inloco

ABRIL 2019

Construindo Uma Plataforma Data-intensive do Zero

Um estudo de caso na In Loco

Vinícius M.R. Cousseau

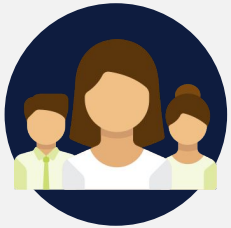
Engenheiro de software na In Loco, trabalhando há 3 anos no time de R&D em localização. Mestrando no CIn-UFPE, pesquisando na área de deep learning aplicada a record linkage. Trabalhou previamente com Subsurface Scattering.

viniciuscousseau

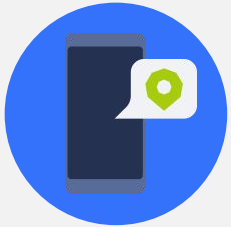
Objetivo

Mostrar um roadmap da nossa experiência na In Loco ao desenvolver uma plataforma data-intensive do zero, para gerar insights e direcionamentos.

Quem somos?



60 MM
usuários únicos em nossa base



16 TB
de dados colhidos diariamente

1.5 BI
visitas registradas mensalmente

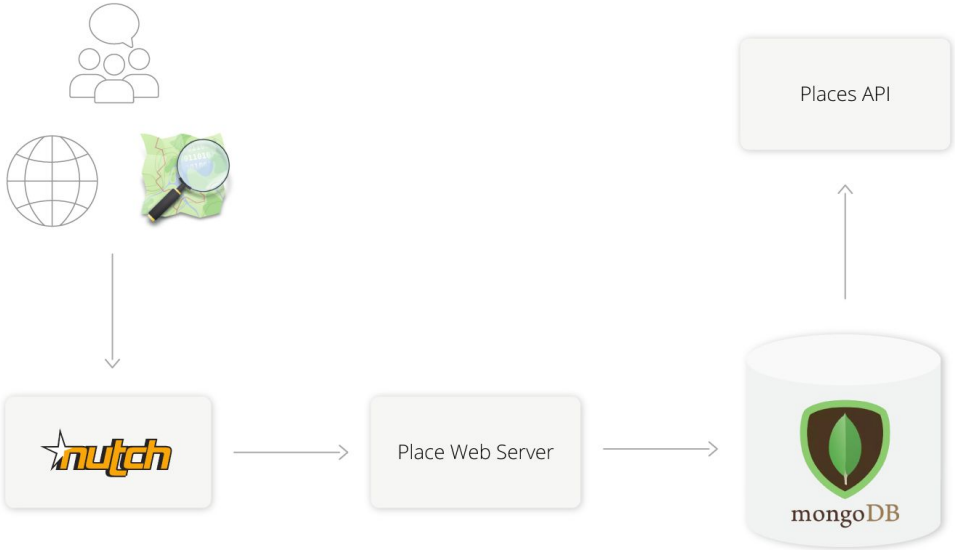
+28MM
locais mapeados no mundo

Desafios

- **Mapear os locais do mundo**
- **Fornecer esses dados através de uma plataforma**
- **Processamento On-line + Off-line**
- **Third party é inviável**

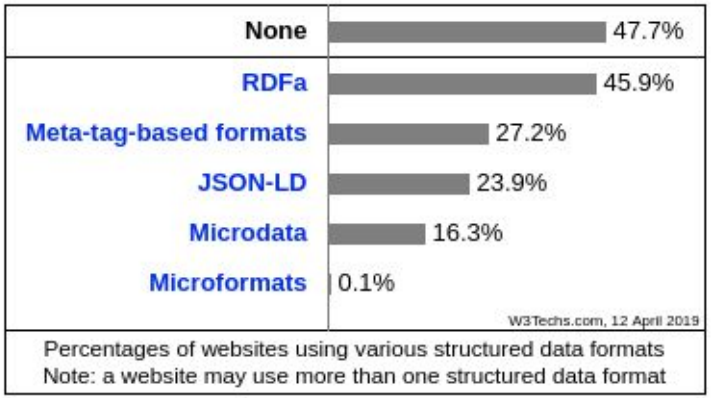
First Steps

Data Extraction



Data Extraction

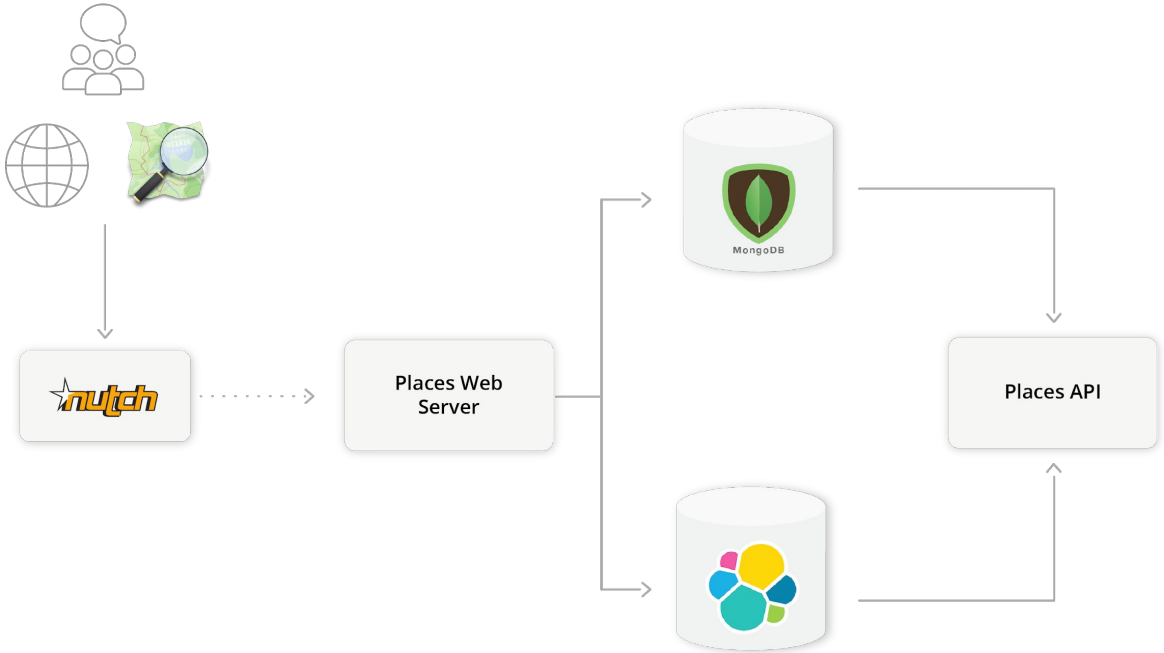
- ***Focused Crawling*** **com** ***nutch plugins*** **de dados estruturados (sempre respeitando as regras)**



