



Amélioration des statistiques d'usage

Jean Pommier

PSC geOrchestra / consultant indépendant
jean.pommier@pi-geosolutions.fr

Mael Reboux

Rennes Métropole
m.reboux@rennesmetropole.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, mvviewer etc)

Flux :

- Security Proxy logge les accès OGC directement en BD Postgresql
- Appli analytics fait l'affichage et une agrégation des données (historique)

OGC Services

Layers

OGC Service	Layer	Request	Number of requests
VMS	geo-pok-kadastre_suit	getmap	25183
	cartes11	getmap	8607
WFS	geo-pok-kadastre_suit	getfeature	8024
	geo-pok-admuni_districts	getmap	5029
WFS	geo-pok-progressive_famery_body_view	getfeature	5120
	geo-pok-admuni_municipalities	getmap	4823
VMS	geo-pok-kadastre_suit	getmap	2727
	geo-pok-admuni_municipalities_suit	getmap	1643
WFS	geo-pok-admuni_municipalities_suit	getfeature	1462
	geo-pok-admuni_municipalities_suit	getmap	1377
VMS	geo-pok-admuni_municipalities_suit	getmap	1084
	cartes11_2013_14-3857	getmap	1000
WFS	geo-pok-progressive_famery_body_view	getmap	905
	geo-pok-progressive_famery_body_view	getcapabilities	844
VMS	geo-pok-progressive_famery_body_view	getcapabilities	775
	geo-pok-progressive_famery_body_view	getfeature	662
WFS	geo-pok-progressive_famery_body_view	getmap	633
	geo-pok-progressive_famery_body_view	getmap	584
VMS	geo-pok-progressive_famery_body_view	getmap	458
	geo-pok-admuni_municipalities_suit	getfeature	407
WFS	geo-pok-admuni_municipalities_suit	getmap	349
	geo-pok-admuni_municipalities_suit	getmap	346
VMS	geo-pok-admuni_municipalities_suit	getlegendgraphic	340
	geo-pok-progressive_famery_body_view	getfeature	330
WFS	geo-pok-progressive_famery_body_view	getfeature	309

Users

Name	Number of requests
anonymousUser	65789
anonymousUser	8007
anonymousUser	8024
anonymousUser	8029
anonymousUser	8120
anonymousUser	8683
anonymousUser	2727
anonymousUser	2843
anonymousUser	2462
anonymousUser	1377
anonymousUser	1084
anonymousUser	1000
anonymousUser	905
anonymousUser	844
anonymousUser	775
anonymousUser	662
anonymousUser	633
anonymousUser	584
anonymousUser	458
anonymousUser	407
anonymousUser	349
anonymousUser	346
anonymousUser	340
anonymousUser	330
anonymousUser	309

Organisms

Organism	Number of requests
PSK	65789
	8007
	8024
	8029
	8120
	8683
	2727
	2843
	2462
	1377
	1084
	1000
	905
	844
	775
	662
	633
	584
	458
	407
	349
	346
	340
	330
	309

Page 1 of 32

Records 1 to 25 of 76

May 2023

Global statistics

Page 1 of 1

Records 1 to 8 of 8

Page 1 of 1

Records 1 to 2 of 2

previous month

next month

Ca ne peut pas continuer comme ça ! ;o)

- On n'a de stats de fréquentation des autres services (GN, mapstore, mviewer etc). OGC seulement.
- Charge beaucoup la BD applicative
- Gateway remplacera bientôt le security proxy et n'assurera pas ces logs analytics en BD

Réflexions

- [Présentation au dernier geocom](#)
- [Plusieurs ateliers durant l'année](#)
- Community meeting Mars 2024



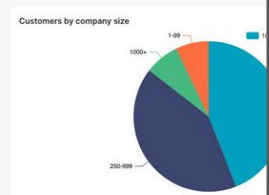
Proposition

- ❖ Essayer de limiter les briques maison (limiter la maintenance)
- ❖ Utiliser si possible des librairies utiles au-delà du sujet *analytics*
 - Visualisation graphique avec Apache Superset
 - Stockage/accès avec TimescaleDB (Postgresql)
- ❖ Collecte des infos avec outils maison

Visualisation

Apache Superset

- <https://superset.apache.org/>
- “Apache Superset™ is an open-source modern data exploration and visualization platform.”
- Outil de dashboarding, *Business Intelligence*
- Servira pour les analytics, *mais pas que*
- Correspond déjà à une fonctionnalité attendue par plusieurs plate-formes



