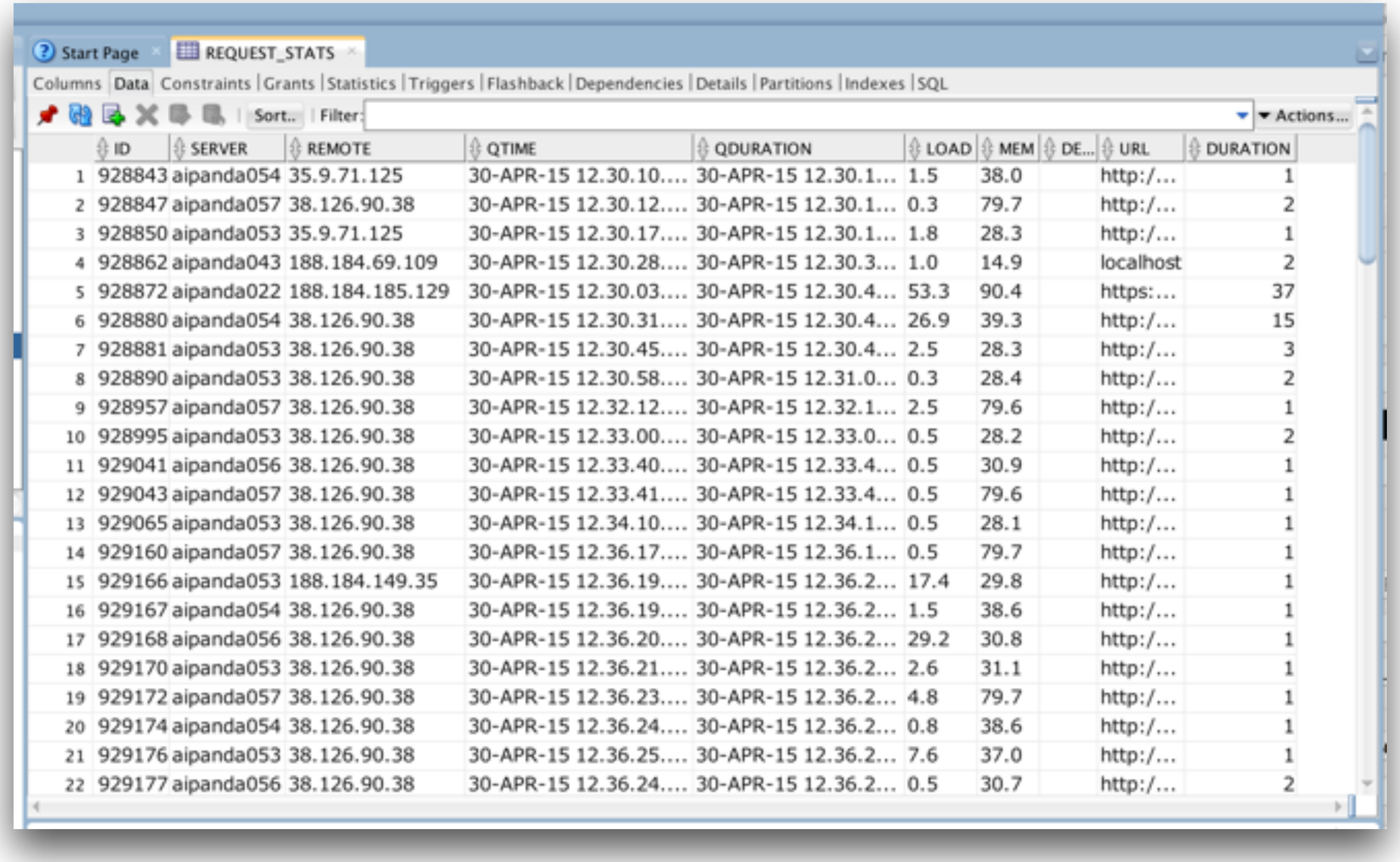


TO DO LIST

✓	Titile	Short description	Who provided input	Assigned priority	Time estimation	Progress Milestones	ID	Comments
×	~~ATLASPANDA-255~~	~~ATLASPANDA-255~~	~~David Cameron~~	~~Major~~		~~14/04/2016 registered~~	~~104~~	
	Add task age histogram and events values to taskList page	Add task age histogram and events values to taskList page	Andrej Filipcic/Ale/Tatiana	Normal		15/04/2016 registered	105	Tatiana
	Add nolimit button	Add nolimit button	Ale	Normal		15/04/2016 registered	106	
×	~~plots on the ES task info page~~	~~plots on the ES task info page~~	~~Rod~~	~~Normal~~		~~15/04/2016 registered~~	~~107~~	~~Tatiana~~
	maxpss per core and HS06s plots	add maxpss per core and HS06s plots to taskInfo pages (both ordinary and es)	Rod	Normal		15/04/2016 registered	108	Tatiana
	ATLASPANDA-256	ATLASPANDA-256	Alessandra Forti	Minor		15/04/2016 registered	109	
×	~~ATLASPANDA-257~~	~~ATLASPANDA-257~~	~~Enrico Tassi~~	~~Blocker~~		~~16/04/2016 registered~~	~~110~~	
							111	
							112	
							113	
							114	
							115	

https://docs.google.com/spreadsheets/d/1cSz2j4iI-zlapLc8Ap9uOd4riODo0hVTHCJ9DYhWsXM/edit#gid=0

User Behavior Analysis

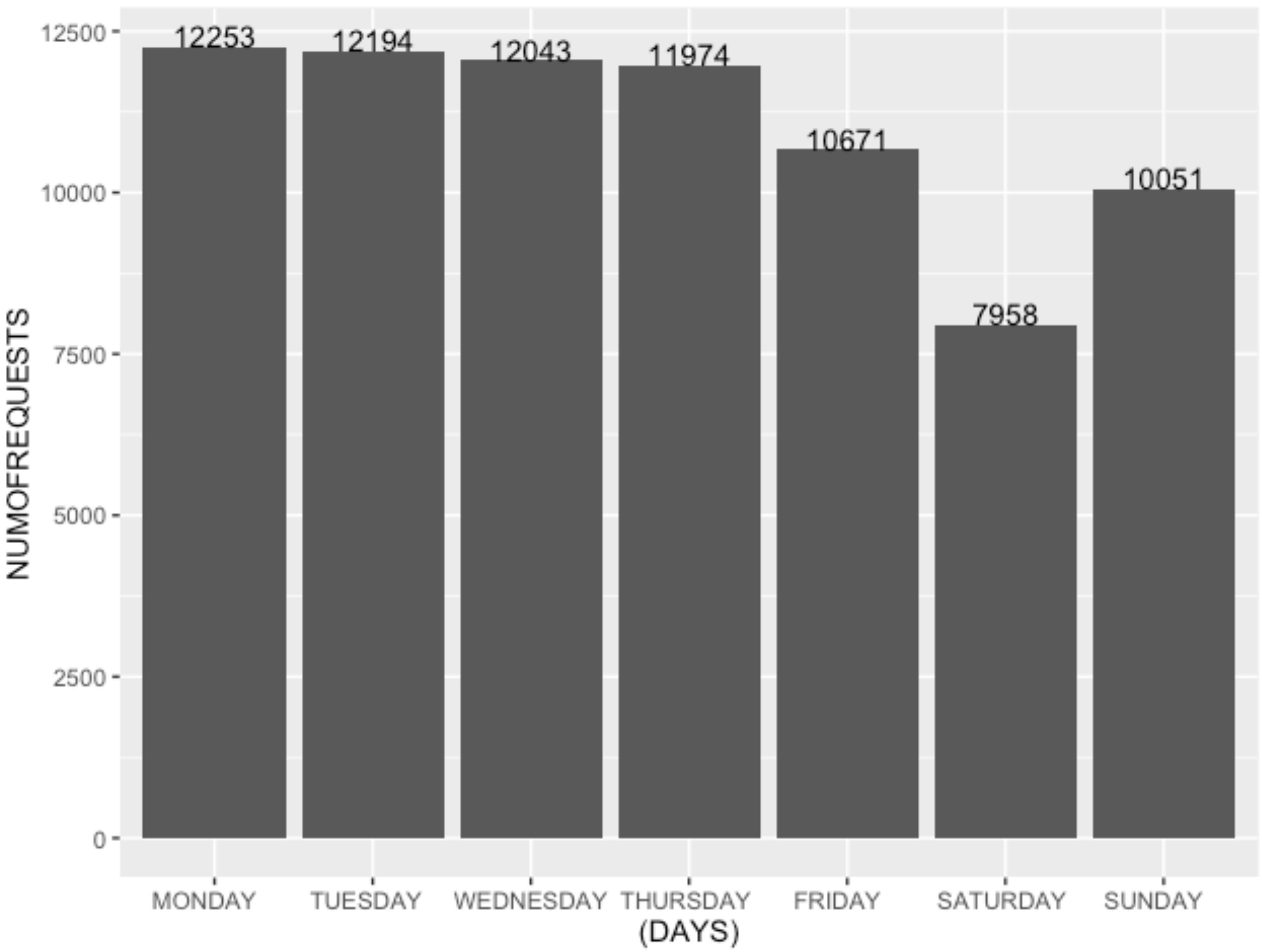


Start Page
REQUEST_STATS
Columns | Data | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:
Actions...