Data Extraction

- 120k records por dia
- 5k RPM na Places API
- Apenas duas rotas
- Expor desde o t0 nos trouxe muitos ganhos

Escolha de Bancos

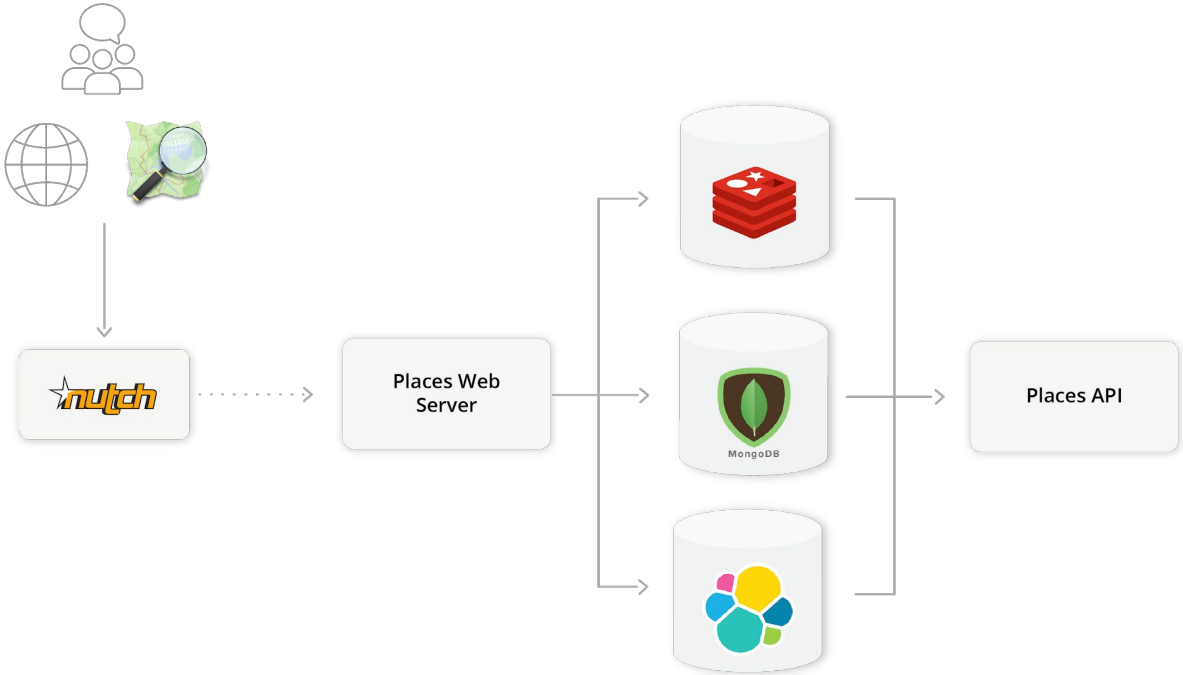


Escolha de Bancos

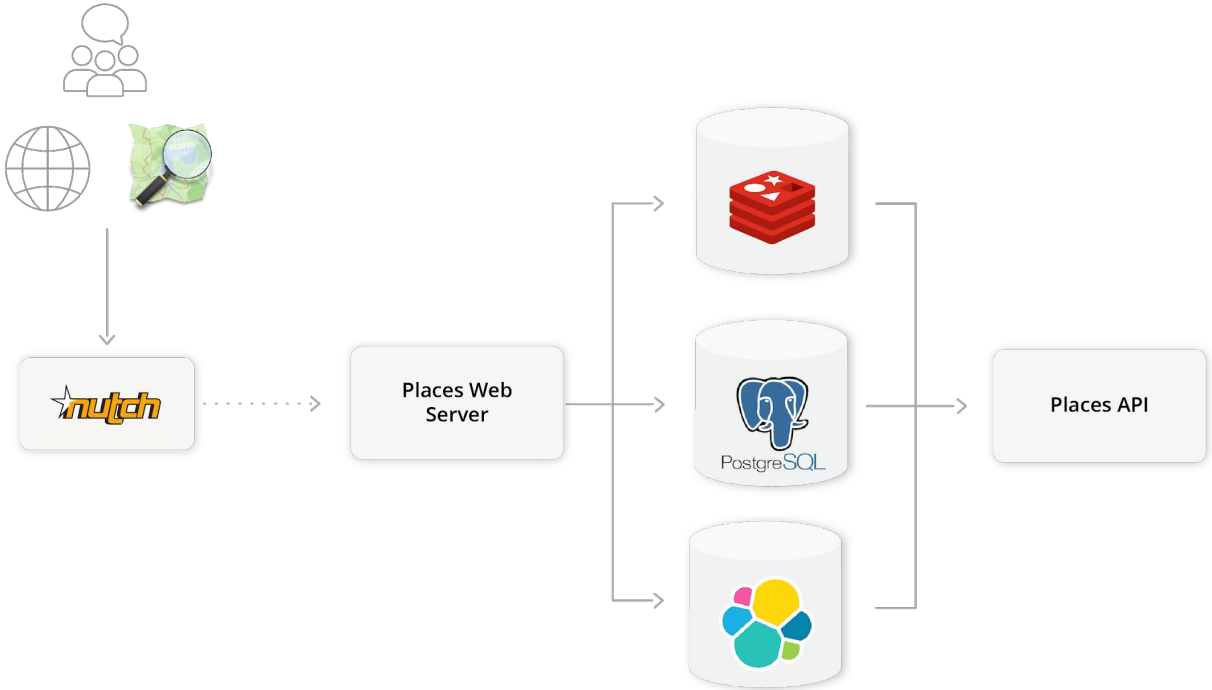
Database	Mean Memory	Mean CPU	Mean Throughput
Redis 3.2.9	5.2% (420MB)	50%	44438.7 rps
Mongo 3.2.9	2% (164MB)	100%	5676.9 rps

