

Supporting CREDO project with scalable data acquisition and processing infrastructure

Maciej Pawlik, Krzysztof Oziomek, Marek Magryś, Patryk Lasoń, Piotr Homola

Presentation plan

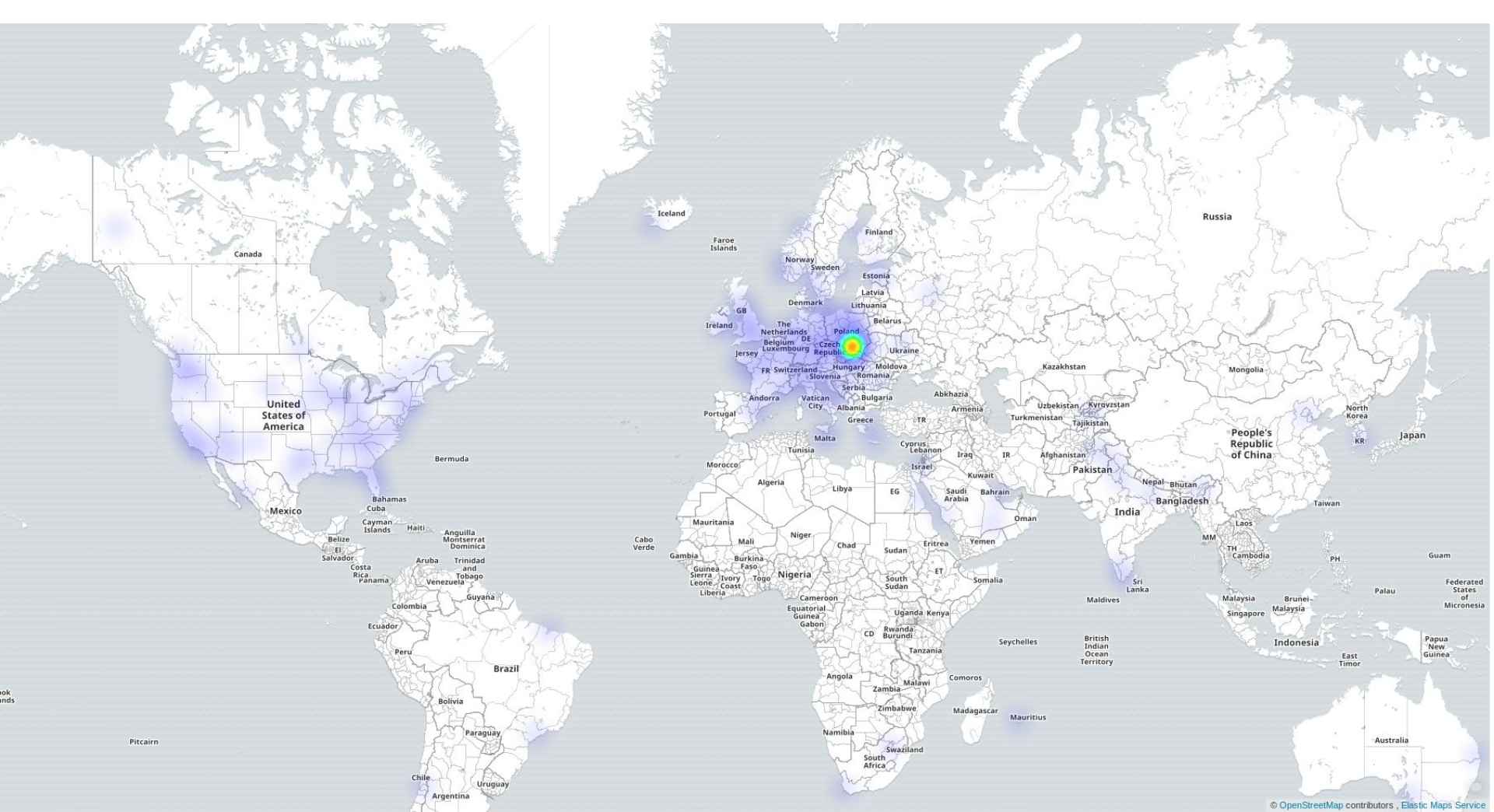
1. Introduction to CREDO, technical details
2. Contribution / challenges
3. CREDO ecosystem components
4. Backend infrastructure
 - a. Software
 - b. Hardware
5. Conclusions and references

