

Escalando para o imprevisível

Iuri de Silvio



Quem sou eu

@iurisilvio

Engenheiro de Computação pelo ITA

10 anos de Python, Django, Flask, Bottle e outros

Pricez + Fast Shop + Black Friday

postmon.com.br com 100 mi requests/mês

ceps.io com 5 mi requests/mês

1 ano e meio de Buser



O que eu aprendi

Temos um monolito Django

A escala aumenta muito rápido

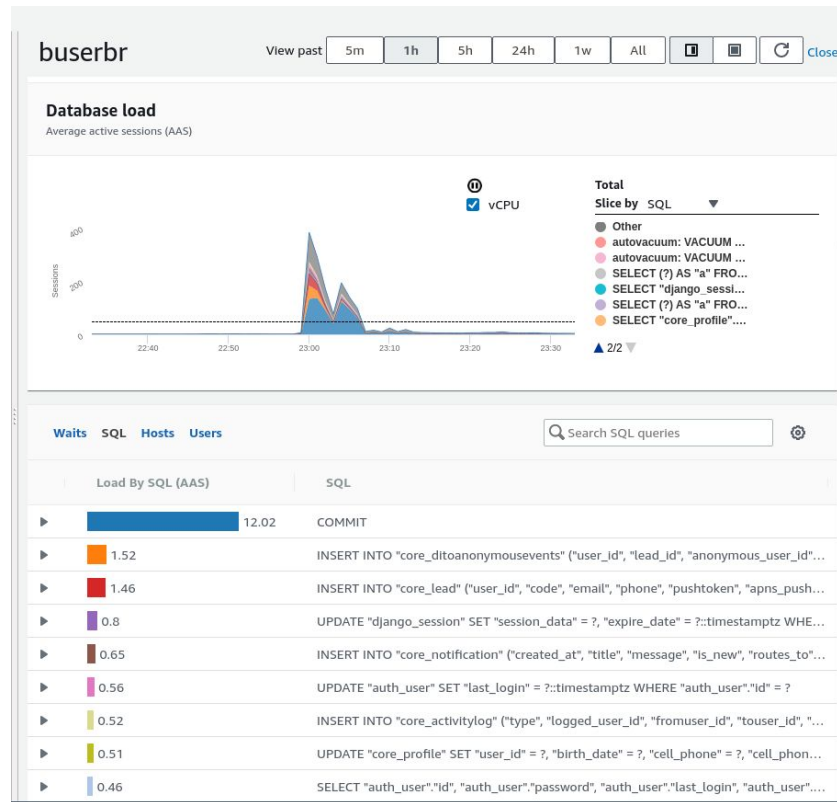
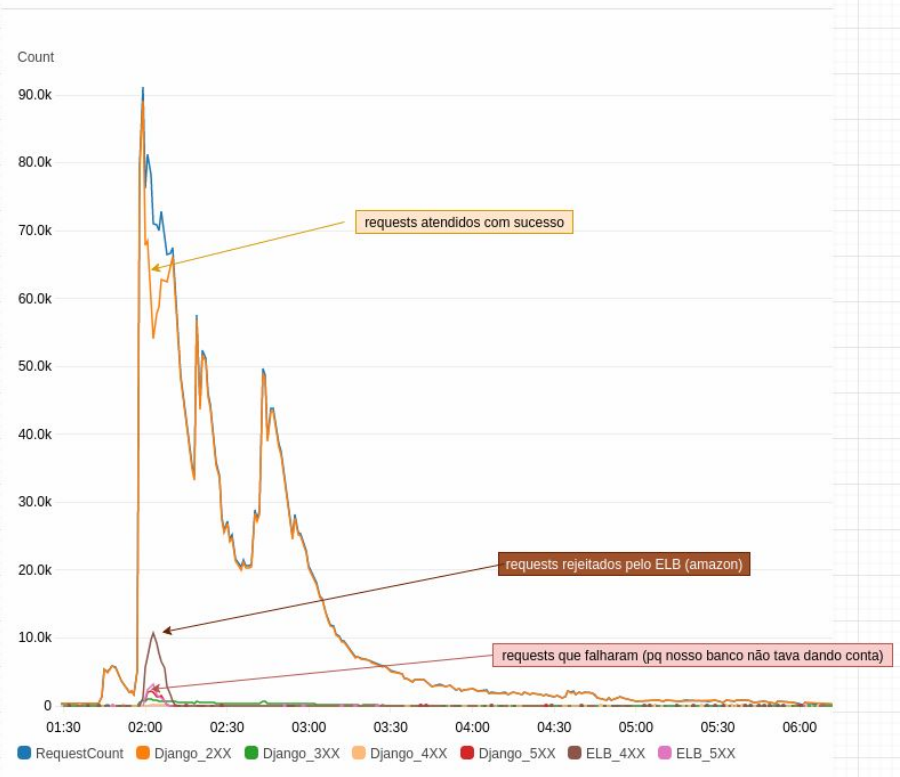
Decisões baseadas nas necessidades