All charts

Featured

Category

- Correlation
- Distribution
- Evolution
- Flow
- KPI
- Map
- Part of a Whole
- Ranking
- Table
- Other

Tags

Search all charts

Area Chart

Bar Chart

Bar Chart (legacy)

Big Number

Big Number with Trendline

Box Plot

Bubble Chart

Bubble Chart (legacy)

Bullet Chart

Calendar Heatmap

Chord Diagram

Country Map

deck.gl 3D Hexagon

deck.gl Arc

deck.gl Contour

deck.gl Geojson

deck.gl Grid

deck.gl Heatmap

deck.gl Multiple Layers

deck.gl Path

deck.gl Polygon

deck.gl Scatterplot

deck.gl Screen Grid

Event Flow

Funnel Chart

Gauge Chart

Generic Chart

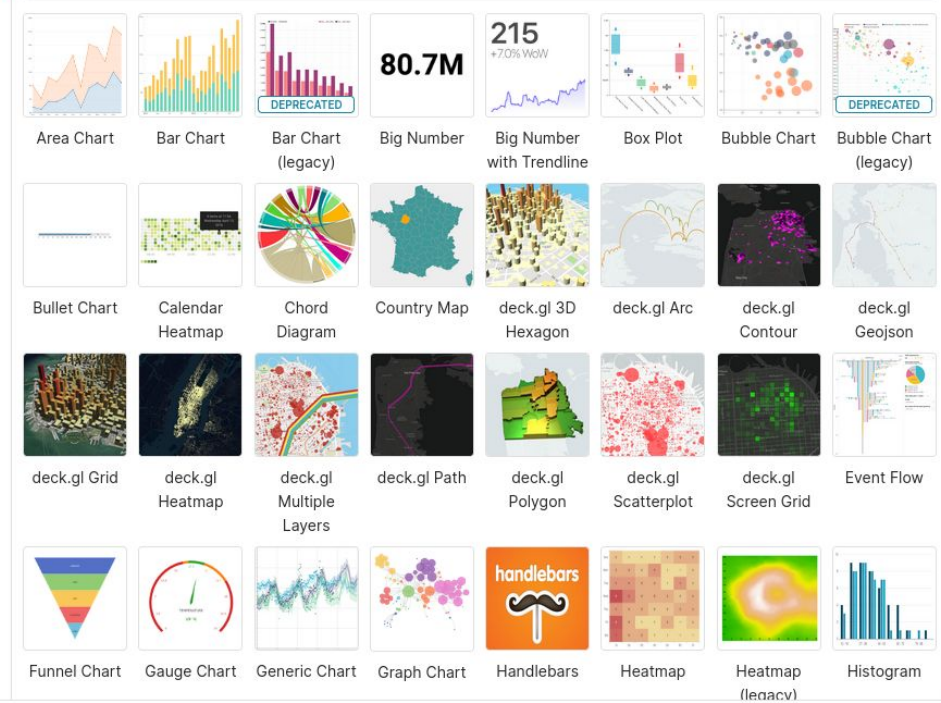
Graph Chart

Handlebars

Heatmap

Heatmap (legacy)

Histogram



Stockage, base de données

TimescaleDB [Timescale: PostgreSQL ++ for time series and events](#)

- C'est du postgresql (une extension)
- Support optimisé pour données temporelles, volumineuses
- Concepts nouveaux:
 - hypertable : optimisations sur une table classique, support de données volumineuses partitionnées automatiquement
 - time buckets : statistiques sur intervalles de temps
 - agrégats continus : simplification et aggrégation de données au fil du temps
 - période de rétention : effacement des données brutes au bout d'un certain temps
- On définira des paramètres par défaut
- Mais personnalisable à l'envi

DB

Une table de base commune à toutes les applis, structurée simplement :

```
CREATE TABLE analytics.logs
(
    time      timestamp with time zone NOT NULL,
    username  TEXT,
    org       TEXT,
    roles     TEXT,
    request_path TEXT,
    request_type TEXT,
    user_agent TEXT,
    app_name  TEXT,
    app_details json,
    response_time TEXT,
    response_size TEXT,
    http_code  TEXT
);
```

Où app_details est un champ JSON dont le contenu dépend de l'appli suivie.

