Reingeniería de la api de items

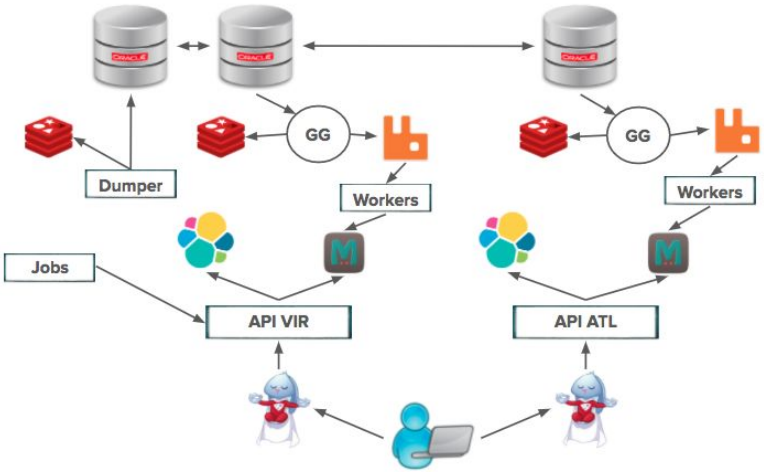
Decisiones arquitectónicas y lecciones aprendidas

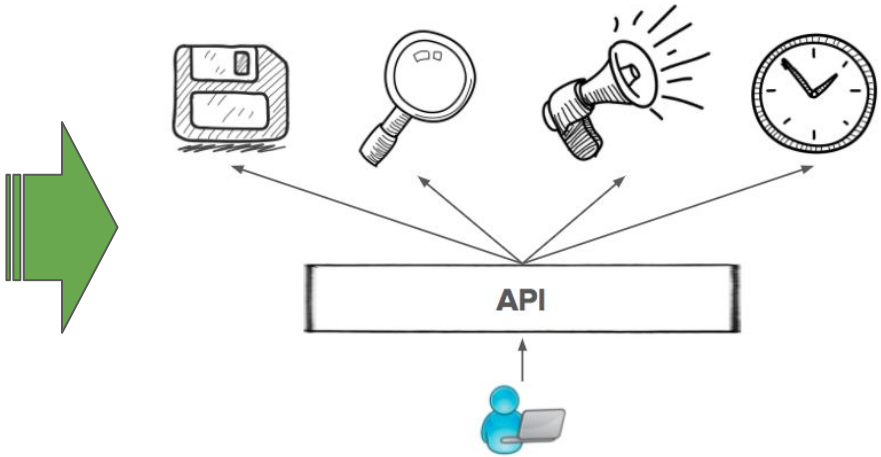
Gustavo Rodriguez

Agenda

- Qué nos planteamos antes de migrar una API
- Plan que seguimos en todas nuestras migraciones
- Arquitectura, caches y problemas asociados
- Manejo de consistencia
- Monitoreo y troubleshooting