Nunca trabalhei nesse tamanho antes

Também não sei como vai ser quando crescer mais

Falhas são aprendizado

ALB - Requests



Não pode ler do banco de dados

Cache de dados

Cache das Django sessions



Não pode escrever no banco de dados

Escrita tem que ser atrasada sempre que possível

Escrita tem que ser em batch (Celery não suporta isso)



Melhor nem falar com o Django

Cache no servidor web: conteúdo estático para usuários anônimos

Cache na CDN: conteúdo estático

The NGINX logo is displayed in a bold, green, sans-serif font. The letters are stylized, with the 'G' and 'I' having unique internal cutouts.

CDN: Content Delivery Network

AWS Load Balancer

ALB é muito mais versátil

ALB precisa de aquecimento para escalar

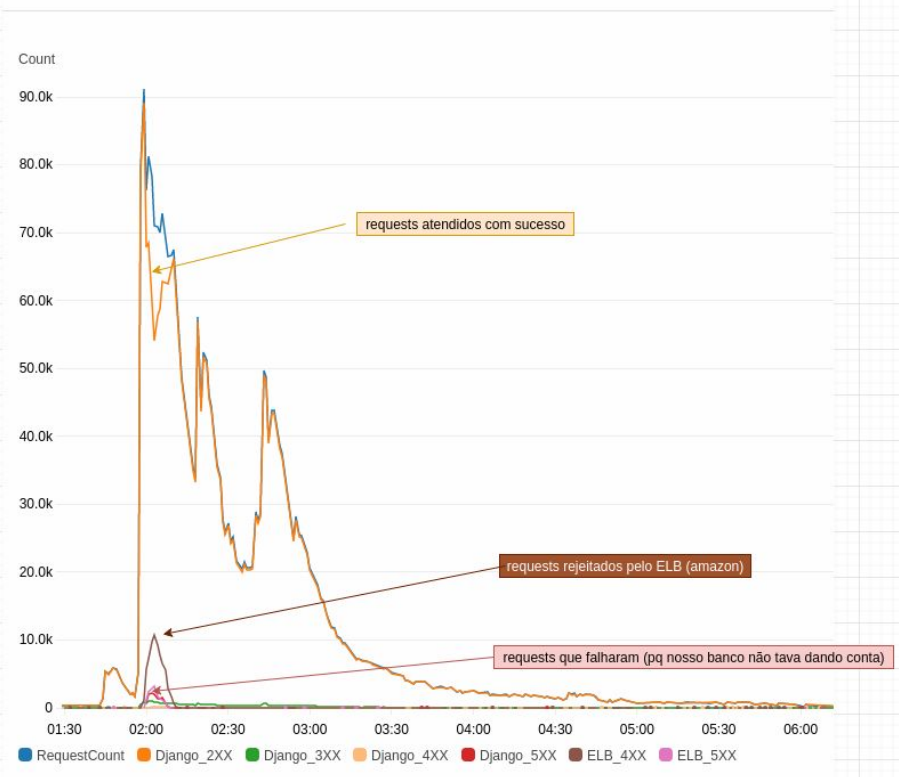
Setup do aquecimento na AWS é via ticket

NLB é mágico

ALB: Application Load Balancer

NLB: Network Load Balancer

ALB - Requests



AWS Auto Scaling

Ninguém te conta que é lento

Não conte com isso em eventos grandes

Super provisionamento é necessário

Lute para não depender de nada disso

E saber usar o Django

```
users = User.objects.all()[:100]
return [user.profile.phone for user in users]
```

```
SELECT "auth_user"."id",
       "auth_user"."password",
       "auth_user"."last_login",
       "auth_user"."is_superuser",
       "auth_user"."username",
       "auth_user"."first_name",
       "auth_user"."last_name",
       "auth_user"."email",
       "auth_user"."is_staff",
       "auth_user"."is_active",
       "auth_user"."date_joined"
FROM "auth_user"
LIMIT 100
```