Geo Queries dinâmicas em um raio de 200m e limite de 100 resultados. Banco com ~25m records

Escolha de Bancos



Escolha de Bancos



Escolha de Bancos

Escolha de Bancos

- **O domínio dos dados é um forte influenciador**
- ***No silver bullet* (out of the box)**
- ***Benchmark*!**

Processamento em Batch

Processamento off-line

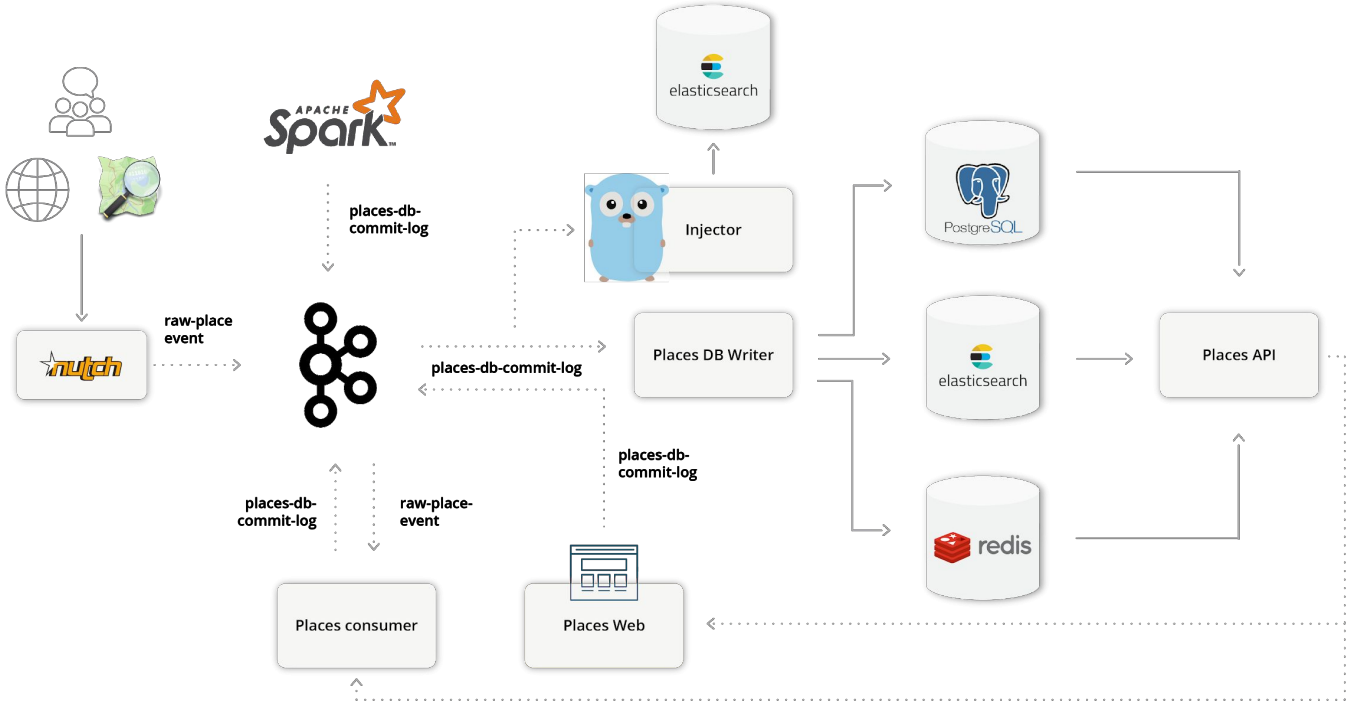
- ***More domain knowledge, more problems***
- **Web Server com dificuldades em picos**