Un poco de contexto

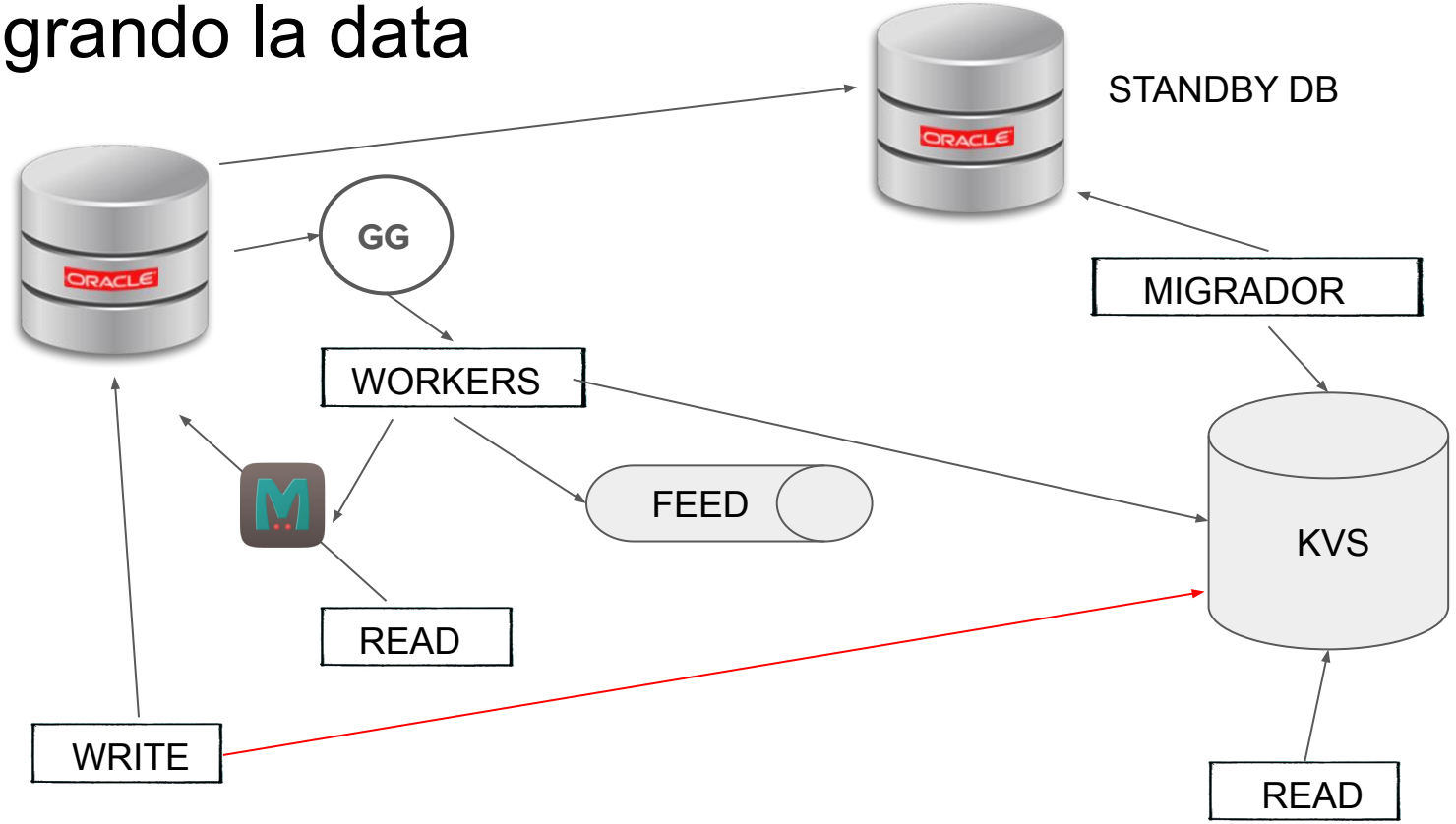




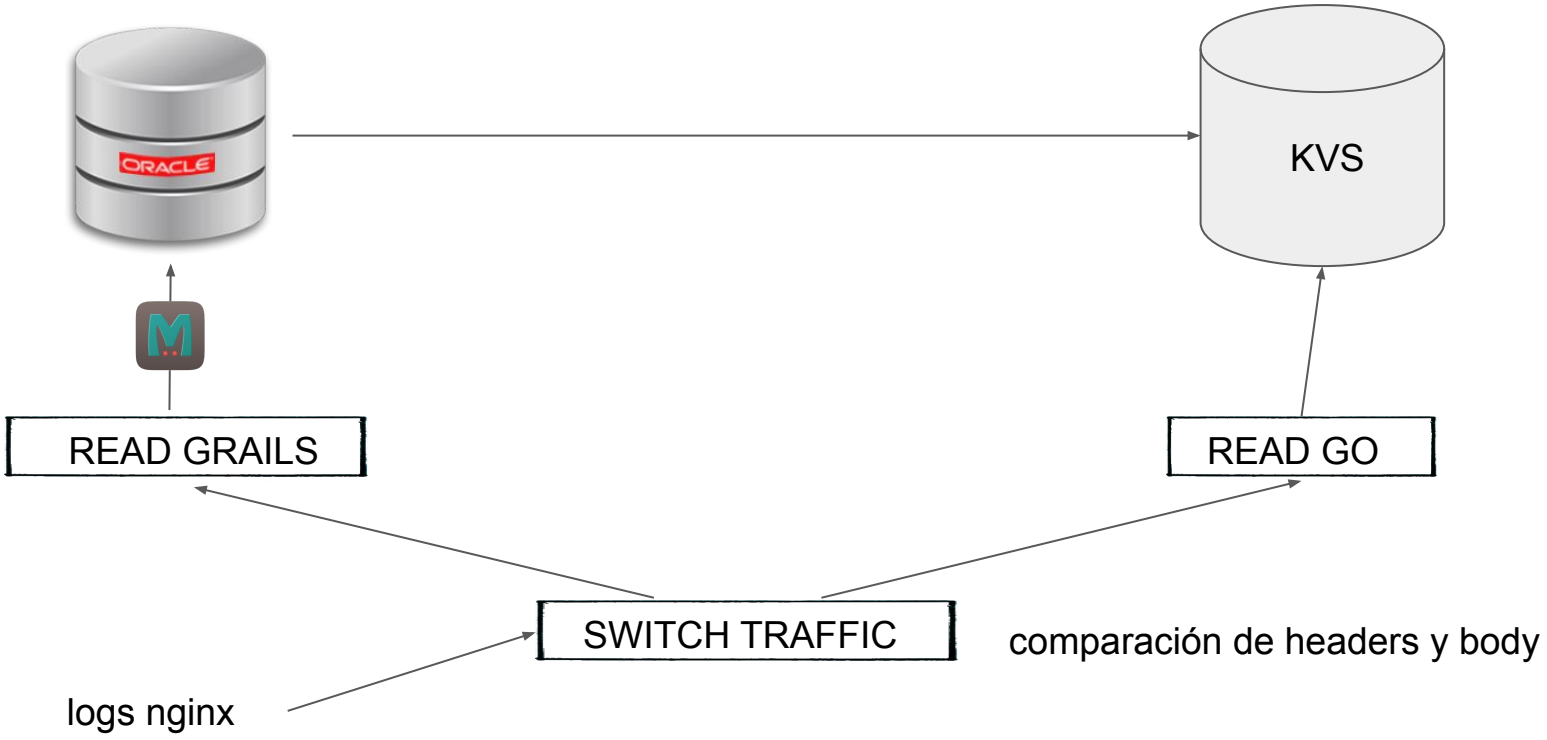
Desafios!

- Tráfico muy alto (>16M RPM)
- Siempre hay que tener un rollback
- Fuerte foco en la consistencia
- Tenemos que tener un excelente monitoreo
- Minimizar tiempos de respuesta
- Posibilidad de profilear
- Magias de mlapi (auth, jsonp, attributes,etc)
- Mucha historia en el código