	ID	SERVER	REMOTE	QTIME	QDURATION	LOAD	MEM	DE...	URL	DURATION
1	928843	aipanda054	35.9.71.125	30-APR-15 12.30.10....	30-APR-15 12.30.1...	1.5	38.0		http:/...	1
2	928847	aipanda057	38.126.90.38	30-APR-15 12.30.12....	30-APR-15 12.30.1...	0.3	79.7		http:/...	2
3	928850	aipanda053	35.9.71.125	30-APR-15 12.30.17....	30-APR-15 12.30.1...	1.8	28.3		http:/...	1
4	928862	aipanda043	188.184.69.109	30-APR-15 12.30.28....	30-APR-15 12.30.3...	1.0	14.9		localhost	2
5	928872	aipanda022	188.184.185.129	30-APR-15 12.30.03....	30-APR-15 12.30.4...	53.3	90.4		https:...	37
6	928880	aipanda054	38.126.90.38	30-APR-15 12.30.31....	30-APR-15 12.30.4...	26.9	39.3		http:/...	15
7	928881	aipanda053	38.126.90.38	30-APR-15 12.30.45....	30-APR-15 12.30.4...	2.5	28.3		http:/...	3
8	928890	aipanda053	38.126.90.38	30-APR-15 12.30.58....	30-APR-15 12.31.0...	0.3	28.4		http:/...	2
9	928957	aipanda057	38.126.90.38	30-APR-15 12.32.12....	30-APR-15 12.32.1...	2.5	79.6		http:/...	1
10	928995	aipanda053	38.126.90.38	30-APR-15 12.33.00....	30-APR-15 12.33.0...	0.5	28.2		http:/...	2
11	929041	aipanda056	38.126.90.38	30-APR-15 12.33.40....	30-APR-15 12.33.4...	0.5	30.9		http:/...	1
12	929043	aipanda057	38.126.90.38	30-APR-15 12.33.41....	30-APR-15 12.33.4...	0.5	79.6		http:/...	1
13	929065	aipanda053	38.126.90.38	30-APR-15 12.34.10....	30-APR-15 12.34.1...	0.5	28.1		http:/...	1
14	929160	aipanda057	38.126.90.38	30-APR-15 12.36.17....	30-APR-15 12.36.1...	0.5	79.7		http:/...	1
15	929166	aipanda053	188.184.149.35	30-APR-15 12.36.19....	30-APR-15 12.36.2...	17.4	29.8		http:/...	1
16	929167	aipanda054	38.126.90.38	30-APR-15 12.36.19....	30-APR-15 12.36.2...	1.5	38.6		http:/...	1
17	929168	aipanda056	38.126.90.38	30-APR-15 12.36.20....	30-APR-15 12.36.2...	29.2	30.8		http:/...	1
18	929170	aipanda053	38.126.90.38	30-APR-15 12.36.21....	30-APR-15 12.36.2...	2.6	31.1		http:/...	1
19	929172	aipanda057	38.126.90.38	30-APR-15 12.36.23....	30-APR-15 12.36.2...	4.8	79.7		http:/...	1
20	929174	aipanda054	38.126.90.38	30-APR-15 12.36.24....	30-APR-15 12.36.2...	0.8	38.6		http:/...	1
21	929176	aipanda053	38.126.90.38	30-APR-15 12.36.25....	30-APR-15 12.36.2...	7.6	37.0		http:/...	1
22	929177	aipanda056	38.126.90.38	30-APR-15 12.36.24....	30-APR-15 12.36.2...	0.5	30.7		http:/...	2

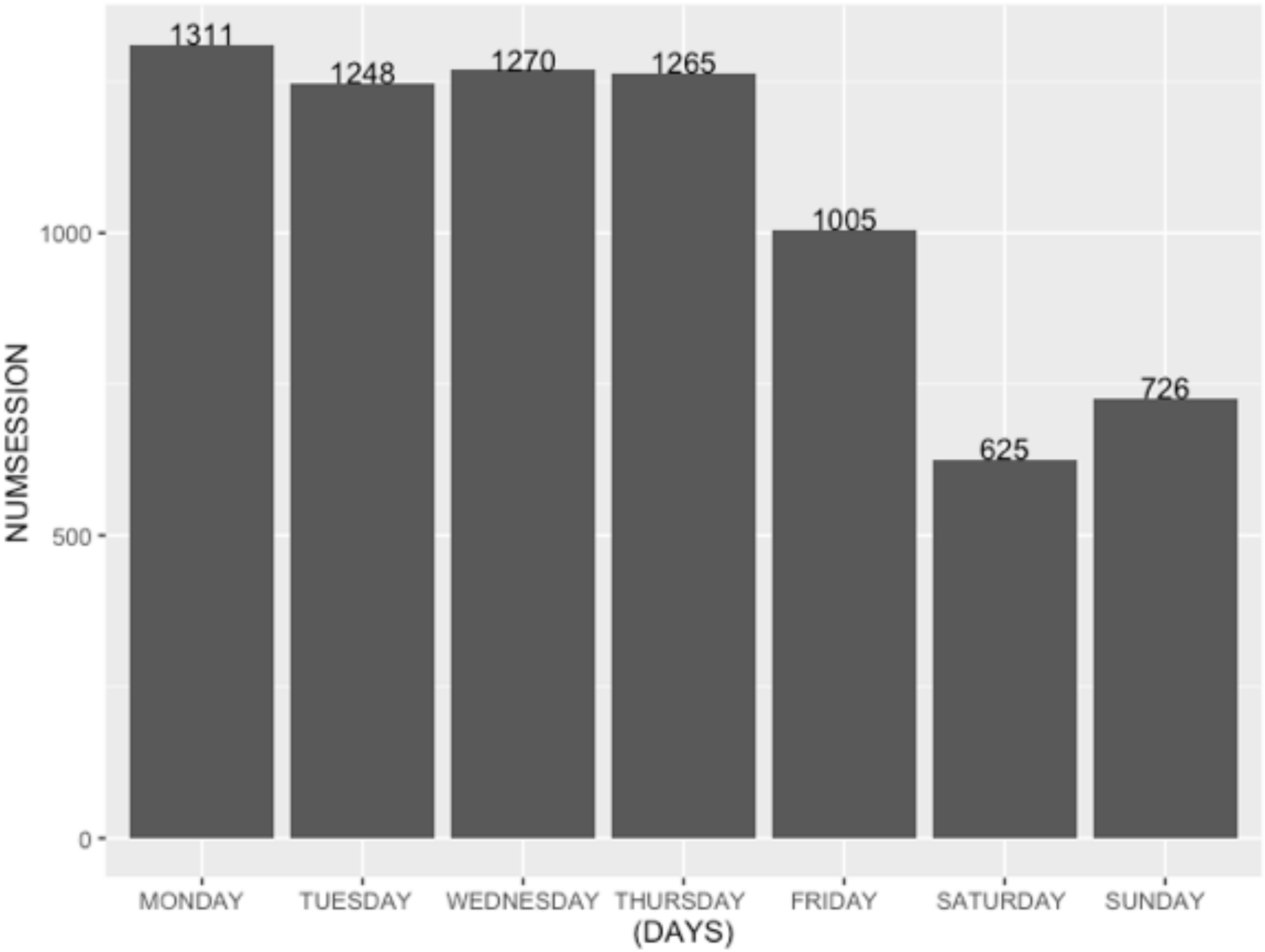
New analytic views were recently developed (SESSIONANALYSIS1, SESSIONANALYSIS2)

