Arkadiusz P. Trawiński, PhD

Jun 2024, PyData, London

ING

Who am I?

I'm Arek

- Graduated in Physics and Computer Science
- PhD in High Energy Physics

- AI Product Lead/Data Scientist in AIOps team Tech Infra, ING
- In ING for 4 years

- Contact: Arkadiusz.Trawinski@gmail.com

- Mentor in EMCC standards

TRAWIŃSKI, A. (ARKADIUSZ) <arkadiusz.trawinski@gmail.com>

Telemetry data

Telemetry data refers to information collected by software applications or systems during their operation. In the context of IT infrastructure, telemetry data might include details such as
- transaction and error rates, response times,
- CPU and memory usage,
- disk I/O, and network throughput,
- logging messages and tracing.
All they are **timestamped records**, either structured or unstructured, with metadata.

We will focus only on logs

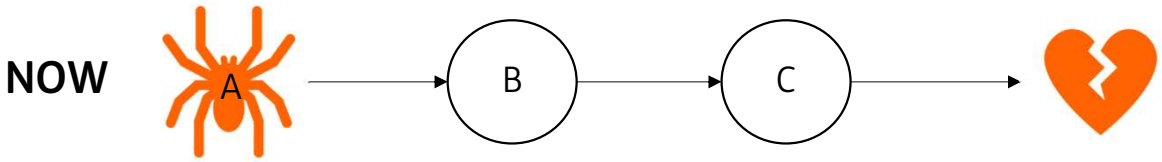
While metrics, traces, and logs are all essential for observability, **logs have the biggest legacy** among these telemetry signals. Most programming languages include built-in logging capabilities or widely used logging libraries.

What is causing what?

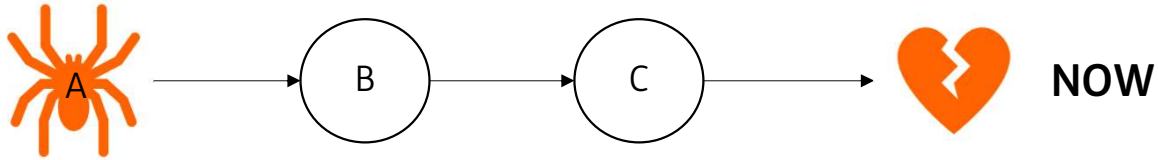
In the complex IT system, we have many applications working together, impacting one another. We are trying to discover a time correlation between them based on logs only.

Two use cases:

a) Building an alert system for detecting coming incidents.



b) Minimalizing the time for finding root cause of an issue.



However, the same goal is:

☞ Founding probability of detecting B given A and the mean time of occurrence.

Before Demo...