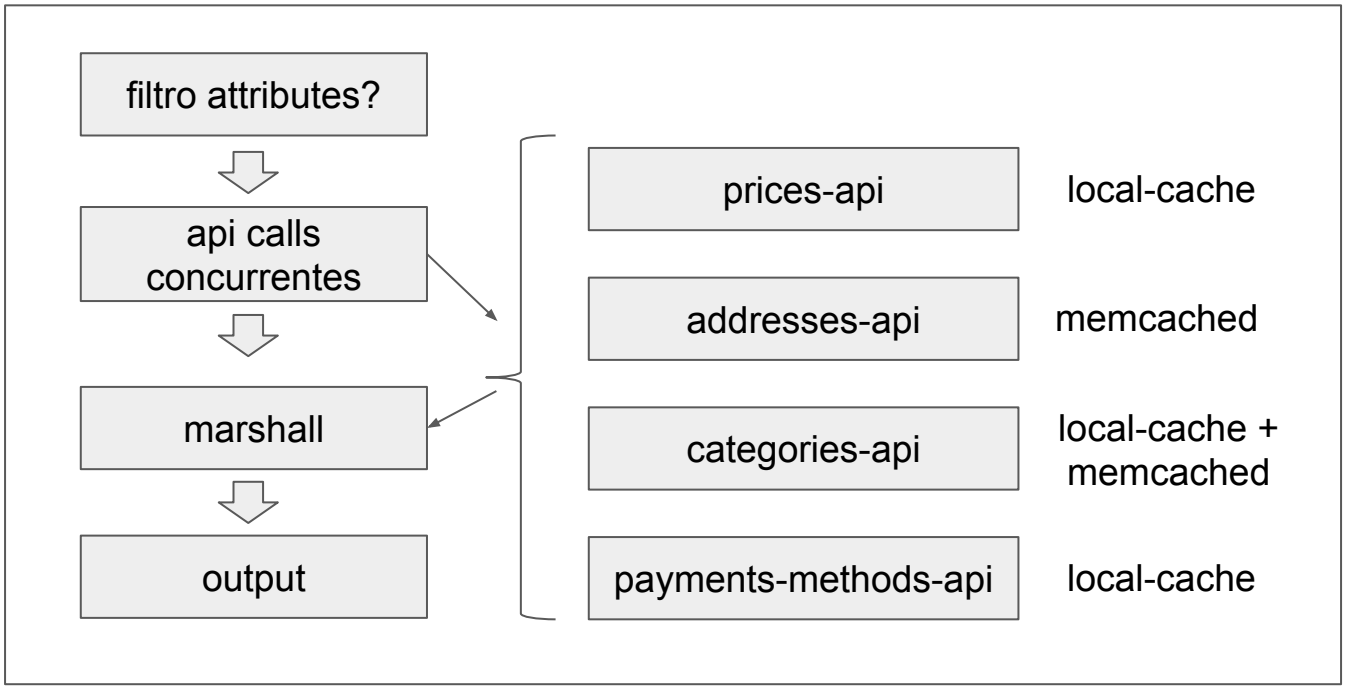
Migrando la data



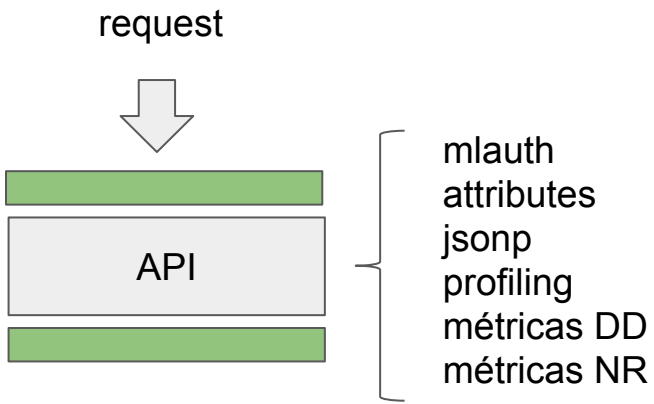
Verificando la consistencia y pruebas de carga



Marshaller Concurrente



Middlewares




```
router = mlhandlers.DefaultMeliRouter()
router.GET("/ping", controllers.Ping)
router.Run(":8080")
```

```
func Ping(c *gin.Context) {
        c.String(http.StatusOK, "pong")
}
```

Attributes: https://api.mercadolibre.com/items/MLA691795514?attributes=id,site_id,pictures.id