User Behavior Analysis

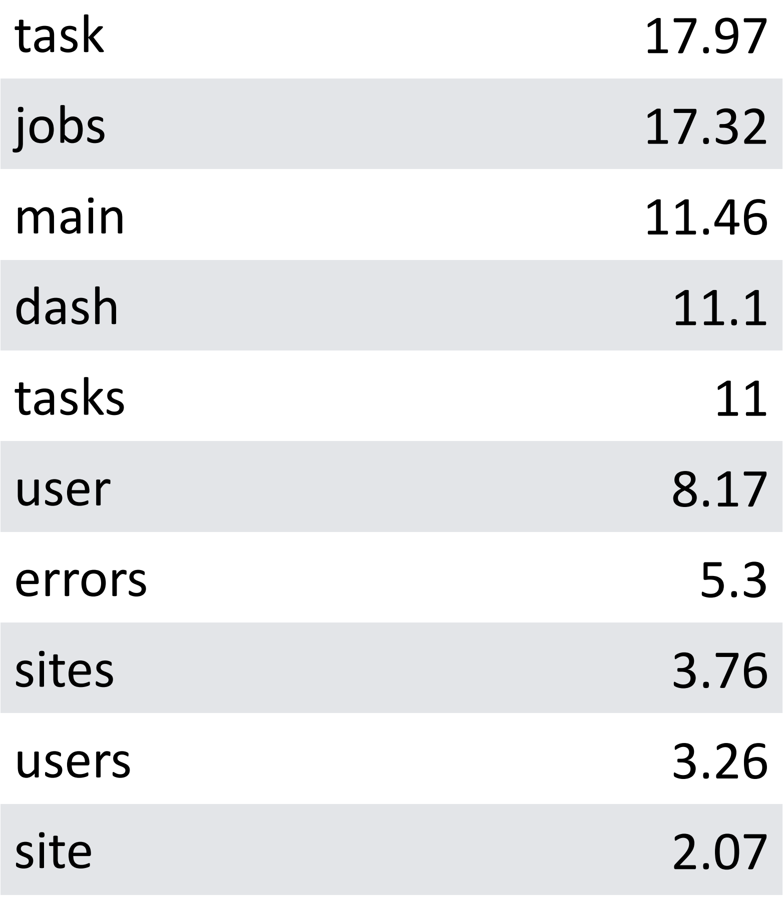


User Behavior Analysis

Session is user actions sequence with timeout less than 30 minutes



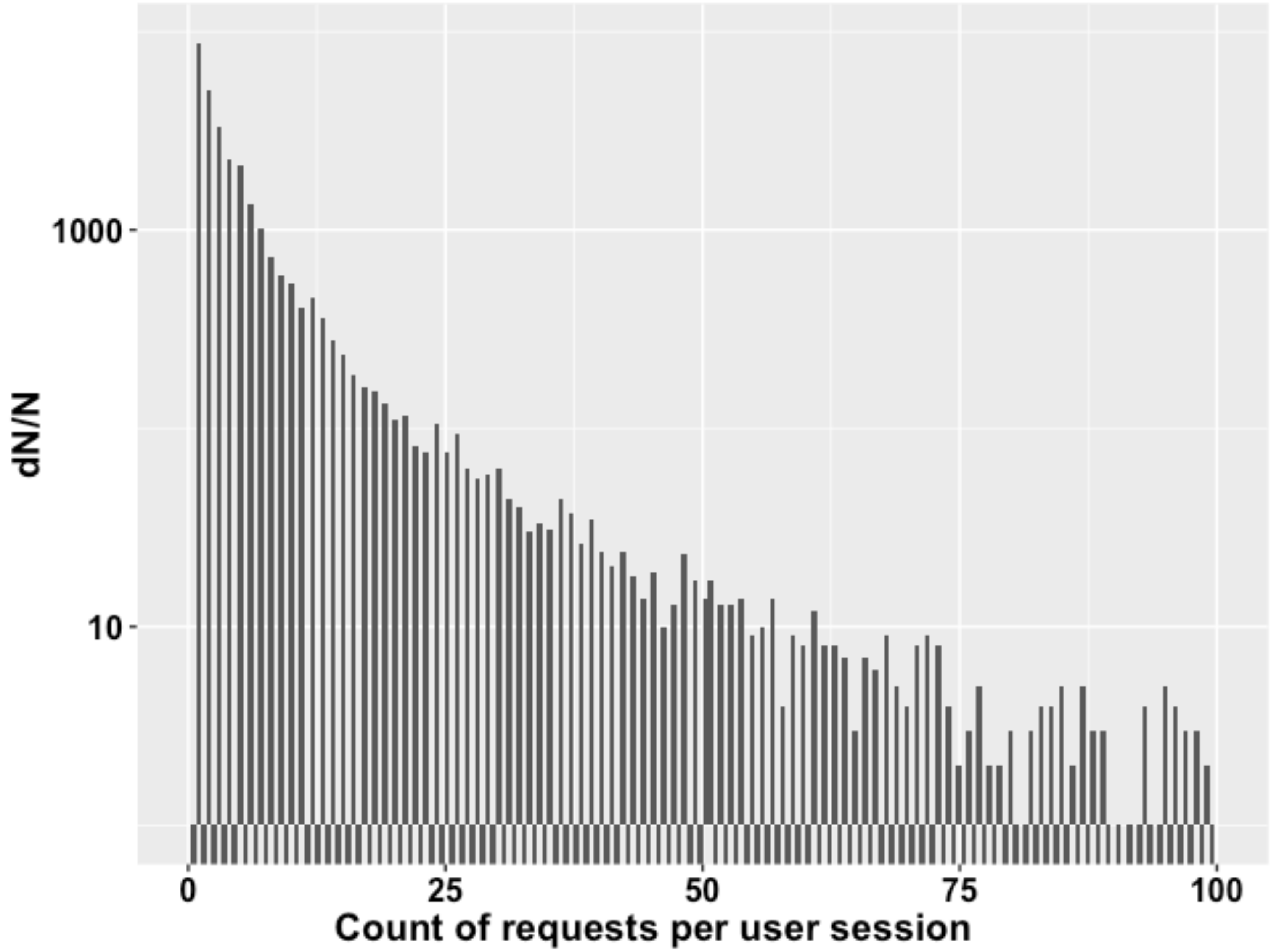
User Behavior Analysis



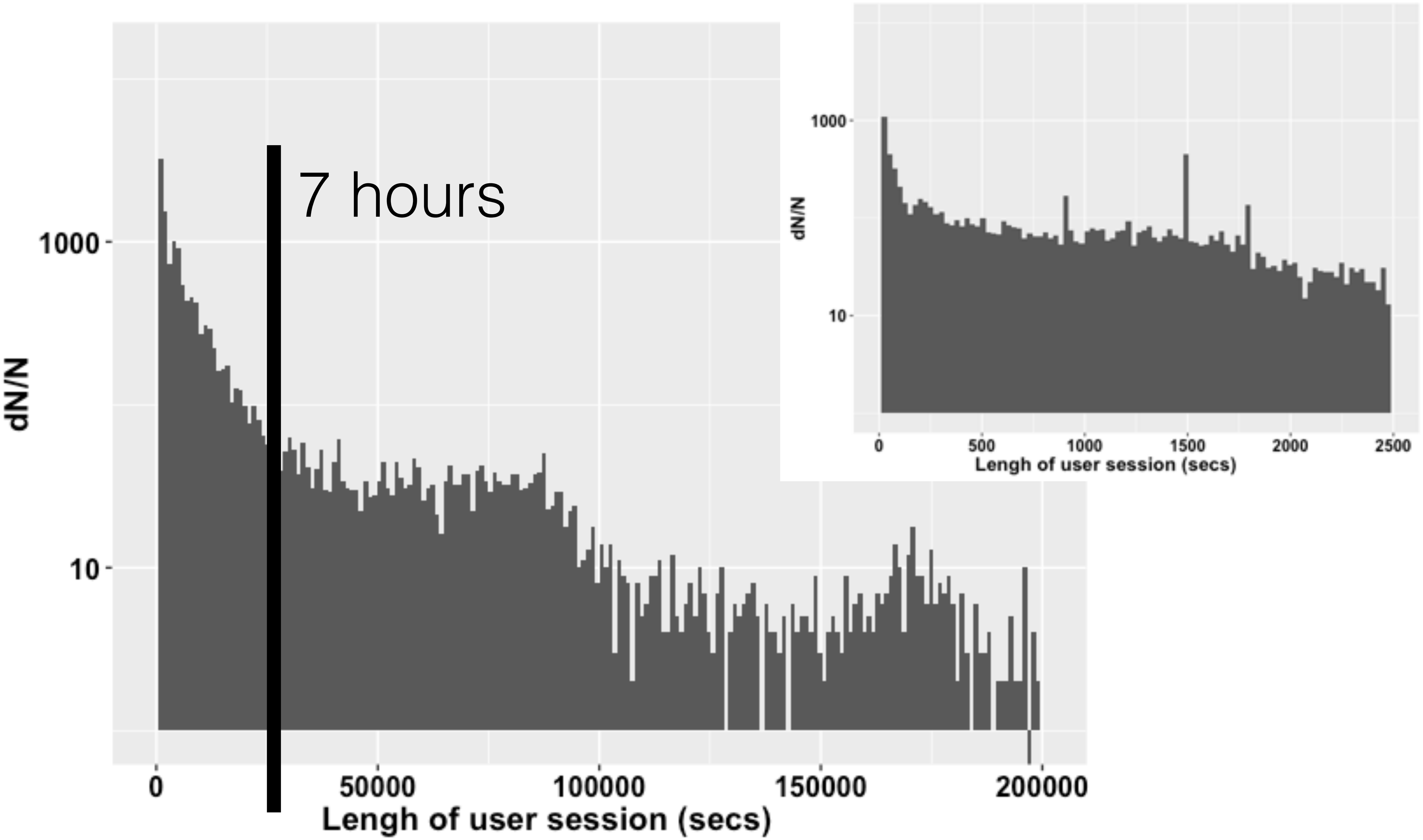
task	17.97
jobs	17.32
main	11.46
dash	11.1
tasks	11
user	8.17
errors	5.3
sites	3.76
users	3.26
site	2.07