Processamento off-line

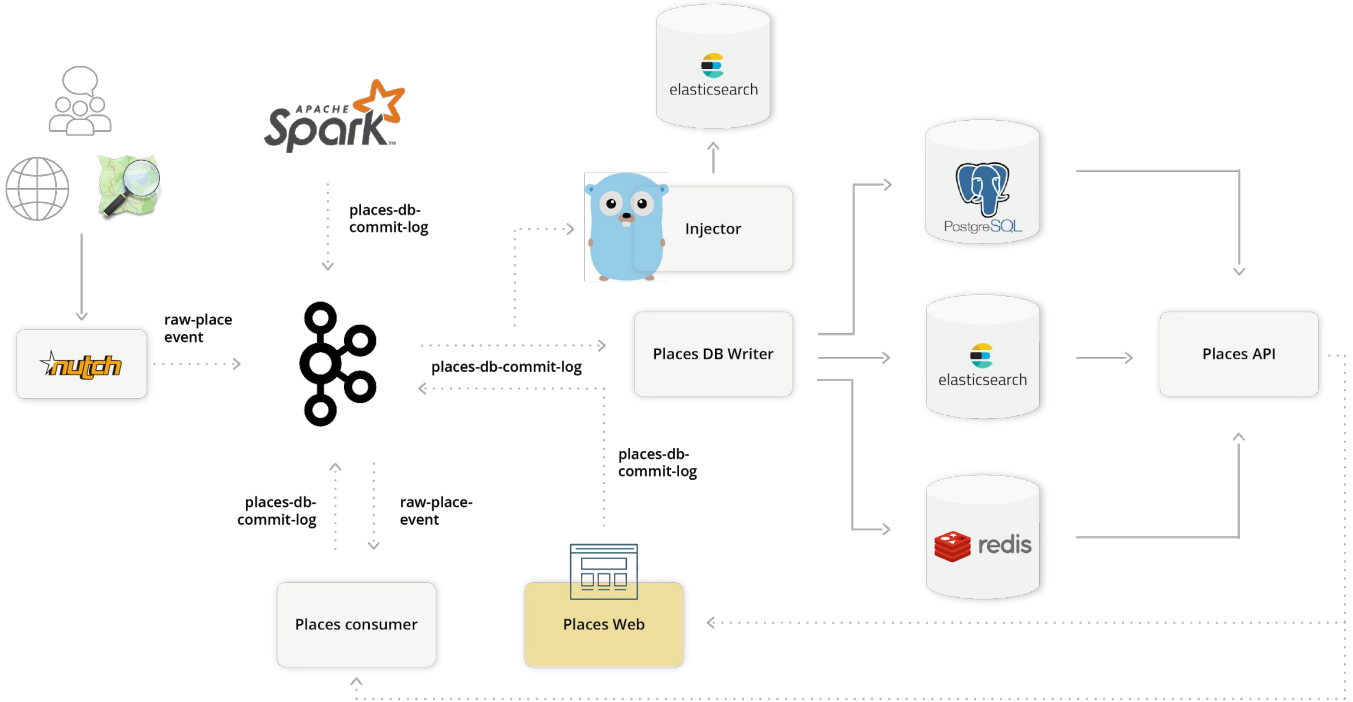
Apache Spark + Kafka

- Retira nossos limitantes de tempo
- Alinhamento com o requisito inicial
- O quão antes for feita essa parte, mais cedo será possível iterar sobre os dados