CREDO project

- Cosmic Ray Extremely Distributed Observatory

aims to:

- Expand our knowledge about the universe, understand dark matter
- Detect super-preshower phenomena - particle showers
 - Use mobile phones as a network of detectors covering a large area
- multiple other applications, including:
 - detecting changes in earth's magnetic field
 - predicting earthquakes
 - influence on human physiology
 - education, community involvement
- Depends on community involvement, community sourced data



Some statistics

- 700K+ visible detections (2M+ overall)
- 1M+ device pings (sums up to 50 years of looking for particles)
- 3K+ users with at least 1 detection
- 5K+ devices
- 1K+ user teams
- 10s of GBs used for storage of data, metrics and backups

Contribution / challenges

- Backend infrastructure
- Provide computing and storage resources
- Manage ingestion of data
- Provide data accessibility
- Integrate into existing ecosystem
- Encourage community involvement

this implies:

- Developing software for gathering and storage information about detections
- User management
- Providing means for extensibility in multiple areas

Non functional design goals

- Open Source everything:
 - API's
 - Server application
 - Detectors (software and hardware!)
 - Tools
- Provide documentation
- Apply *single purpose principle*, *KISS*, etc.
- Provide means for growth -> improve science reproducibility!





credo-science

Repositories 14

People 15

Teams 0

Projects 0

Find a repository...

Type: All

Language: All

New

CREDO-PC-Windows

Forked from marekmarcos/CREDO

Credo Detector. Cosmic ray detection by Webcam. Source code in Lazarus Pascal. PC / Windows