some theoretical basics

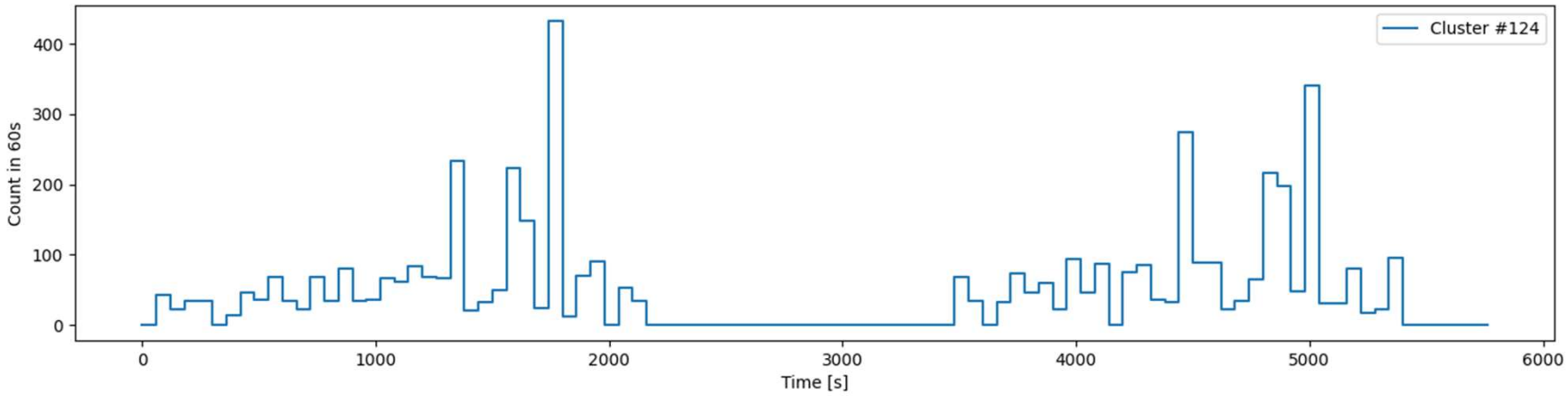
Few examples

Original log message	Drain3 log template
ERROR executor.Executor: Exception in task 8.0 in stage 0.0 (TID 11)\njava.io.IOException: Failed to create local dir in /opt/hdfs/nodemanager/usercache/curi/appcache/...	ERROR executor.Executor: Exception in task <:NUM:>.<:NUM:> in stage <:NUM:>.<:NUM:> (TID <:NUM:>) java.io.IOException: Failed to create local dir in /opt/hdfs/nodemanager/usercache/curi/appcache/...
INFO storage.BlockManagerInfo: Added broadcast_194_piece0 in memory on mesos-slave-26:35542 (size: 4.2 KB, free: 27.8 GB)	INFO storage.BlockManagerInfo: Added broadcast <:NUM:> piece0 in memory on mesos-slave-<:NUM:>:<:NUM:> (size: <:NUM:>.<:NUM:> <:*:> free: <:NUM:>.<:NUM:> GB)
INFO spark.SparkContext: Starting job: max at IPLoM.py:557	INFO spark.SparkContext: Starting job: max at IPLoM.py:<:NUM:>
INFO storage.BlockManagerInfo: Added rdd_62_0 in memory on mesos-master-1:39721 (size: 138.4 KB, free: 27.1 GB)	INFO storage.BlockManagerInfo: Added rdd <:NUM:> <:NUM:> in memory on mesos-master-<:NUM:>:<:NUM:> (size: <:NUM:>.<:NUM:> <:*:> free: <:NUM:>.<:NUM:> GB)
INFO scheduler.DAGScheduler: Got job 134 (max at IPLoM.py:557) with 13 output partitions	INFO scheduler.DAGScheduler: Got job <:NUM:> (max at IPLoM.py:<:NUM:>) with <:NUM:> output partitions

Log messages over time

After clustering log messages, we can plot number of occurrence for a specific cluster/type over time.

☞ This set up a time scale.



Hawkes process

Hawkes processes are point processes defined by the intensity:

$$\forall i \in [1 \dots D], \quad \lambda_i(t) = \mu_i + \sum_{j=1}^{D} \sum_{t_k^j < t} \phi_{ij}\left(t - t_k^j\right)$$

where:

- D is the number of clusters,
- μ_i are the baseline intensities,
- ϕ_{ij} are the kernels,
- t_k^j are the timestamps of all logs of cluster j.

☞ Objective:

$$\forall \, t \in [0 \dots T], \qquad \forall i \in [1 \dots D], \qquad N_i(t) = \int_0^t \lambda_i(t')\, dt'$$

For more information contact Joost Göbbels our former master student.