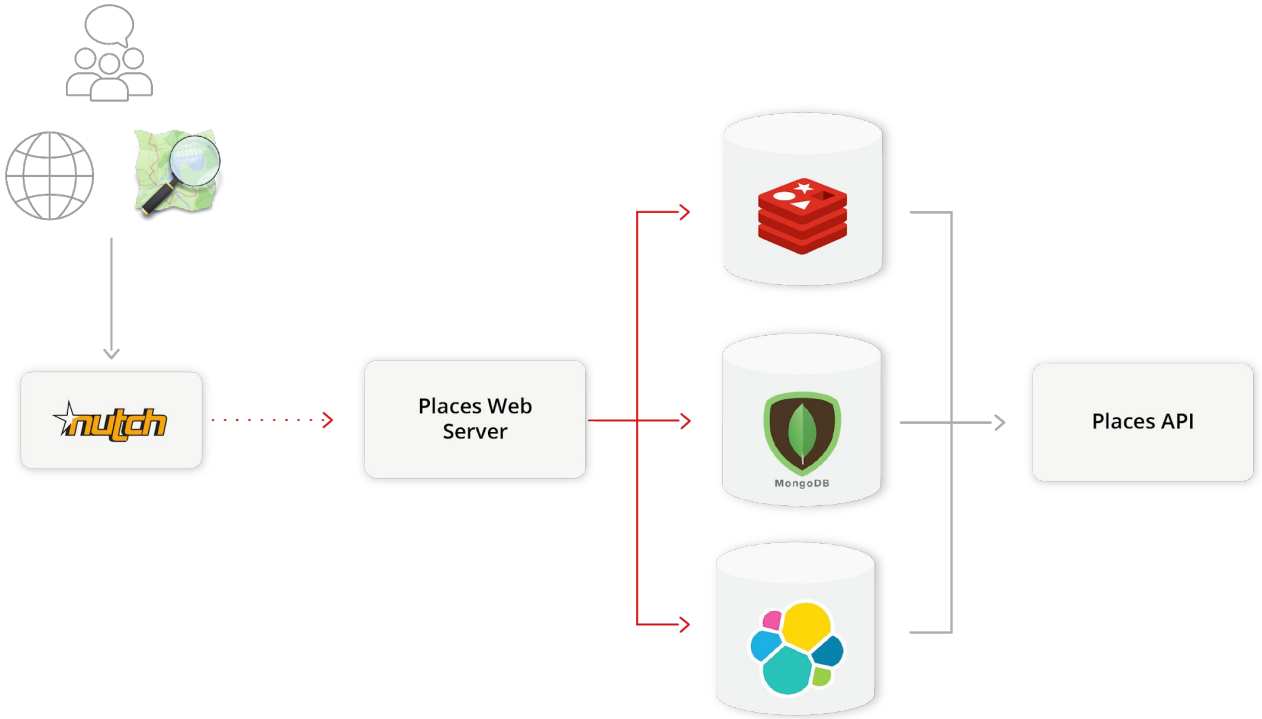
Versão Atual



Versão Atual

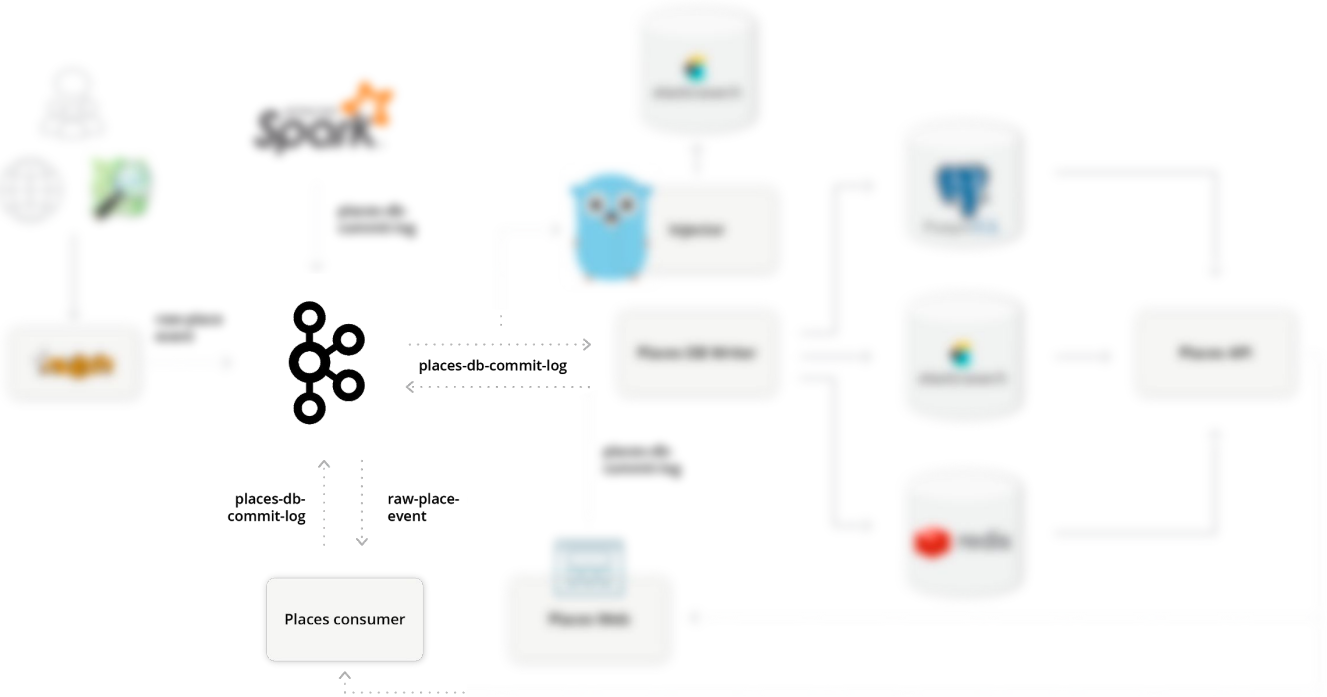


Remoção de Gargalos

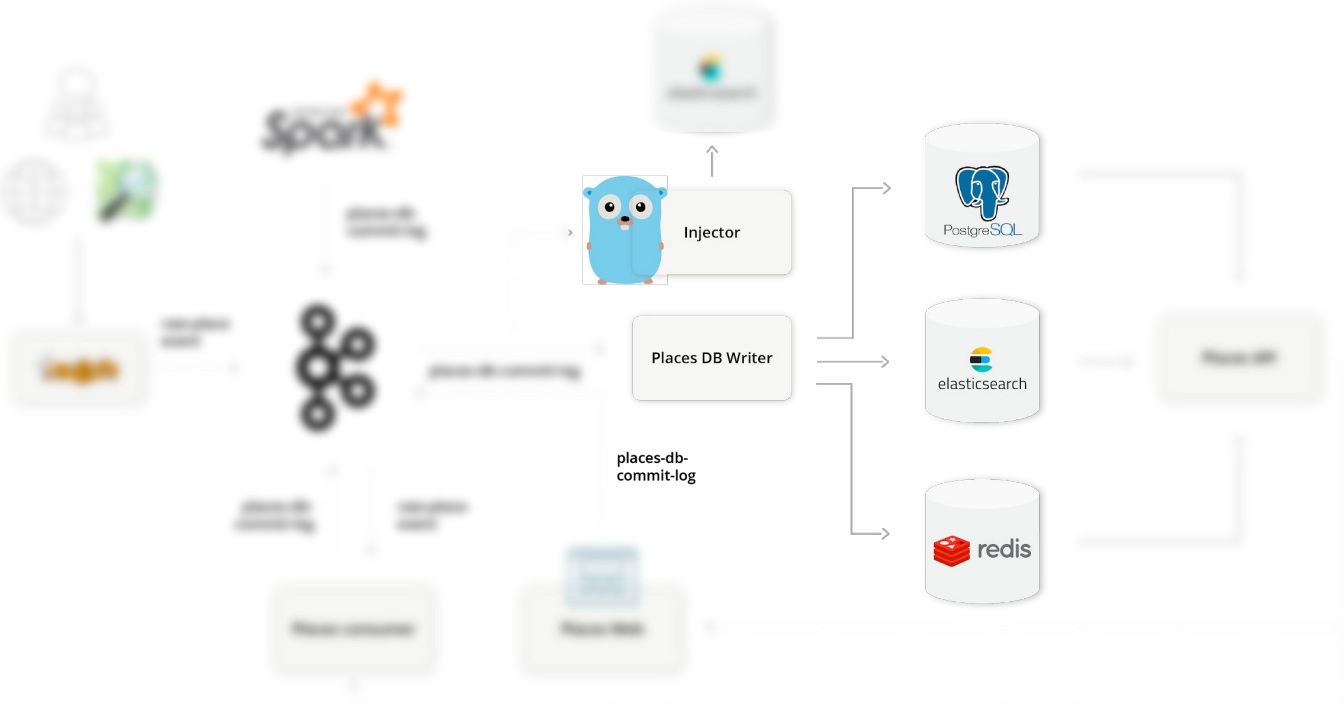


Remoção de Gargalos

Eventos


places-db-commit-log
places-db-commit-log
raw-place-event
Places consumer