Pascal 1 MIT Updated 16 hours ago

Maps

Updated 7 days ago

credo-hackathon-nkg

Release of hackathon issue

Python MIT Updated 12 days ago

credo-detector-android

CREDO detector as Android application

Kotlin 2 4 MIT 1 issue needs help Updated 12 days ago

credo-webapp

Credo web application

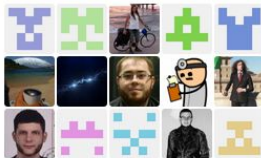
Python 1 2 MIT Updated 14 days ago

Top languages

Python C++ Kotlin CSS
Pascal

People

15



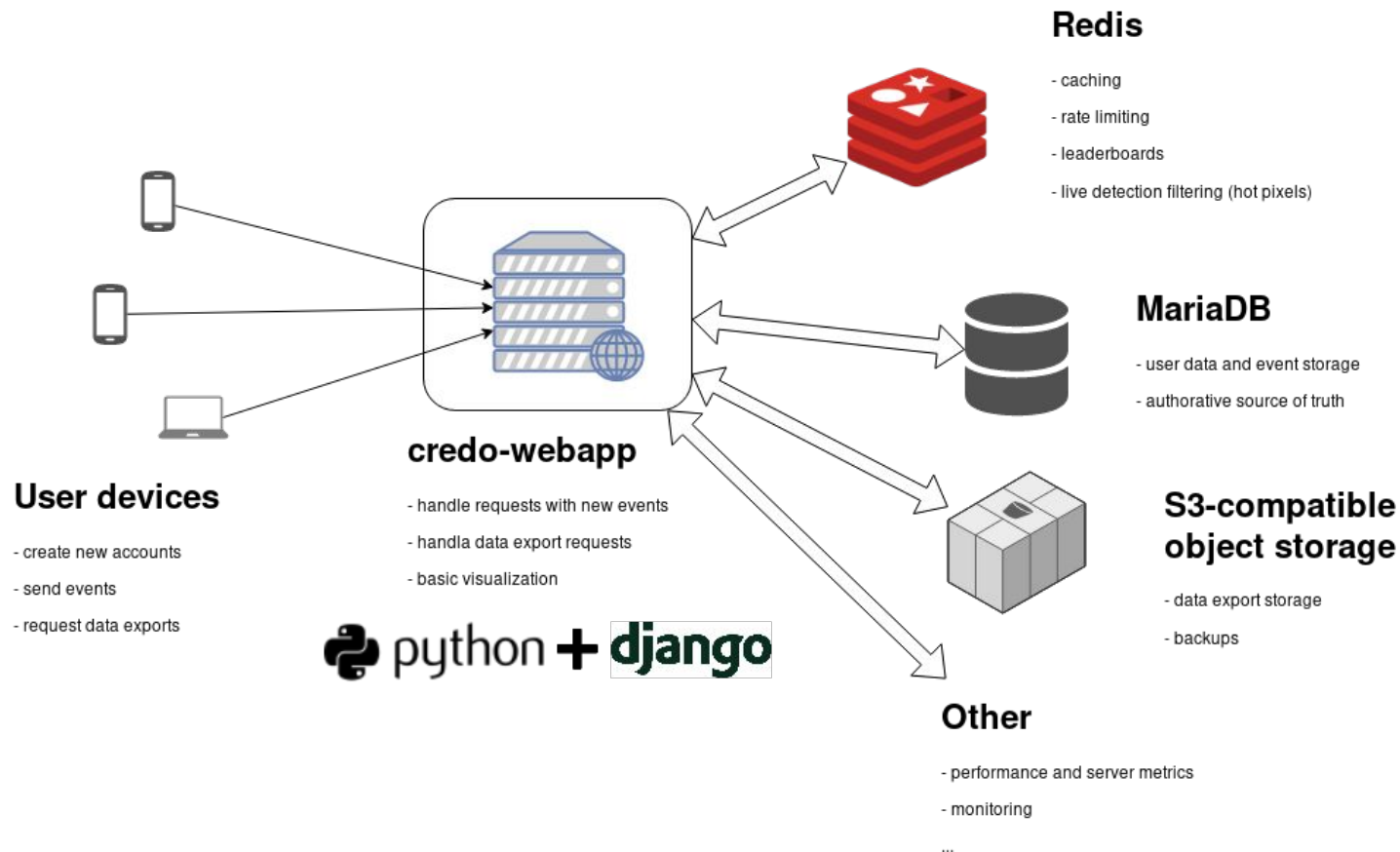
<https://github.com/credo-science>

Software components of CREDO project

- credo-webapp (Server application)
- credo-api-tools (Utilities)

- credo-detector-android
- CREDO-monitor-TimeClusteringAlgo
- Credo-Desktop-Detector
- CREDO-PC-Windows
- ...

Component diagram



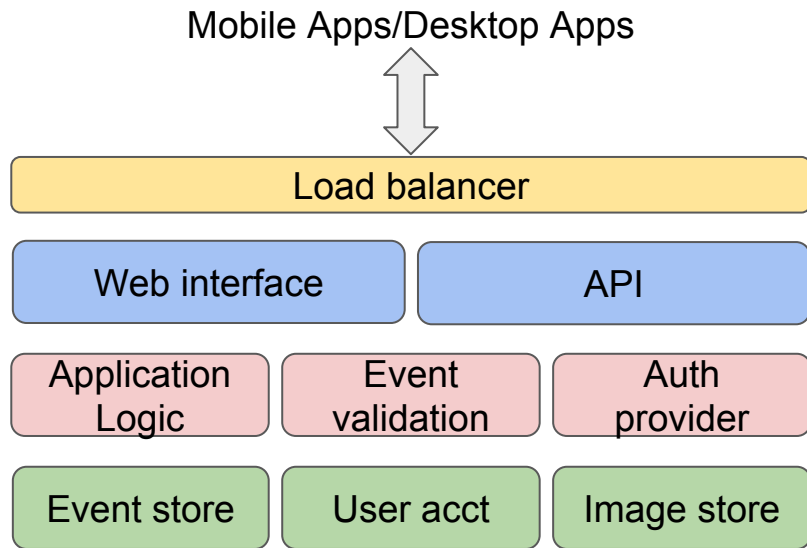
Credo Server application

- <https://github.com/credo-science/credo-webapp>
- Django app running under Apache and mod_wsgi
- Display real time detection information
 - basic on-line analysis of data
 - filtering based on provided criteria
- Provides API for other components
 - versioned APIs!
- Manages user accounts
- Provides data export facilities



django

django
REST
framework





Main page

Detections (702821)

