GDAL 2.1
Les nouveautés

Even Rouault
SPATIALYS

Plan

- Présentation générale de GDAL/OGR
- Activité communautaire
- GDAL 2.1 : les nouveautés !
- Directions futures potentielles
- Questions ?

Qui suis-je ?

- Contributeur à GDAL/OGR depuis 2007 et responsable du comité de pilotage depuis 2015.
- Contributeur à MapServer, QGIS
- Co-mainteneur: libtiff, libgeotiff, PROJ.4
- Fondateur de Spatialys, SS2L dans la géomatique. Forte expertise sur GDAL / MapServer

GDAL/OGR : Introduction

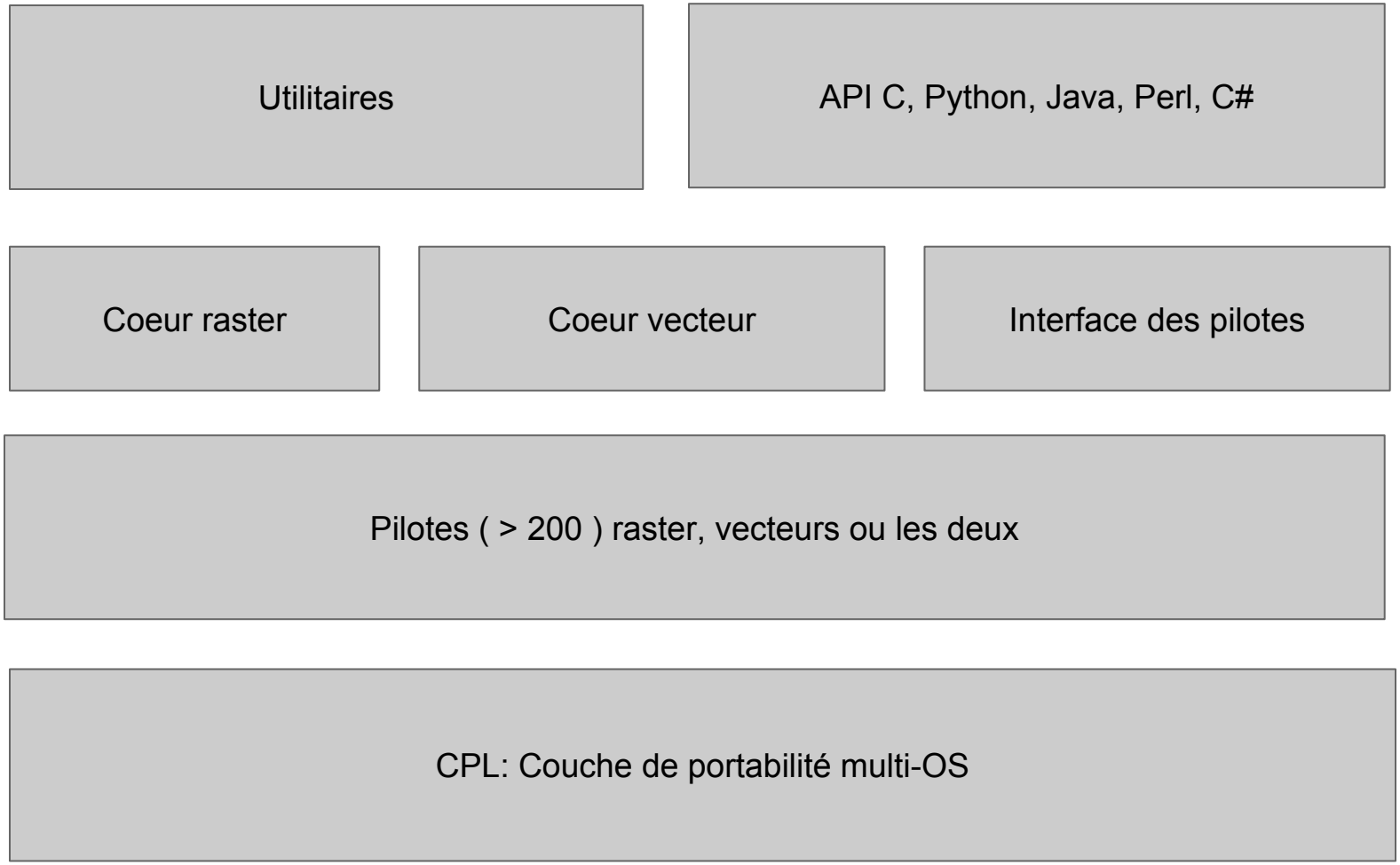
- GDAL? Geospatial Data Abstraction Library. Le couteau suisse du géomaticien
- Raster (GDAL) et Vecteur (OGR)
- Accès lecture/écriture à plus de 200 formats et protocoles de données (principalement) géospatiales.
- Utilisé très largement (Open Source et propriétaire): GRASS, MapServer, Mapnik, QGIS, PostGIS, OTB, SAGA, FME, ArcGIS, Google Earth…

