

LE **GE**COM

les rencontres
geOrchestra

_2023



Saint-Mandé

CRESO & IGN Géoroom

—
30, 31 Mai &
1, 2 Juin

Amélioration des statistiques d'usage

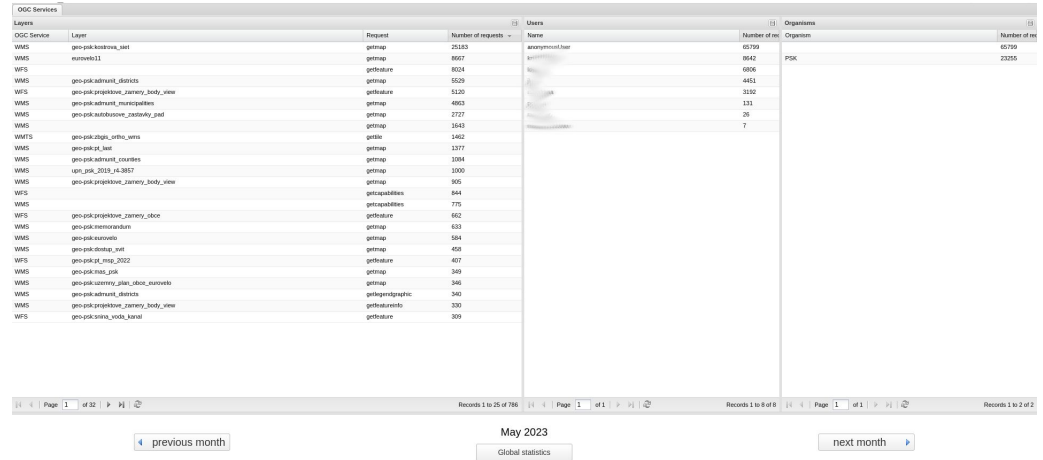
Jean Pommier

PSC geOrchestra / consultant indépendant

jean.pommier@pi-geosolutions.fr

Actuellement

- Stats OGC (geoserver)
- Stats utilisateur
- Appli Java
- Stockage PostgreSQL
- Charge beaucoup la BD applicative
- Pas de stats de fréquentation des autres services (GN, mapstore, mviewer etc)



OGC Services				Users		Organisations	
OGC Service	Layer	Request	Number of requests	Name	Number of requests	Organization	Number of requests
WMS	geo-pik-kadastre_sml	getmap	25183	anonymousUser	65799		65799
WMS	euronv121	getmap	8007	anonymousUser	8842	PSK	23255
WFS		getfeature	8024				6896
WMS	geo-pik-adminu1_disasters	getmap	6529				4451
WFS	geo-pik-projetkive_1primary_body_view	getfeature	5120				3152
WMS	geo-pik-adminu1_municipalities	getmap	4603				121
WMS	geo-pik-natlidiscove_2cadastre_sml	getmap	2727				26
WMS		getmap	1643				7
WMTS	geo-pik-stg1_ortho_wmts	gettile	1462				
WMS	geo-pik-st1_sml	getmap	1377				
WMS	geo-pik-adminu1_routes	getmap	1004				
WMS	ipn_psk_2023_v4_3857	getmap	1000				
WMS	geo-pik-projetkive_1primary_body_view	getmap	905				
WFS		getcapabilities	844				
WMS		getcapabilities	775				
WFS	geo-pik-projetkive_1primary_1boc	getfeature	662				
WMS	geo-pik-memorandum	getmap	633				
WMS	geo-pik-eurovets	getmap	584				
WMS	geo-pik-m1map_int	getmap	458				
WFS	geo-pik-st1_img_2022	getfeature	407				
WMS	geo-pik-net_psk	getmap	349				
WMS	geo-pik-adminu1_1g1n_1boc_eurovets	getmap	346				
WMS	geo-pik-adminu1_disasters	getfeatureinfo	340				
WMS	geo-pik-projetkive_1primary_body_view	getfeatureinfo	330				
WFS	geo-pik-smla_vnode_haral	getfeature	309				