Users (3711)

Teams (1503)

Top users

Recently registered users

Login	Detections	Login	Detections
Barti	32992	Izka	0
Wzorzysty	23901	m.jurczak	0
smph-kitkat	22626	Mateusz	0
Natalia	17389	Patrycjaborczyk	0
Emanuele Maria Latorre	17129	Nivv	0

Last 20 detections

date	login	team	img
2018-10-22 09:57:50.135	Patryk Olszowski	CREDO ASP OXFORD	
2018-10-22 09:56:07.544	smph-kitkat	IFJ	
2018-10-22 09:55:44.510	Patryk Olszowski	CREDO ASP OXFORD	
2018-10-22 09:52:49.200	Marcin Kuzniak		

<https://api.credo.science>

Credo Api Tools

- <https://github.com/credo-science/credo-api-tools>
- Collection of tools designed to simplify process of exporting and analyzing available data
- “data-exporter” handles making API calls to authenticate user, request and download data export incrementally
- “data-processor” handles incremental data processing and provides simple plugin interface for scientists to write their own code

Hardware infrastructure

- Virtualized environment
- Hosted at Cyfronet's cloud
- Runs on 8 VMs
- Periodic multi tier backups

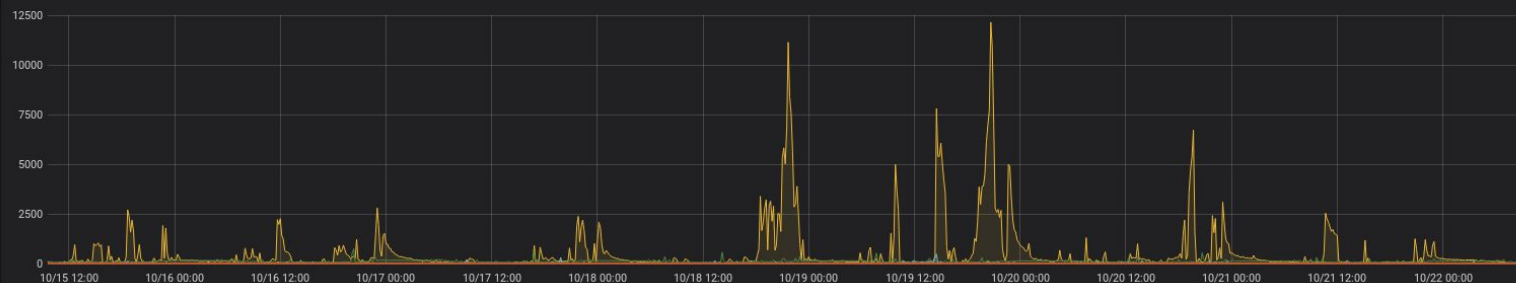
- Automated deployment, through Ansible

- Available resources allow for scaling if required

Performance and availability monitoring

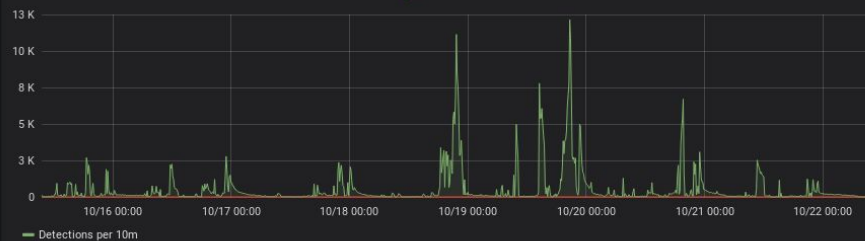
- Application level
- Operating system level
- Time series DBs: django-influxdb-metrics + InfluxDB + Grafana
 - request count
 - latency
- Event and log monitoring: Raven + Sentry
 - warnings/errors
- Elasticsearch + Kibana:
 - operating metrics and utilization (Metricbeat)
 - log shipping (Filebeat)
 - high-level overview of gathered data (heatmaps, rates)