Possible parametrization of the kernels

1. An exponential kernel

$$\phi_{ij}(t) = \alpha^{ij}\beta^{ij}\exp\left(-\beta^{ij}t\right)1_{t>0}$$

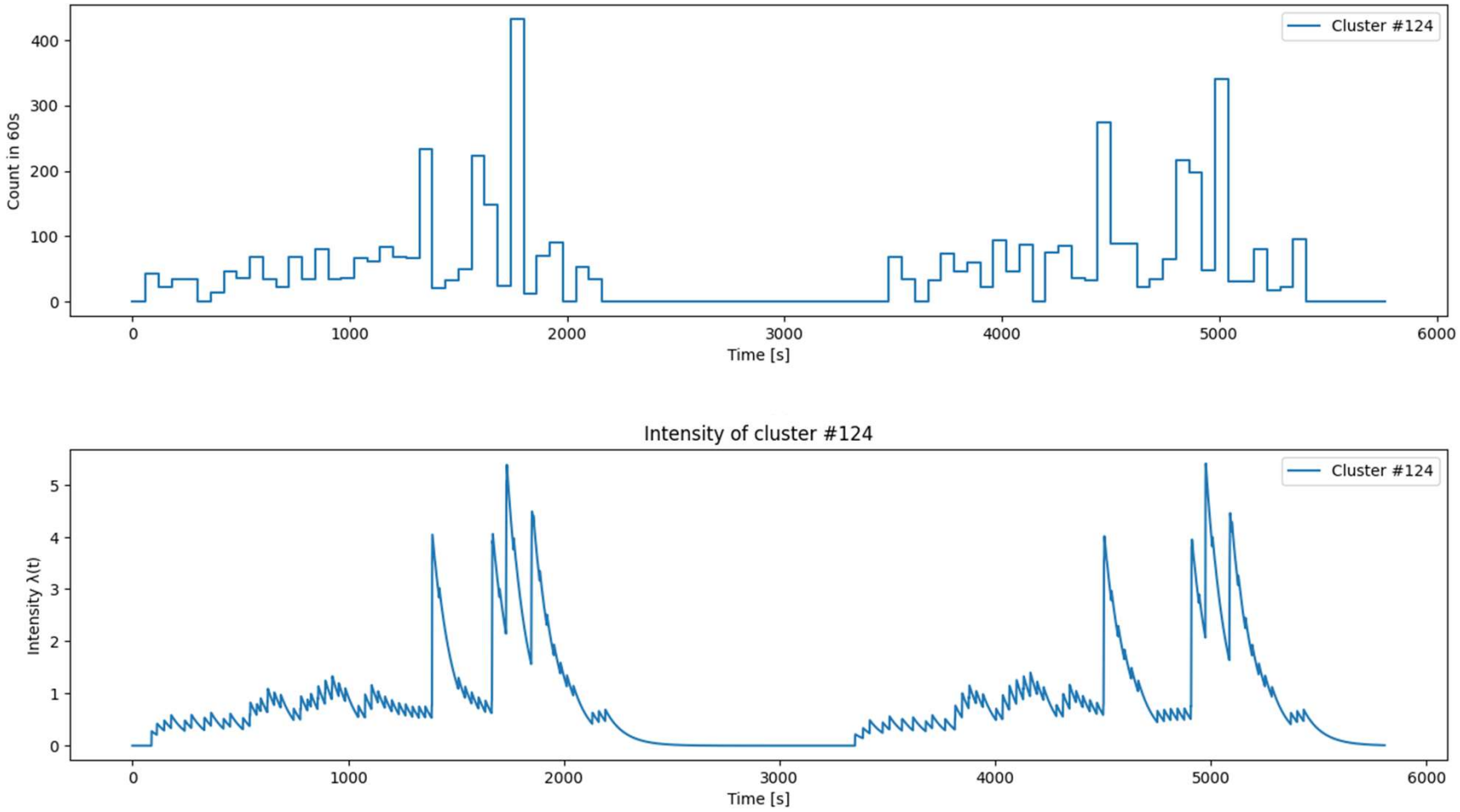
2. A Gaussian kernel

$$\phi_{ij}(t) = \frac{\alpha^{ij}}{2\pi\sigma^{ij\,2}}\exp\left(-t^2/\left(2\sigma^{ij\,2}\right)\right)$$

3. A kernel as sum of basis functions

$$\phi_{ij}(t) = \sum_{m=1}^{M} \alpha_m^{ij}\ f_m(t)$$

Result of fitting Hawkes process to Spark log messages



do your thing