Flux :

- écriture de logs en format spécifique
- Appli analytics stocke dans BD

Utiliser les logs d'accès

Fait sens car

- Tout passe par le SP / gateway
- Pas de logique spécifique à ajouter dans la gateway
- On maîtrise la structure des logs, quelle que soit l'infrastructure utiliséeFlexible dans le choix de la solution
- Suffisamment d'infos
- Prise en compte facile de nouvelles applis : si on ajoute des applis derrière la gateway, elle apparaîtra dans les access logs
- Prévoir peut-être une brique de filtre / formatage des logs pour faciliter l'indexation/aggregation (loki fournit, je crois)
- Enjeu : ne pas s'appuyer sur un stockage complet des logs (Elastic par exemple) -> peut vite faire exploser les besoins en stockage + mémoire pour indexation des logs

Analytics != monitoring

Monitoring

- granularité à la seconde
- court-terme (qq jours)
- objectifs
 - recevoir une alerte quand une mesure sort des bornes attendues
 - identifier des changements dans l'utilisation de ressources,
 - résoudre dysfonctionnement

Analytics

- granularité type = jour/semaine/mois
- long-terme (qq années)
- objectifs
 - fréquentations / appli ou page
 - identifier les points d'intérêt majeur / les ressources (pages) négligées
 - compter les téléchargements
 - justifier un travail, un investissement auprès des financeurs

Plusieurs pistes

- Stack Elasticsearch
- Matomo
- Loki
- Stockage en BD optimisée time (ex. timescaleDB) + appli maison

Discussions

- sur la mailing-list (sujet Amélioration de l'outil analytics)
- <https://github.com/georchestra/improvement-proposals/issues/5>

Elasticsearch

- Déjà utilisé pour GN 4
- Puissant, capable d'archiver et indexer le contenu des logs
- Mais:
 - Réputé extrêmement gourmand en ressources
 - Si index copieux, nécessite une configuration méticuleuse
 - pb de licence
 - La config pour GN est spécifique et ne sera peut-être pas simple à adapter

“that's more or less discouraged, as kibana is configured for GN indexes only, and there's some hairy url rewriting being done too...”

(<https://github.com/georchestra/improvement-proposals/issues/5>)

Matomo

- Déjà utilisé un peu partout en remplacement de Google Analytics
- Conçu pour des analytics de site web
- SAAS ou on-premise
- Simple à installer (PHP + mysql beurk)
- Modérément adapté, mais on doit pouvoir faire avec -> à voir comment indexer les requêtes OWS
- Fournit les dashboards (on peut en rajouter)
- Peut collecter les logs.
 - Script fourni (<https://github.com/matomo-org/matomo-log-analytics/>).
 - Sans doute nécessaire de remodeler un script maison
 - insérer un filtrage sur les logs pour nettoyer / restructurer certaines URLs(dédupliquer les requêtes OWS par exemple)

Matomo, suite

Consommation des ressources :

<https://fr.matomo.org/faq/on-premise/how-to-configure-matomo-for-speed/>

“A rough estimate of Matomo Mysql database size usage is approximately 1GB for every 5M page views. If your website tracks 100k page views per day (3M page views per month), you can expect a DB size of ~ 7GB after 1 year.”