100x

```
SELECT "core_profile"."id",
       "core_profile"."user_id",
       "core_profile"."phone",
       "core_profile"."cpf"
FROM "core_profile"
WHERE "core_profile"."id" = 3
LIMIT 1
```

E saber usar o Django

```
users = User.objects.select_related('profile').all()[:100]
return [user.profile.phone for user in users]
```

```
SELECT "auth_user"."id",
       "auth_user"."password",
       "auth_user"."last_login",
       "auth_user"."is_superuser",
       "auth_user"."username",
       "auth_user"."first_name",
       "auth_user"."last_name",
       "auth_user"."email",
       "auth_user"."is_staff",
       "auth_user"."is_active",
       "auth_user"."date_joined",
       "core_profile"."id",
       "core_profile"."user_id",
       "core_profile"."phone",
       "core_profile"."cpf"
FROM "auth_user"
LEFT OUTER JOIN "core_profile" ON (
    "auth_user"."id" = "core_profile"."user_id"
) LIMIT 100
```

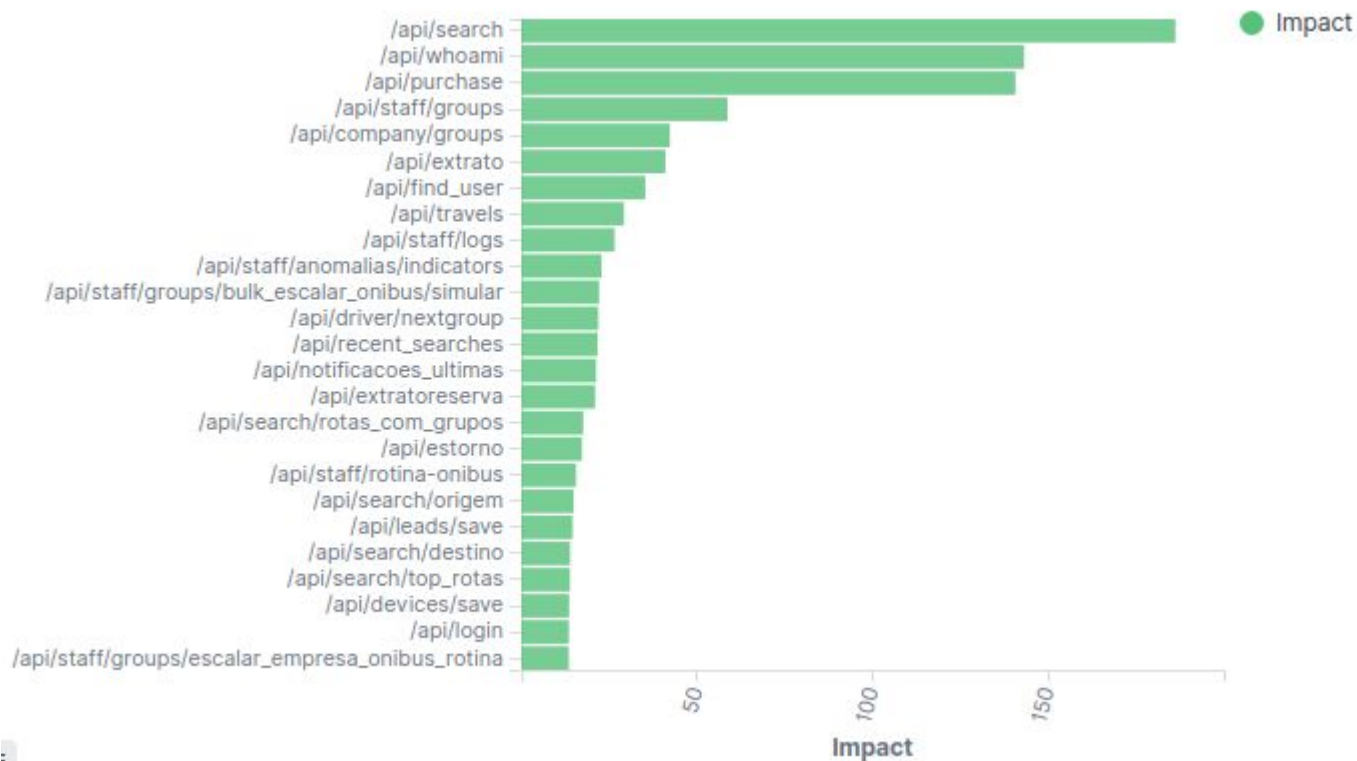
E saber usar o Django

```
phones = User.objects.values_list(
    ... 'profile__phone', flat=True
)[:100]
return list(phones)
```

```
SELECT
  ... "core_profile"."cell_phone"
FROM "auth_user"
LEFT OUTER JOIN "core_profile" ON (
  ... "auth_user"."id" = "core_profile"."user_id"
) LIMIT 100
```