Ex. Geoserver :

```
{
  "service": "wms",
  "version": "1.3.0",
  "request": "getmap",
  "format": "image/png",
  "transparent": "true",
  "workspace": "boundaries",
  "layers": "frontieres_nationales",
  "styles": "boundaries:lines_borders",
  "crs": "epsg:3857",
  "size": "2112x941",
  "bbox":
    "354259.98504683096,5862291.025534596,1645740.01495
    3169,6437708.974465404"
}
```

DB, suite

..et enrichie par TimescaleDB

```
SELECT create_hypertable('analytics.logs', by_range('time'));  
CREATE INDEX idx_username_time ON analytics.logs (username, time DESC);  
CREATE INDEX idx_org_time ON analytics.logs (org, time DESC);  
CREATE INDEX idx_app_time ON analytics.logs (app_name, time DESC);
```

DB, suite

Des agrégats continus (fonctionnalité TimescaleDB):

- Vues sur les données, pour dataviz
- S'appuient sur des vues matérialisées,
 - Regroupe les données suivant un `time_bucket`
 - Enrichies au fur et à mesure
- Possiblement en real-time

Ex. : liste des requêtes OGC

```
CREATE MATERIALIZED VIEW
analytics.logs_geoserver_ogc_daily
WITH (timescaledb.continuous) AS
SELECT time_bucket (INTERVAL '1 day', time) AS
bucket,
    app_details->>'service' AS ogc_service,
    app_details->>'layers' AS ogc_layer,
    app_details->>'request' AS ogc_request,
    count (*)
FROM analytics.logs
WHERE app_name='geoserver'
GROUP BY bucket, app_details->>'service',
app_details->>'layers', app_details->>'request'
ORDER BY bucket, ogc_layer, ogc_service,
ogc_request;
```

liste des requêtes OGC par jour

	bucket timestamp with time zone 	ogc_service text 	ogc_layer text 	ogc_request text 	count bigint 
1	2022-06-13 00:00:00+00	wms	chefslieux_hauptorte	describelayer	14
2	2022-06-13 00:00:00+00	wms	chefslieux_hauptorte	getmap	272
3	2022-06-13 00:00:00+00	wms	contourcrs_umrissork	describelayer	14
4	2022-06-13 00:00:00+00	wms	contourcrs_umrissork	getmap	272
5	2022-06-13 00:00:00+00	wms	departements_landkreis_kanton	describelayer	14
6	2022-06-13 00:00:00+00	wms	departements_landkreis_kanton	getmap	272
7	2022-06-13 00:00:00+00	wms	relief_etr89	describelayer	14
8	2022-06-13 00:00:00+00	wms	relief_etr89	getmap	272
9	2022-06-13 00:00:00+00	wms	rhin_rhein_etr89_2016	describelayer	14
10	2022-06-13 00:00:00+00	wms	rhin_rhein_etr89_2016	getmap	257
11	2022-06-14 00:00:00+00	wms	chefslieux_hauptorte	describelayer	16
12	2022-06-14 00:00:00+00	wms	chefslieux_hauptorte	getmap	256
13	2022-06-14 00:00:00+00	wms	contourcrs_umrissork	describelayer	16
14	2022-06-14 00:00:00+00	wms	contourcrs_umrissork	getmap	256
15	2022-06-14 00:00:00+00	wms	departements_landkreis_kanton	describelayer	16
16	2022-06-14 00:00:00+00	wms	departements_landkreis_kanton	getmap	256
17	2022-06-14 00:00:00+00	wms	relief_etr89	describelayer	16

Collecte des données

Collecte des données

On doit supporter au moins deux cas de figure :