Access impact (%), only non cached pages

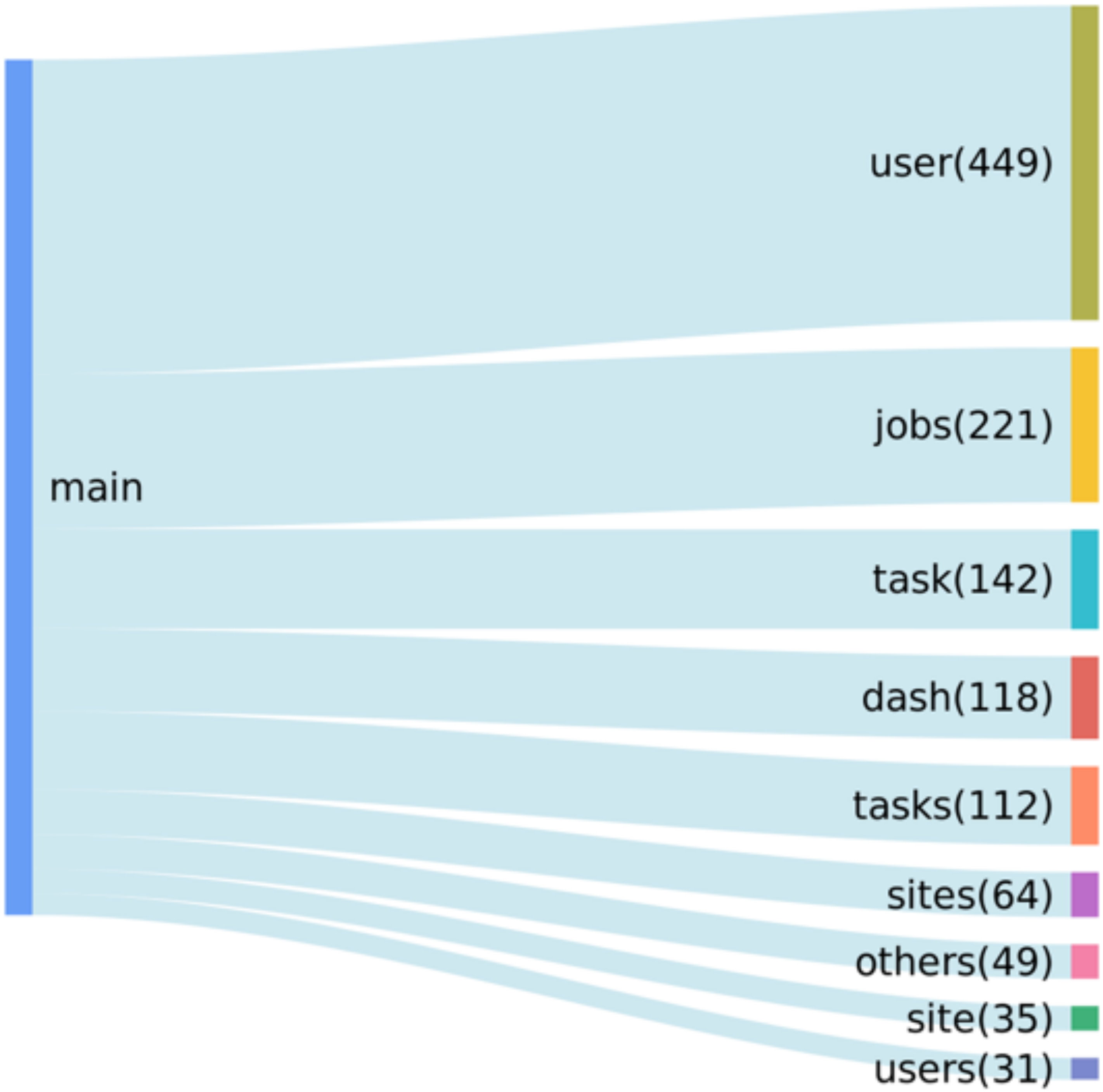
User Behavior Analysis



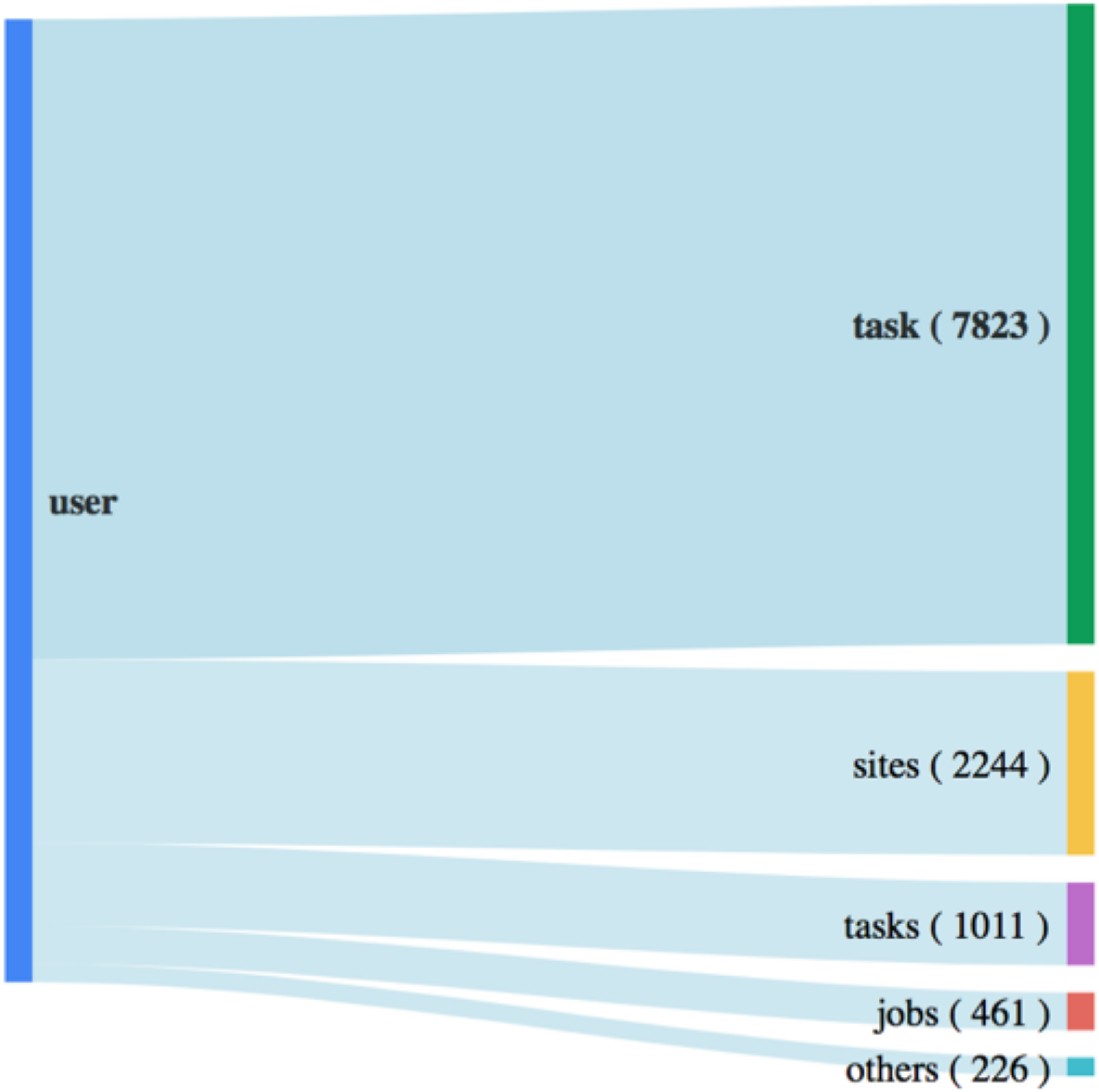
User Behavior Analysis



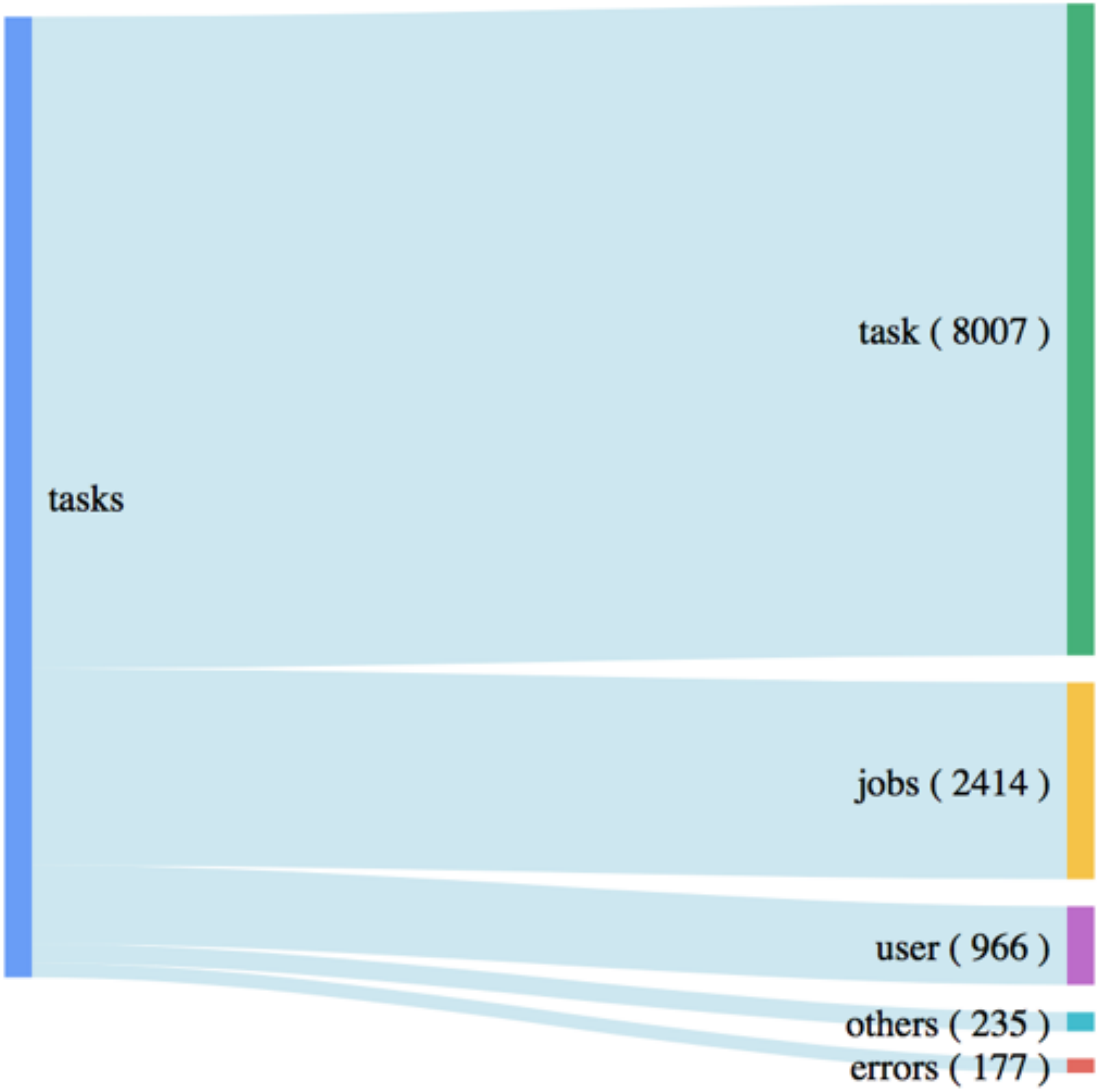
User Behavior Analysis



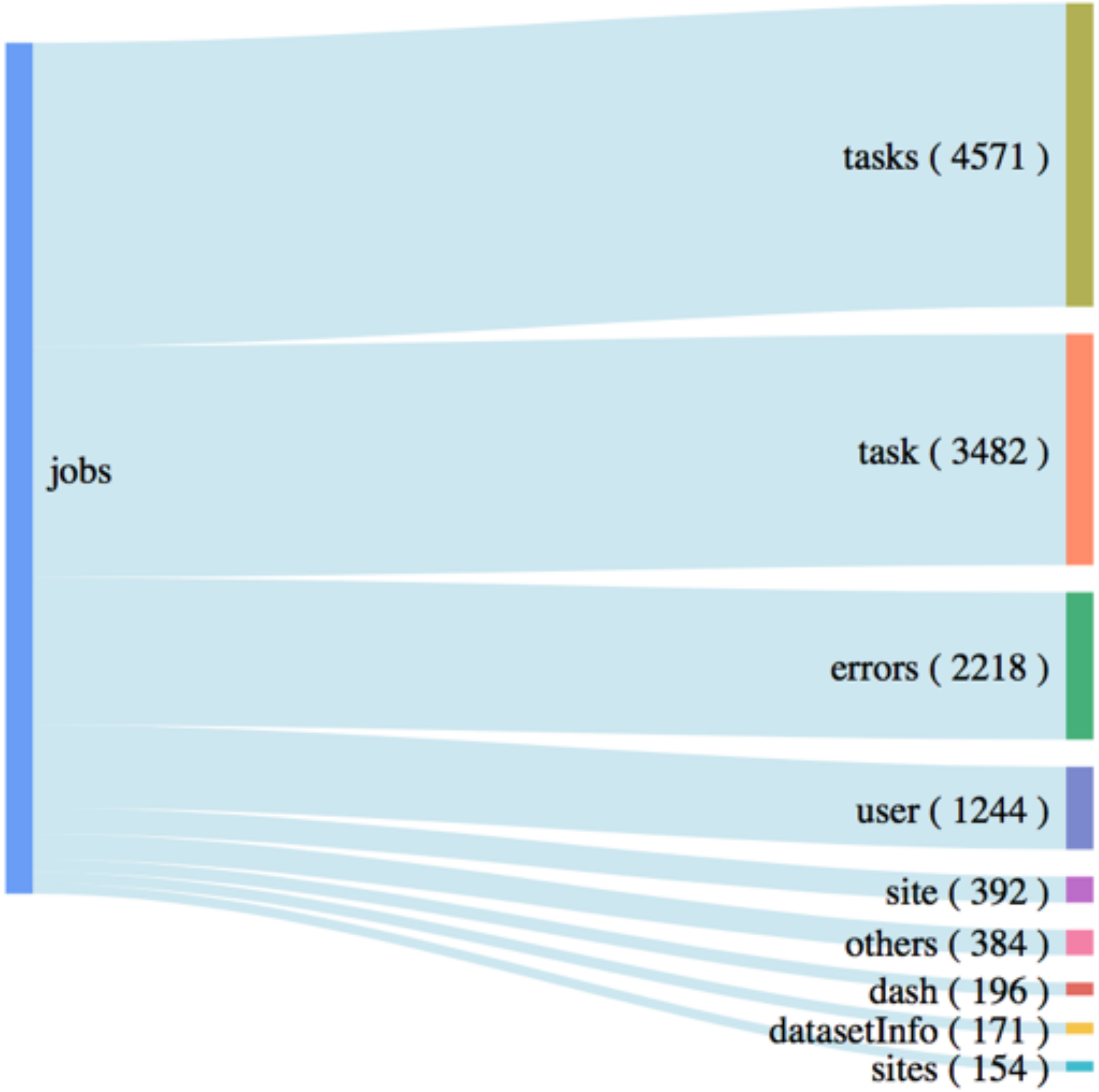
User Behavior Analysis



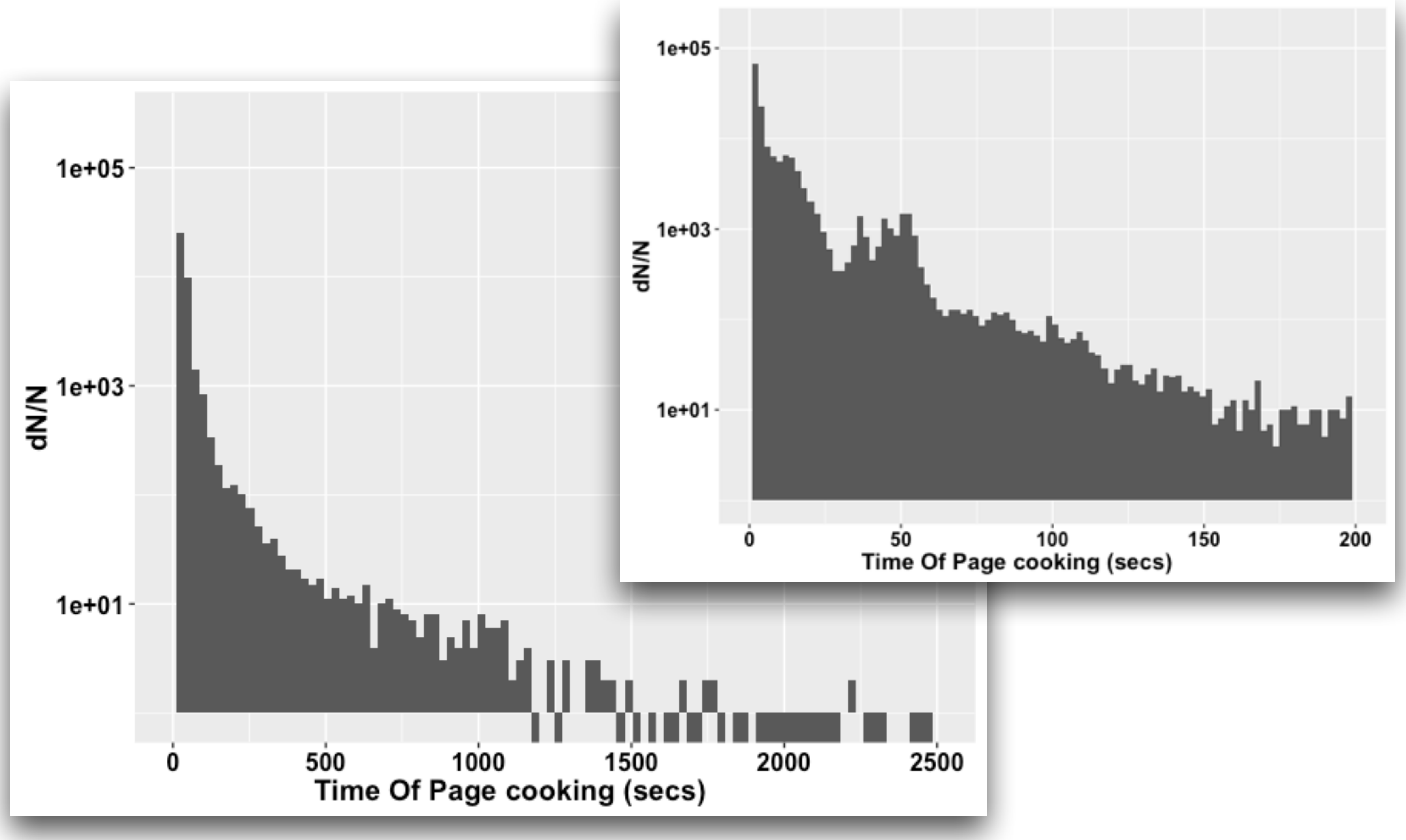
User Behavior Analysis



User Behavior Analysis



System performance



Bottleneck analysis

Jobs page

1. Retrieve jobs (4 active + possible archived)
2. If (dynamic conditions) retrieve JediJobRetryHistory, reconstruct chain
3. if user request conditions, go to JediDatasetContents
4. if (dynamic conditions) go to JediDatasets
5. go to Filestable4
6. if (dynamic conditions) go to FilestableArch
7. if (dynamic conditions) go to Filestable4
8. if (dynamic conditions) go to FilestableArch
9. if (dynamic conditions) go to JEDI_EVENTS

~ 7 different paths of DB access
branched logic

~13 individual DB queries, ~30 MB retrieved from DB,

DB data transfer ~ 1M/sec, 98% of time posix.waitpid

Requirements for the next iteration

Significantly reduce frequency of data access operations and amount of transferred bytes

↓

Most of processing logic should be implemented nearby the data

Tools we (can) try

- Google BigQuery
- Indexed data storage on top of Apache Lucene (Elastic Search, Solr)
- Spark

Google BigQuery

- Pros:
    - Developed by Google (expecting reliability, high performance)
    - Highly scalable
- Cons:
    - UDF is a fresh technology (less than year - small knowledge base)
    - Critical part of monitoring is going to proprietary technology, outsourcing to external company, on shared resource.
    - Every user query costs money (rough estimation: few cents for each processing a large table). The more complex algorithm of the data processing - more it costs. New metric - cost of request. New error - billingTierLimitExceeded.
    - Network connectivity between CERN and BigQuery warehouse become critical part of the ability of monitoring to deliver fresh data to the end user.
    - Deploying monitor to another experiments involves accepting proprietary technologies, payments to google for processing, storage, import, etc, additional technology to support

Apache Lucene solutions (ES, Solr)

A new prototype was developed in 2015:

- Cheap and quick approached - architecture has been kept
- A class to translate Django Query to Solr language developed
- Performance didn’t increase significantly and R&D stopped

Apache Spark

- Pros:
    - Highly scalable (Tencent example with 8000 nodes and 150PB +1 PB a day)
    - Easily deploying on virtual or physical clusters
    - Advanced underlying technologies (full Java support)
    - Native integration with R (one of the most popular analytic and machine learning tool)
    - Open Source
- Cons:
    - Have to maintain cluster or buy service

Spark prototype

Goal: Check is technology stack applicable to monitoring, assess speed

Scope:

- Program skeleton to run embedded Spark server and «user queries»
- Establish mutual data stream
- Perform preprocessing
- Perform fast selection
- Perform complex data aggregation

Spark prototype

- Stand alone Java application
- Few simultaneous version of data
- As new version of data is imported, it become actual and programs address queries to this dataset
- Only skimmed data is transferring (thanks to ORA_ROWSCN)
- Preprocessing is applied
- Speed of data aggregation is 1000 rows / core / sec

Possible scenarios

- Big query:
  - Initial technology assessments, debugging, data streaming, delivery to web server (1-3 weeks)
  - Implementing monitor real case (2-4 weeks)
  - Assessment speed, cost, reliability, working on development plan (1 week)

- Spark
  - Setup the development spark cluster (~1w).
  - Setup the uninterrupted data stream (Oracle-Spark) (~1-2 w).
  - Implementing backend for Errors Page (~1-1.5 months).
  - Implementing backend for DashBoards (~2-2.5 months).
  - Implementing backend for Jobs (~3 months).
  - Implementing backend for Tasks page(~3.5 months).
  - Other pages which cooked with huge delays (~4.5 months).