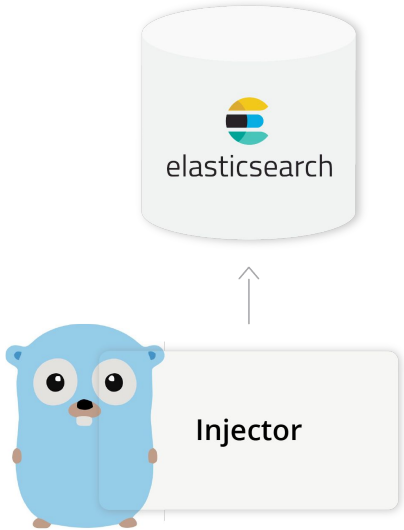
Eventos


Injector
PostgreSQL
Places DB Writer
elasticsearch
redis
places-db-
commit-log

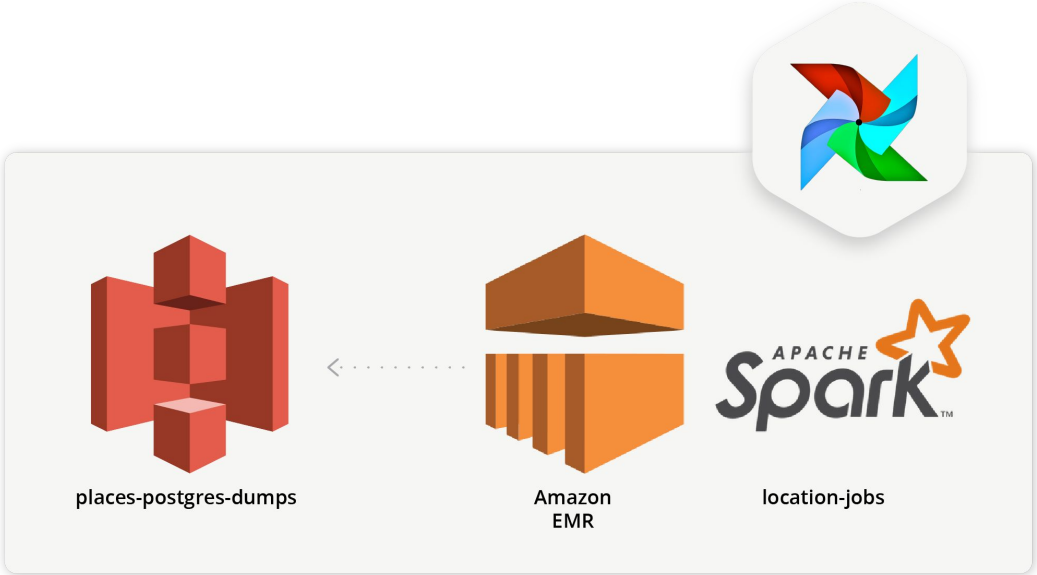
Eventos

- **Replicação de CRUD nos 3 bancos**
- **Reversibilidade**
- **Visibilidade - Crucial!**
    - ***Can’t improve what you cannot see***