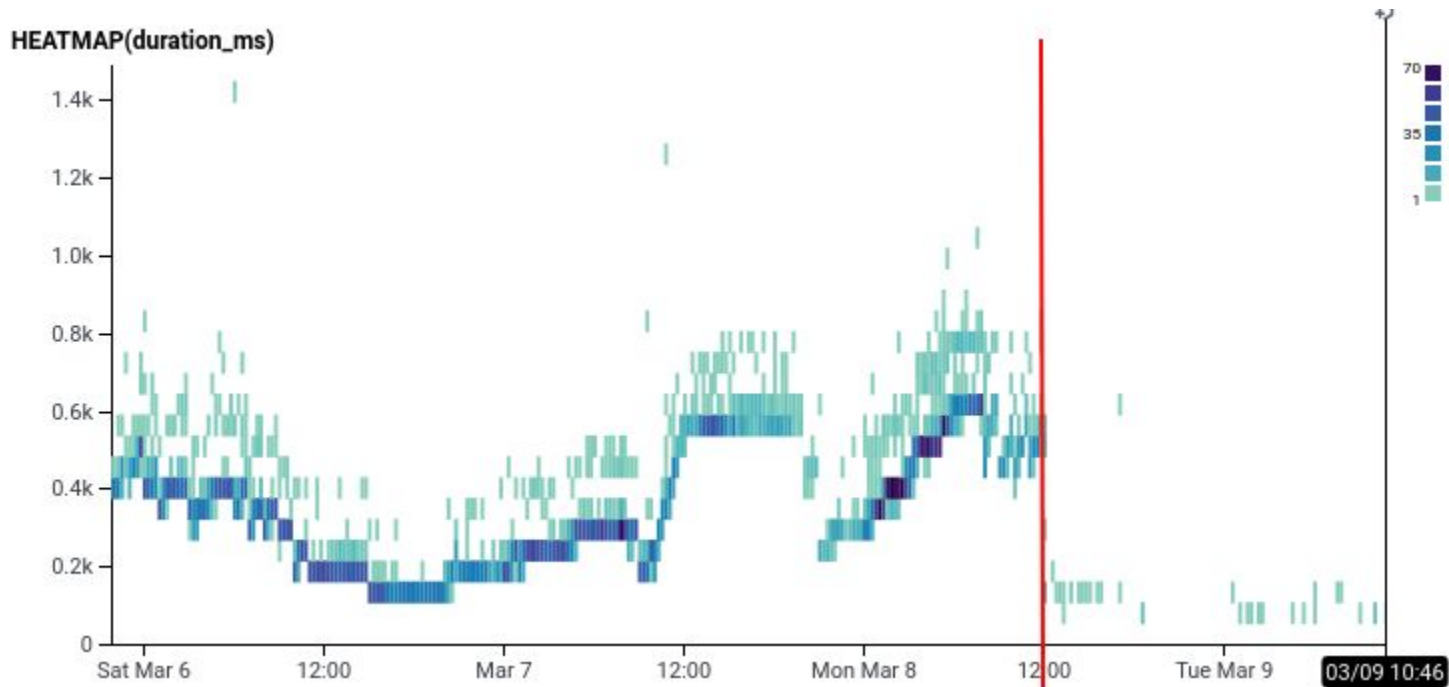
Sempre medir impacto

Soma de tempos de execução indica prioridade



Sempre medir impacto

A visão global também é importante



Observabilidade



11:30 às
12:05

Observability: Um olho no peixe e outro no gato em Python

Alex Coqueiro

Muitas organizações estão adotando arquiteturas de microserviço em python nesta jornada para a criação do mundo digital. Garantir a escalabilidade incorporando a visibilidade do que esta ocorrendo em produção de forma integrada é um dos grandes desafios levando em consideração a pluralidade de multi...

+



Futuro

Microserviços

Serverless

Q&A

 buser



Obrigado!

@iurisilvio
<https://iurisilv.io>

Estamos contratando

buser.com.br/carreiras