Requests per 10m

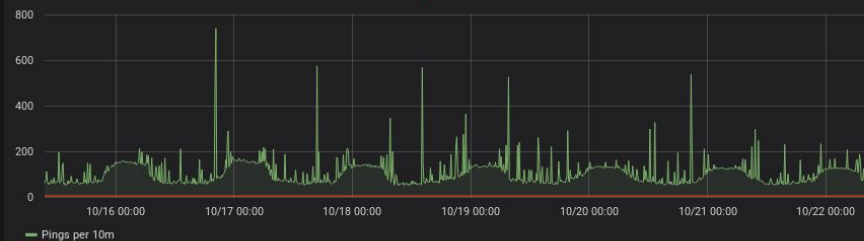


	total
DetectionView	530772.0
PingView	108240.0
team_page	10585.0
team_list	9874.0
user_page	9866.0
UserLoginView	2373.0
index	1408.0
user_list	1379.0
InputFrameHandler	1060.0
UserRegistrationView	624.0
report	457.0
confirm_email	347.0

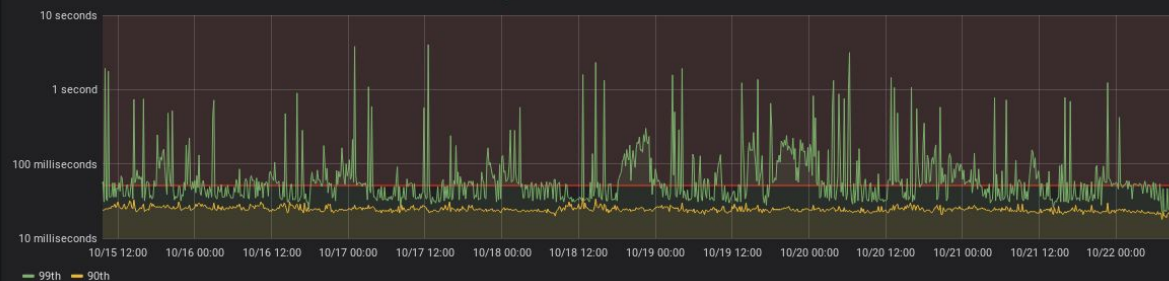
Detection count



Ping count



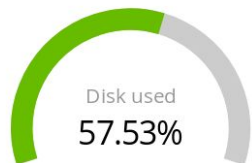
Detection/Ping latency percentiles



Slowest views (95th percentile)

view	Time spent
detection_list	2 seconds, 88 milliseconds
user_page	423 milliseconds
team_page	309 milliseconds
index	303 milliseconds
finished_jobs	295 milliseconds
UserRegistrationView	233 milliseconds
DataExportView	204 milliseconds
MannionEventView	108 milliseconds

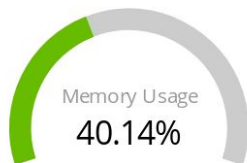
Disk used [Metricbeat System]



Load Gauge [Metricbeat System]



Memory Usage Gauge [Metricbeat Sys...



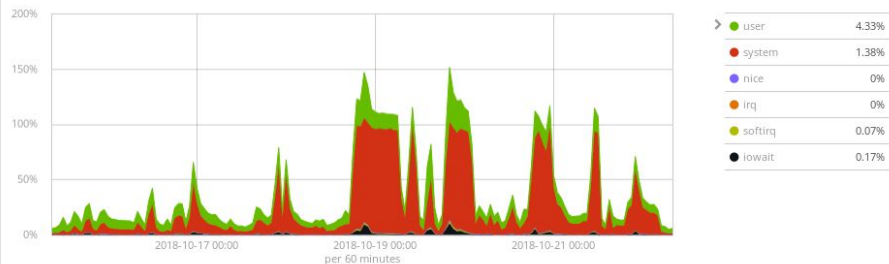
Inbound Traffic [Metricbeat System]