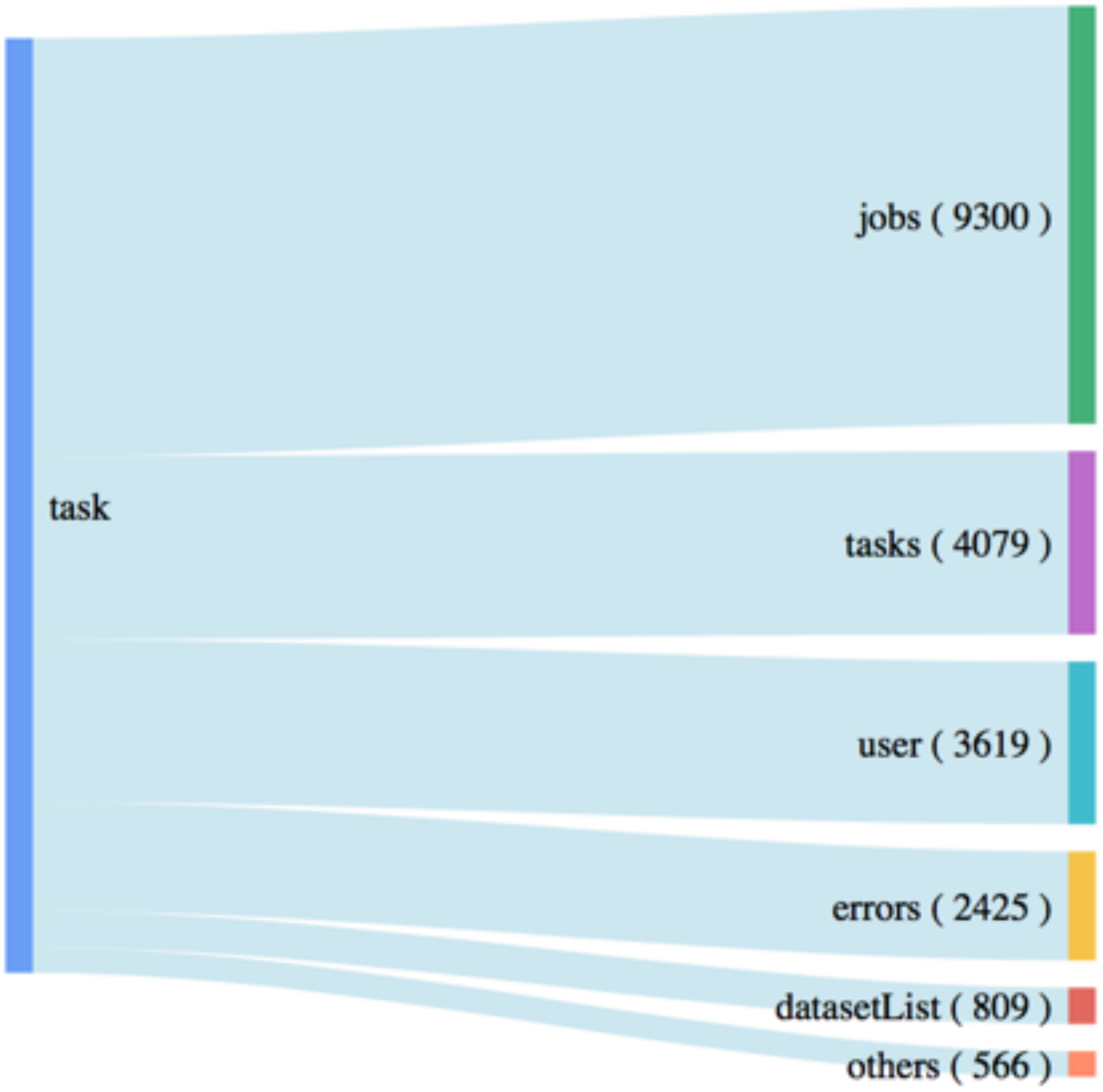
  The best-case estimate: 3 months, the most likely estimate: 4.5 months, the worst-case estimate: 6 months, weighted average (E = (best + 4mostlikely + worst) / 6) = 4.5

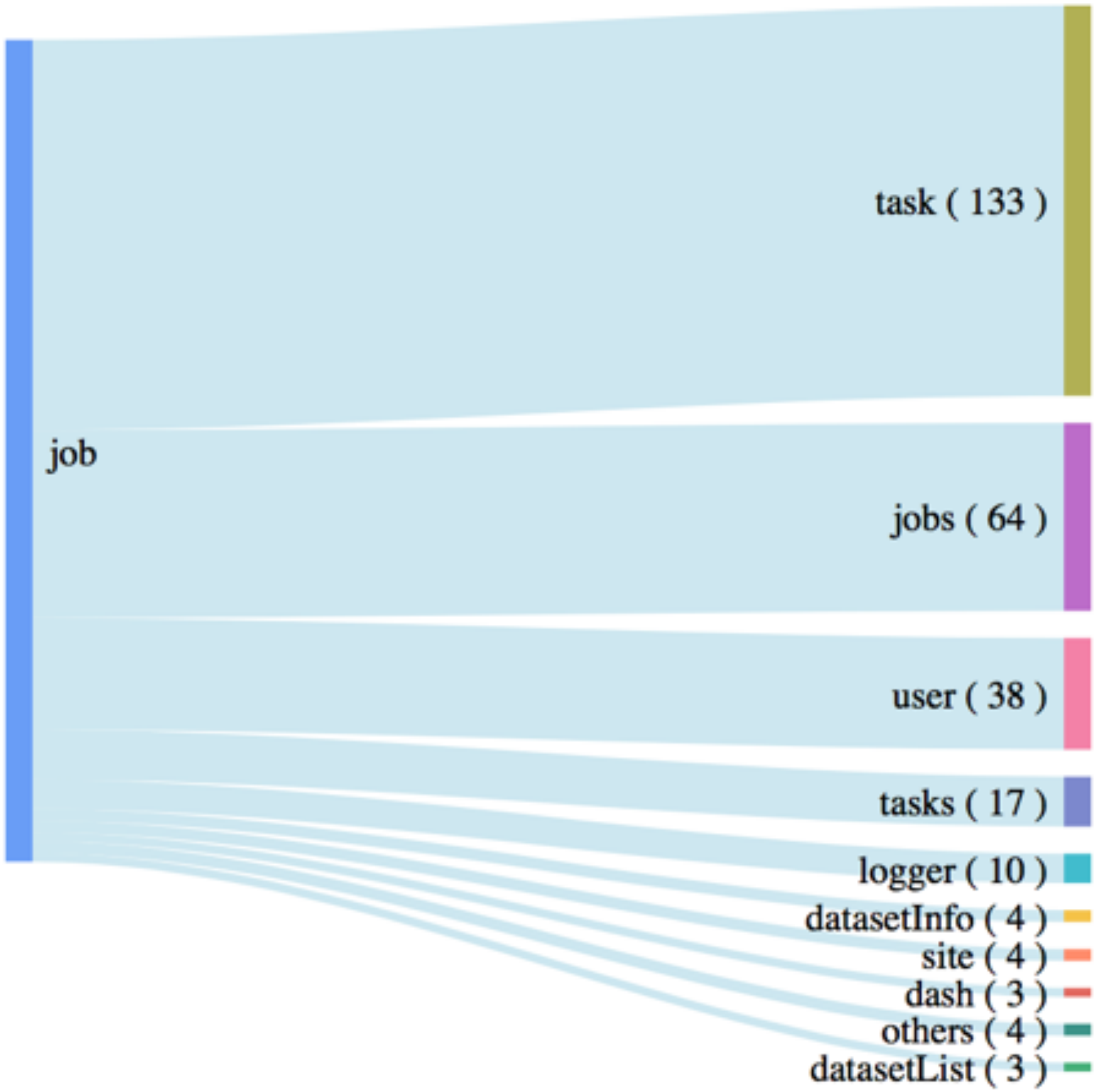
- Elastic search

  Development schema transformation, preprocessing, start importing data, working case prototype (1-2 months)