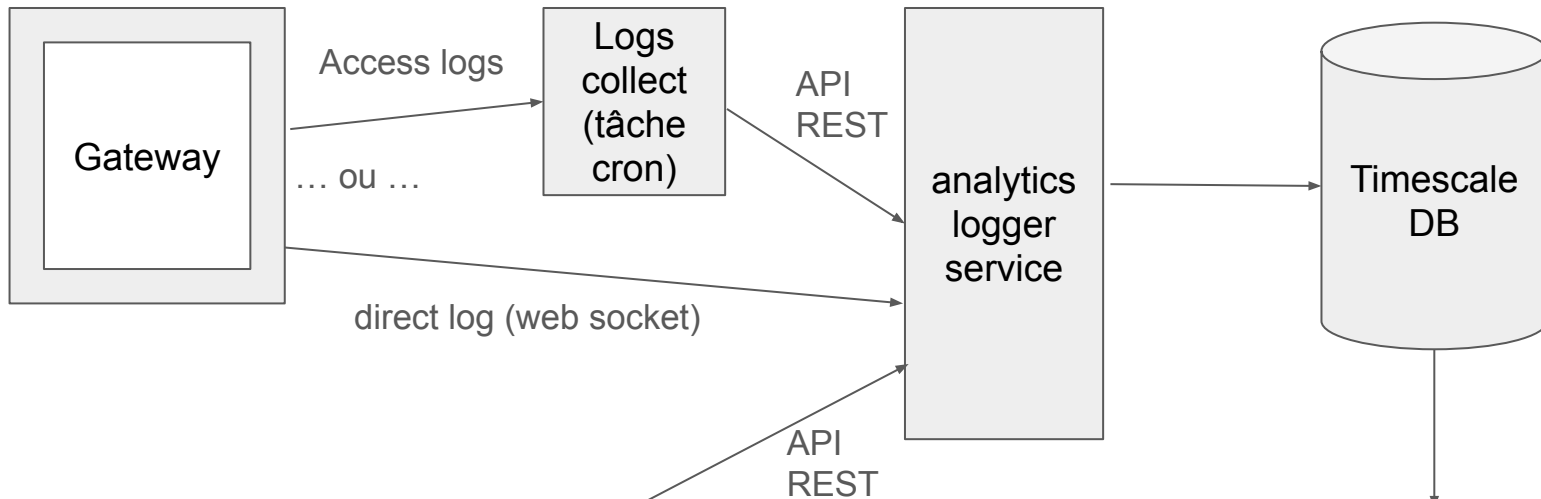
- Application backend, ex. serveur OGC
 - on intercepte les logs d'accès à l'entrée (SP, Gateway) -> on a l'URL, les infos d'authentification
 - on filtre et on structure les infos
 - on stocke en base
 - -> peu intrusif (pas de développement spécifique à chaque appli)
- Application frontend, type one-page application (mviewer, mapstore)
 - On n'a pas assez d'infos (ex <https://georchestra-127-0-1-1.traefik.me/mapstore/#/context/contexte1> on ne verra passer que <https://georchestra-127-0-1-1.traefik.me/mapstore/>)
 - Nécessite que l'appli frontend logge ses infos elle-même (appel d'un endpoint sur un service web analytics)

Collecte des logs d'accès

- Facile à configurer sur Gateway et SP pour des logs anonymes
- Nécessite des modifs spécifiques pour rajouter les infos utilisateur (username, org, roles)
- on ne collecte que pour les chemins pertinents
- un script filtre et restructure les informations avant ingestion en base (prototype sur https://github.com/georchestra/analytics/tree/main/georchestra_access_logs_ingester)
- En tâche cron sur un fichier de logs, ou bien en real-time via un web-socket et une config de logging adéquate

Application frontend, type one-page application

- Les requêtes web interceptées par le SP/gateway ne sont pas pertinentes
- L'application doit rapporter elle-même ses actions pertinentes
- => on fournit un service REST à laquelle l'app. peut fournir les infos d'usage
- Nécessite des modifs dans le code de l'appli (ajout de la logique de rapportage)



The screenshot shows the Apache Superset interface with the **geOrchestra analytics** dataset selected. The table displays the following data:

bucket	ogc_service	ogc_layer	ogc_request	count
2022-06-13	wms	chefdeleux_hauptorte	describeLayer	14
2022-06-13	wms	chefdeleux_hauptorte	getmap	272
2022-06-13	wms	contours_urbanisok	describeLayer	14
2022-06-13	wms	contours_urbanisok	getmap	272
2022-06-13	wms	departements_landkreis_kanton	describeLayer	14
2022-06-13	wms	departements_landkreis_kanton	getmap	272
2022-06-13	wms	relief_atrs89	describeLayer	14
2022-06-13	wms	relief_atrs89	getmap	272
2022-06-13	wms	rhrh_meln_atrs89_2016	describeLayer	14
2022-06-13	wms	rhrh_meln_atrs89_2016	getmap	257
2022-06-14	wms	chefdeleux_hauptorte	describeLayer	16

Dashboards
Apache Superset

Et maintenant ?

Next steps

- Déverrouiller les derniers points délicats
 - Intégrer proprement Superset dans geOrchestra
 - Collecter les infos utilisateurs dans les logs d'accès Gateway
- Le nerf de la guerre :
 - les finances
 - mais aussi : récupérer des logs d'accès copieux, pour tester la charge
- Votre avis ?
- Des volontaires ?

Merci