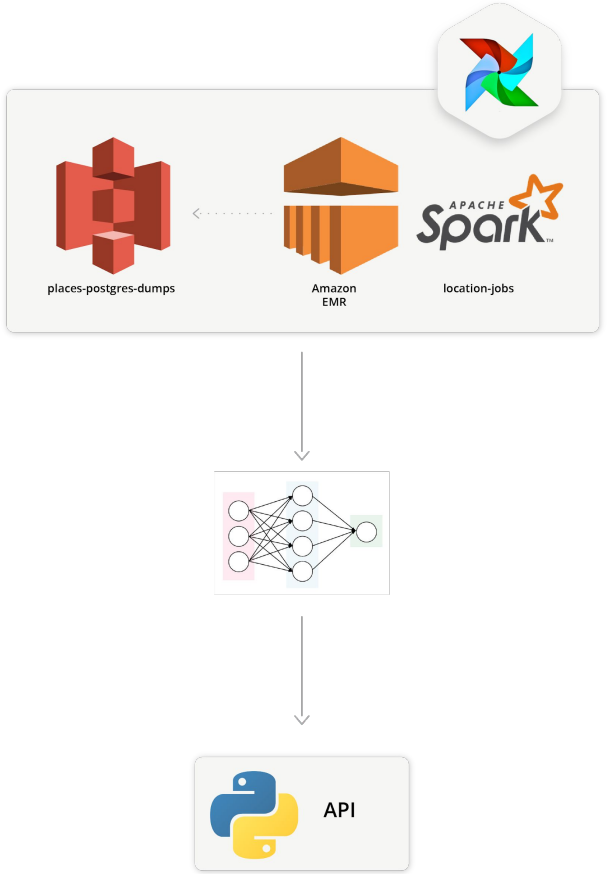
Visibilidade - Injector



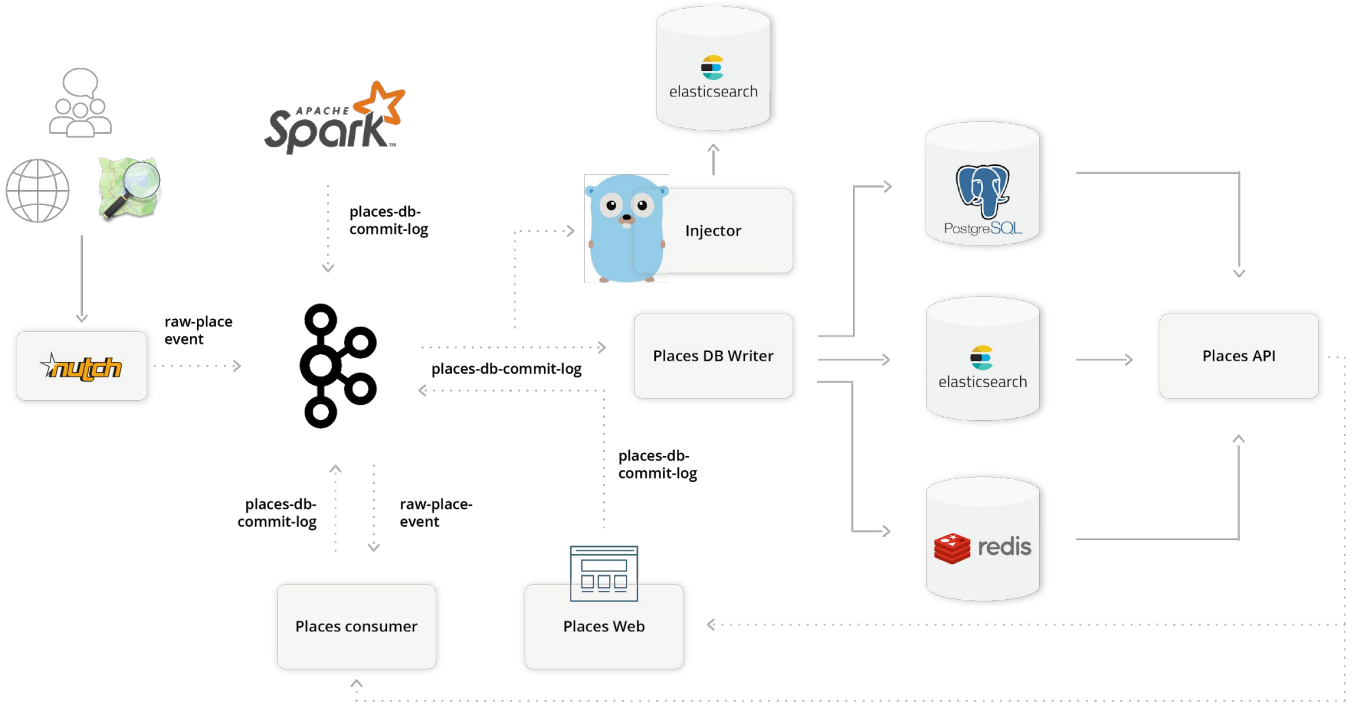
Orquestrando Jobs



Machine Learning

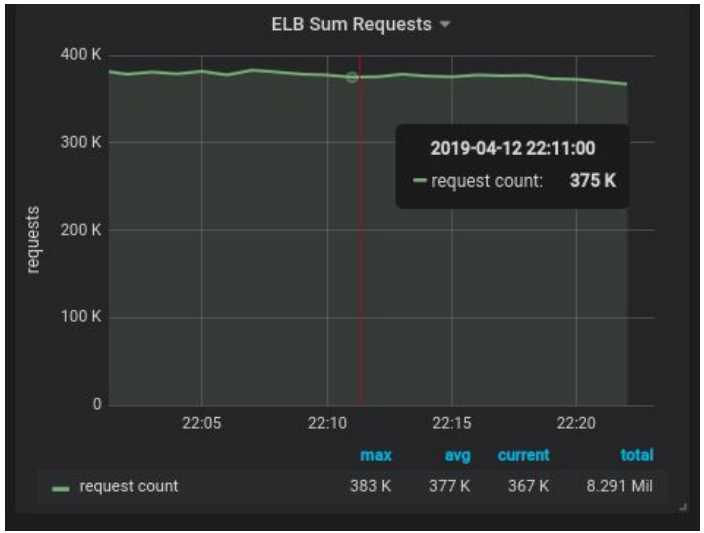


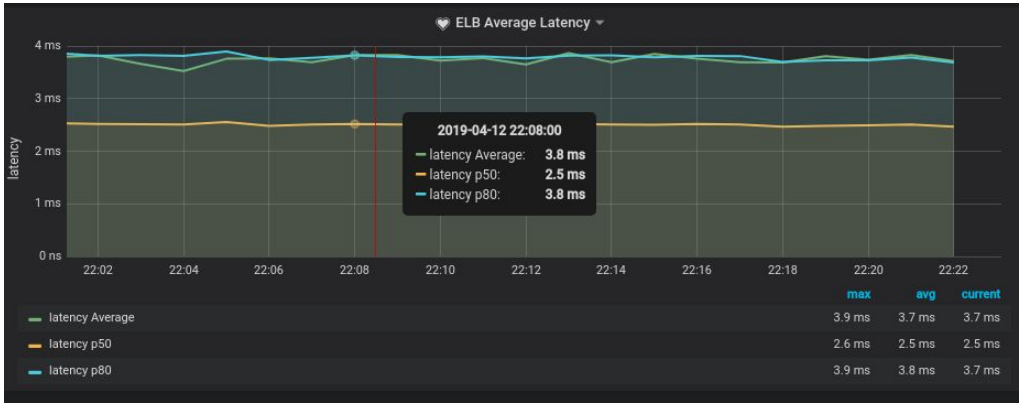
Versão Atual



Outcomes

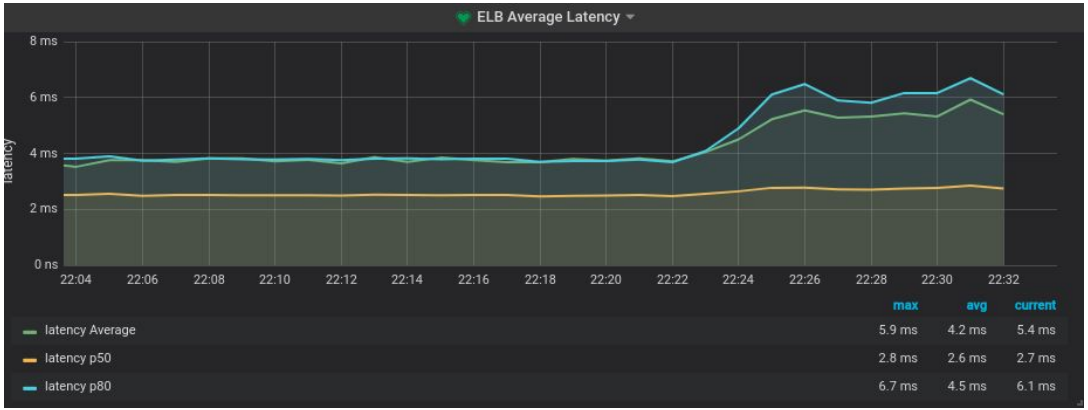
Escala


ELB Sum Requests
400 K
300 K
200 K
100 K
0
requests
22:05
22:10
22:15
22:20
2019-04-12 22:11:00
request count: 375 K
max avg current total
request count 383 K 377 K 367 K 8.291 Mil


ELB Average Latency
4 ms
3 ms
2 ms
1 ms
0 ns
latency
22:02
22:04
22:06
22:08
22:10
22:12
22:14
22:16
22:18
22:20
22:22
2019-04-12 22:08:00
latency Average: 3.8 ms
latency p50: 2.5 ms
latency p80: 3.8 ms
max avg current
latency Average 3.9 ms 3.7 ms 3.7 ms
latency p50 2.6 ms 2.5 ms 2.5 ms
latency p80 3.9 ms 3.8 ms 3.7 ms

Escala


Throughput
2019-04-12 22:24:00
read: 67.5 MB/s
write: 8 kB/s
22:18
22:20
22:22
22:24
22:26
22:28
22:30
22:32
22:34
min max avg current
8.2 MB/s 68.9 MB/s 30.0 MB/s 65.6 MB/s
3 kB/s 295 kB/s 64 kB/s 6 kB/s


ELB Average Latency