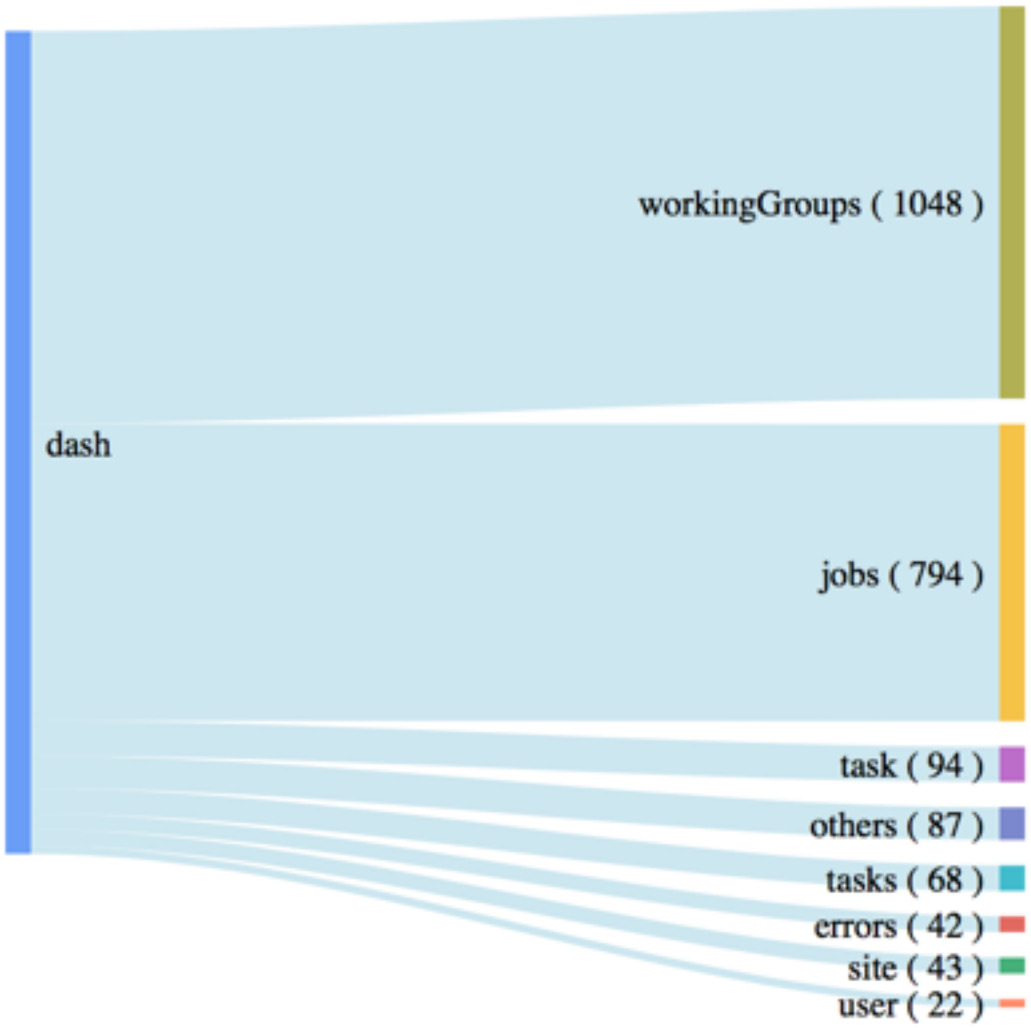
Backup

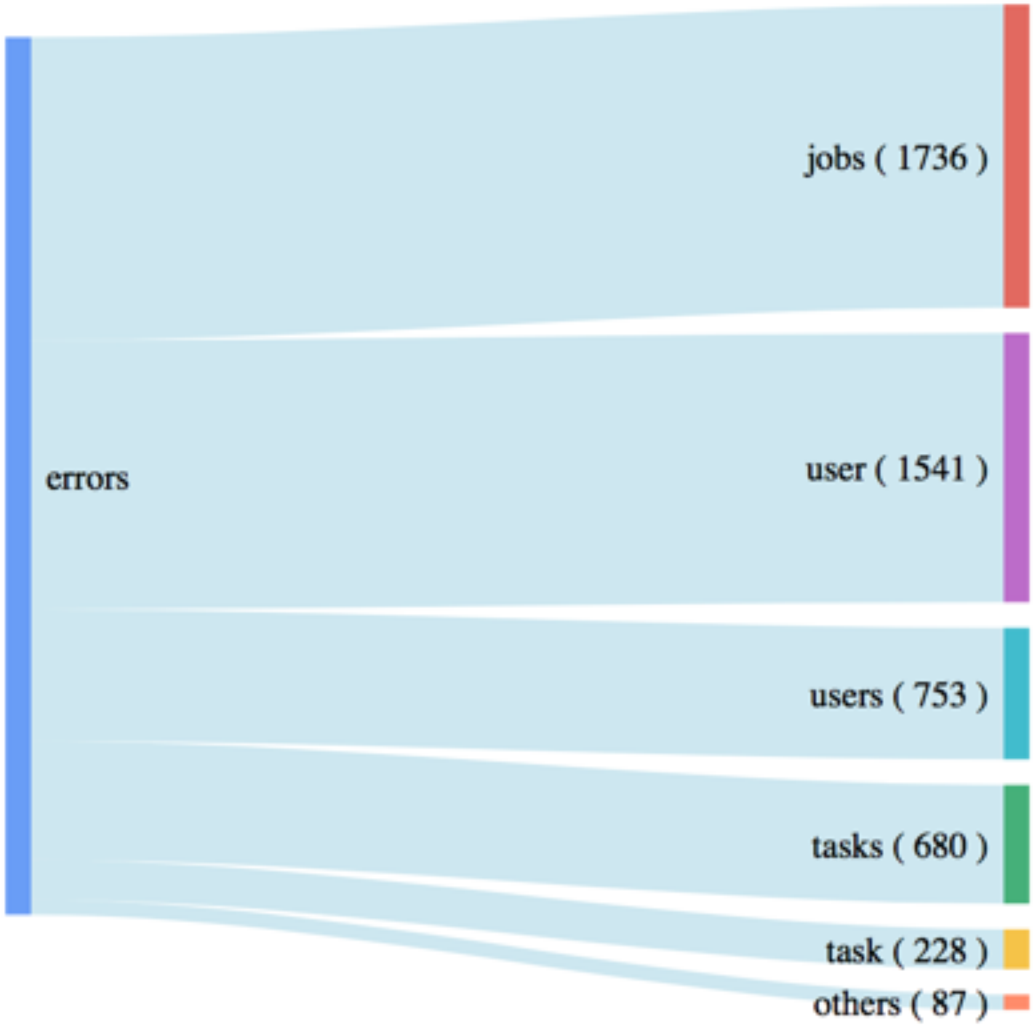
User Behavior Analysis





User Behavior Analysis





BigQuery cost estimation

Lets say we update a biqquery warehouse once a 10 minutes. That correspond to 220M of data transfer only for the jobs table. $0.01 (per 200M) x 6 (times per hour) x 24 (hours per day) x 365 = 525.6$. Just for refreshing jobs table

In the most simple case when we retrive jobs from jobsarchine table processing only MODIFICATIONTIME field:
10GB*0.001*5 = 5 cents per one query to jobsarchive.