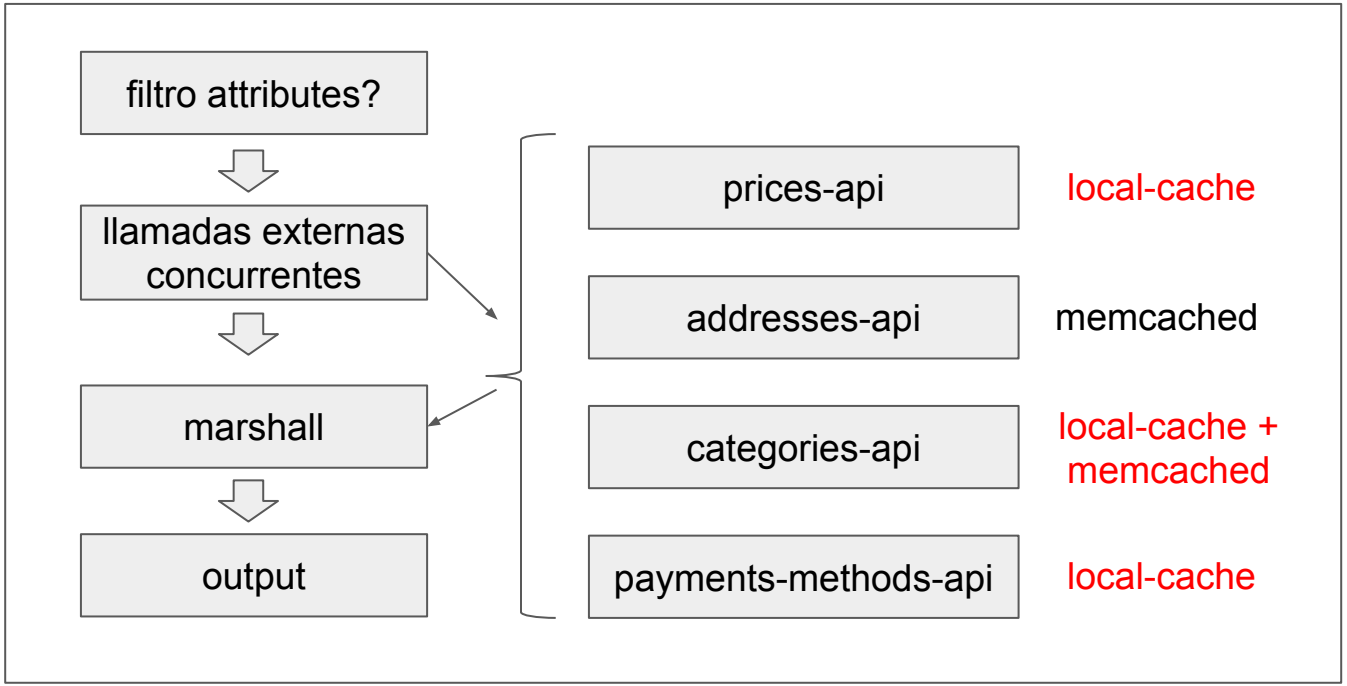
JsonP: https://api.mercadolibre.com/items/MLA691795514?callback=foo

Profiling: go tool pprof http://xxxxxxx/debug/pprof/heap|profile|block|trace

Monitoreo (DataDog + NewRelic)

- Tráfico total y por método (NR y DD)
- Response time (NR y DD)
- Slow transactions y percentiles (NR)
- Errores y stack traces (NR)
- Respuestas 2xx/4xx/5xx (DD)
- Uso de cpu y memoria (DD) + Goroutinas (NR)
- KVS throughput, errores y response time (DD)
- DS throughput , errores y response time (DD)
- Memcached miss / hit / timeout /error por target
- External api calls (throughput, retries, response time y status por target)
- Comportamiento de restclient pools (conexiones nuevas, reutlizadas, errores)

Puntos donde tuvimos que iterar



Resultados

- Response time: De 30 a 6 ms
- Se bancó aumento de tráfico que hubo en los últimos meses
- Se redujo 90% el hardware (simplificó deploys y administración)
- KVS demostró estabilidad y que escala horizontalmente
- Reducción de caches simplificó temas de consistencia
- Disminución de problemas de guardia
- Otras apis similares: locations, categories, questions, users, sites, etc

Gracias