Loki / grafana

- A la base, solution de monitoring (court terme)
- Réputé très performant et peu gourmand en ressources
- Indexe des “tags” sur les logs et travaille essentiellement sur les index
- On peut configurer la durée de rétention + une agrégation sur une plus faible granularité temporelle :
<https://grafana.com/docs/loki/latest/operations/storage/retention/#compactor>
- Fournit en bonus un usage de monitoring basé sur les logs

- KPIs



Recent requests

```

request for /a/618867854/alternative-to-throw-paper-in-bin-play-paper-ball-toss.html with HTTP status: 200
request for /a/1455895297/alternative-to-crazy-crazy-scatters-slots.html with HTTP status: 200
request for /?q=alternative=1258315958genre=finance&sort=relevance&page=2 with HTTP status: 200
request for /a/1804424024/alternative-to-waterfall-photo-frames-unlimited.html with HTTP status: 200
request for /a/498544723/alternative-to-jpegmini.html with HTTP status: 200
request for /a/1169418786/alternative-to-management-of-contrast-agent.html with HTTP status: 200
request for /a/586472181/alternative-to-philter.html with HTTP status: 200
request for /id/666258016/bag-nursery.html with HTTP status: 200
request for /a/1422558786/alternative-to-drinks-recipes-fruit-juice.html with HTTP status: 200
request for /a/1891684426/alternative-to-my-boo-town-pocket-world-game.html with HTTP status: 200
request for /a/129169888/alternative-to-bible-quizzer.html with HTTP status: 200
request for /a/145351293/alternative-to-ocussafe-evidence-collection.html with HTTP status: 200
request for /a/1452986718/alternative-to-wizz-make-new-friends.html with HTTP status: 200
request for /?q=alternative=1381615484genre=Entertainment&sort=relevance&page=4 with HTTP status: 200
request for /a/1452986718/alternative-to-wizz-make-friends.html with HTTP status: 301
request for /a/383981928/alternative-to-olympic-national-park-gps-map-navigator.html with HTTP status: 200
request for /a/128143277/alternative-to-escalade-parking-school-suv-driving-simulator.html with HTTP statu
    
```

- Request statistics over time



- Acquisition and Behaviour

Top HTTP Referrers		Top User Agents	
HTTP Referrer	Requests ↓	User agent	Requests ↓
https://www.google.com/	28493	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	284355
https://www.google.co.uk/	1346	Mozilla/5.0 (compatible; AhrefsBot/7.0; +http://ahrefs.com/robot/)	203662
https://duckduckgo.com/	670	Mozilla/5.0 (compatible; SemrushBot/7~b; +http://www.semrush.com/bot.html)	113188
https://www.google.ca/	666	Mozilla/5.0 (compatible; Seekport Crawler; http://seekport.com/)	81461

Loki / grafana : infrastructure

- Dockerisé mais aussi dispo sous forme d'applications binaires facilement exécutables -> s'adaptera sans souci
- Infra pas trop complexe pour débiter : promtail + loki + grafana
- Peut stocker ses données (logs notamment) sur S3 (file-based storage)
- Il existe des solutions pour les cas extrêmes (énormes volumes de logs) : Mimir, Thanos, M3... mais l'infra devient bien plus complexe (on ne devrait pas en arriver là)
- Ressources :
 - très léger en mémoire/processeur
 - stockage : Landry rapporte 14Go sur 2 ans, sur des logs mapserver/mapproxy (donc usages carto)

Loki / grafana : flexibilité

- On trouve directement des configs pour suivre des logs nginx de base
- Mais décortiquer les requêtes OGC sera plus délicat -> discriminer par service, namespace, layer
- Attribution des tags basée sur expressions régulières
- Définir des règles en fonction de l'appli visée : geoserver, geonetwork, mviewer etc
- Assez simple d'ajouter des règles si on ajoute des applis derrière le SP

Stockage en BD optimisée time

- Suggestion de Julien Sabatier
- Ingestion pourrait se faire via du log4j2 (ex. <https://github.com/georchestra/cadastrapp/blob/master/cadastrapp/src/main/resources/log4j2.properties>)
- Optimiser la BD: utiliser une solution type timescaleDB / influxDB optimisée pour donnée temporelles (partitionnement automatique, [aggrégation](#) et politique de rétention)
- Graphiques via Grafana

Et vous, qu'en pensez-vous ?

On en parle au code-sprint ?
Ou bien sur la [GIP#5](#)