Inbound Traffic
2.0KB/s
Total Transferred 3.3GB

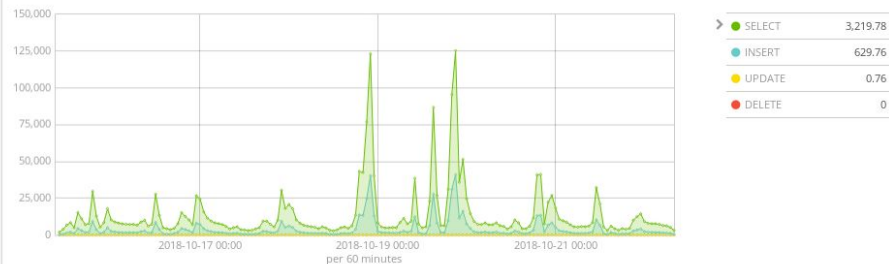
Outbound Traffic [Metricbeat System]

Outbound Traffic
5.5KB/s
Total Transferred 28.8GB

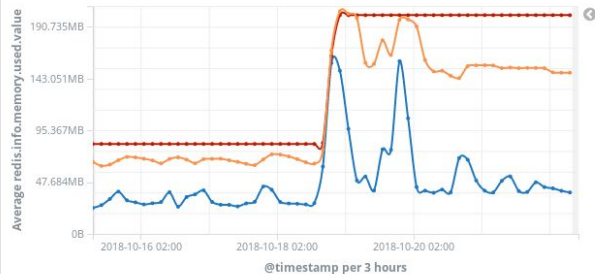
CPU Usage [Metricbeat System]



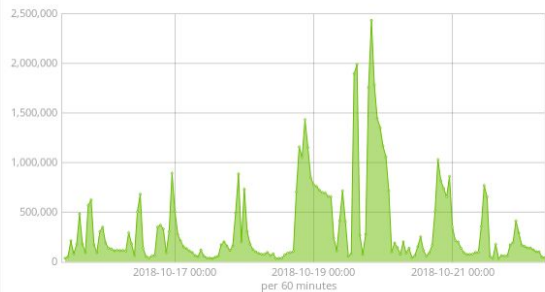
Command rates [Metricbeat MySQL]