8 ms
6 ms
4 ms
2 ms
0 ns
latency
22:04
22:06
22:08
22:10
22:12
22:14
22:16
22:18
22:20
22:22
22:24
22:26
22:28
22:30
22:32
max avg current
latency Average 5.9 ms 4.2 ms 5.4 ms
latency p50 2.8 ms 2.6 ms 2.7 ms
latency p80 6.7 ms 4.5 ms 6.1 ms

Sandbox

- **10 jobs rodando semanalmente**
- **Time de BI extraindo insights diretamente dos nossos dados**
- **Em 1 mês, conseguimos iterar um dos nossos modelos (*record linkage*) 11 vezes**

Onde Erramos

You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future.

Steve Jobs

Pontos de Melhora

- **Forçamos demais o uso de algoritmos distribuídos**
- **Fizemos as frente on-line e off-line em silos**
    - **Atrasamos o benefício de *active learning***
- **Demoramos a implementar o ferramental para reverter ações**

Pontos de Melhora

- **Muito trabalho a se fazer ainda!**

inloco

vinicius.cousseau@inloco.com.br
inloco.com.br

Saiba mais em:
medium.com/inlocotech

Quer fazer parte do nosso time?
https://inloco.com.br/en/careers

Follow us

/inlocoglobal

@inlocoglobal

/inlocotech