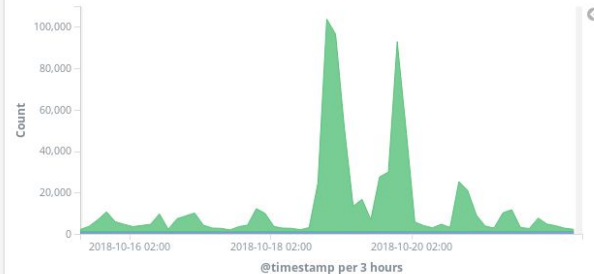
Redis memory usage



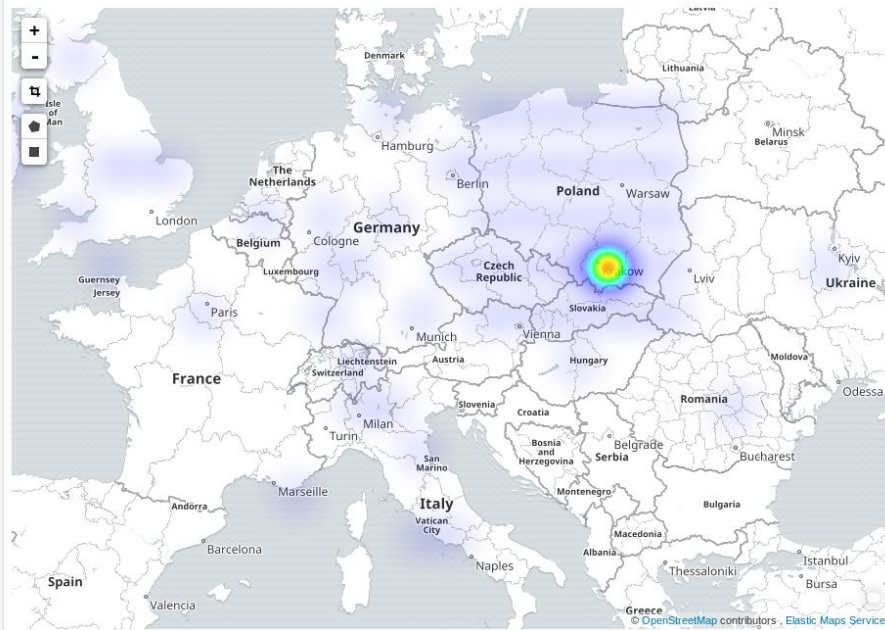
Redis OPS



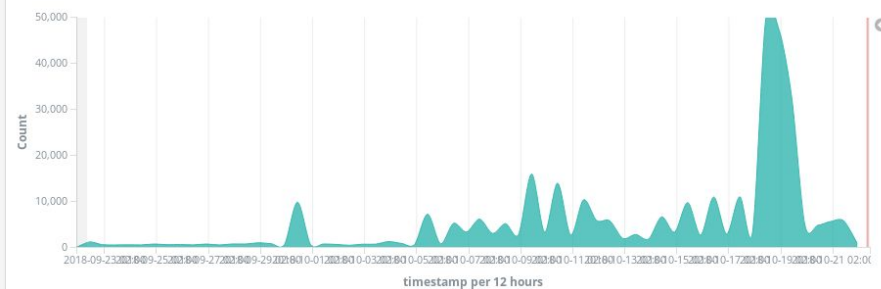
Keyspaces [Metricbeat Redis]



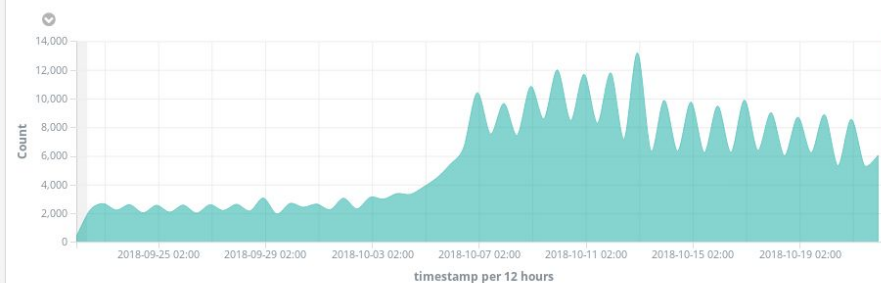
[CREDO] Detection heatmap



[CREDO] Visible detections rate



[CREDO] Ping rate



[CREDO] Top users (detection count)

User ID	Detection count
8006	32,992
8003	23,956
6983	17,389
8001	14,114
6977	11,462
Other	234,060
	333,973

[CREDO] Top users (by on time)

User ID	On time	Ping count
4980	3 months	13,868
4125	3 months	13,457
157	2 months	9,139
1163	2 months	4,288
7358	a month	4,708
Other	5 years	302,462
	6 years	347,922

Conclusions

- Implemented system delivered the required functionality
 - Implementation methodology encourages community involvement
- APIs proved to be reusable, multiple integrated components
- Methodology used to develop software was a success, multiple contributions

remaining challenges:

- Traffic can be unpredictable - ongoing work

References

1. Homola, P., Bhatta, G., Bratek, Ł., Bretz, T., Cheminant, K. A., Castillo, D. A., ... & Jarvis, J. F. (2018). Search for Extensive Photon Cascades with the Cosmic-Ray Extremely Distributed Observatory. arXiv preprint arXiv:1804.05614.
2. Conrad, C. C., & Hilchey, K. G. (2011). A review of citizen science and community-based environmental monitoring: issues and opportunities. *Environmental monitoring and assessment*, 176(1-4), 273-291.
3. CREDO's first light: The global particle detector begins its collection of scientific data: https://www.eurekalert.org/pub_releases/2018-10/thni-cfl100418.php (accessed 4.10.2018)