  (> 100 http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal)
- Débuté en 1998 par Frank Warmerdam
- Project de l'OSGeo depuis 2008
- License open-source MIT/X (permissive)
- > 1M de ligne de code pour la bibliothèque, les utilitaires, ...
- > 150K lignes de test Python

Principales fonctionnalités

- Gère des jeux de données de taille arbitraire
- Bibliothèque écrite en C++ avec une API C
- Multi OS: Linux, Windows, MacOSX/iOS, Android, ...
- “Bindings” en Python, Perl, C#, Java,...
- Gestion des divers formats au travers de pilotes implémentant une interface commune
- Utilitaires de transformation, reprojection, extraction, mosaïquage, tuilage, interpolation, indexation …
- Gère les fichiers locaux, distants, compressés, en mémoire...

Architecture générale



Fonctionnalités Raster

- Gestion efficace des grandes images (tuilage, pyramides)
- Gestion des référencements classiques (transformation affine, points de contrôle, RPC)
- Cache de blocs
- Moteur de reprojection
- Algorithmes divers: rastérisation, vectorisation (création de polygones / contours), interpolation de pixels nuls, filtres

Formats Raster

- Images: JPEG, PNG, GIF, WebP, BPG ...
- Images géo-référencées: GeoTIFF, .img, NITF, ...
- Ondelettes: JPEG 2000, ECW, MrSID, ...
- RDBMS: Oracle Raster, PostGIS Raster, Rasdaman
- BDD portatives: Rasterlite, MBTiles, GeoPackage
- Web Service: WMS, WCS
- Radar: CEOS, Envisat
- Elevation: DTED, USGS DEM, SRTM HGT
- Conteneurs: HDF4, HDF5, NetCDF
- Autres: Geospatial PDF
- Spécifiques GDAL: mémoire, VRT (virtuel)

⇒ 144 Formats

Fonctionnalités vectorielles

- Modèle de “feature” et géométries basé sur OGC/ISO
- GEOS pour les opérations géométriques
- Moteur de reprojection
- Capacités SQL:
  - OGR SQL ou SQLite pour l’ensemble des formats
  - Passe-plat SQL pour RDBMS

Formats vectoriels

- SID: Shapefile, MapInfo, ESRI Personal/File Geodatabase
- CAO: DXF, DWG, DGN (pre-V8)
- RDBMS: PostGIS, Oracle, MySQL, Ingres, MSSQL, ODBC
- BDD portatives: SQLite/Spatialite, GeoPackage
- Echange: KML (*GDAL 1.11:reference implementation*) GML, GeoJSON
- Web Service: WFS, Fusion Tables, CartoDB, CouchDB, Cloudant, GME
- Formats nationaux: SDTS, NAS, NTF, TIGER/Line, Interlis, VFK, Edigeo, SOSI, SXF, MTK GML, RUIAN GML, INSPIRE Cadastral GML
- Non spatial : CSV, XLS, XLSX, ODS
- Spécifiques GDAL: mémoire, VRT (virtuel)

⇒ 86 formats

Activité communautaire

- 58 développeurs avec accès SVN
  - 19 actifs / 12 derniers mois + 65 contributeurs occasionnels
  - https://www.openhub.net/p/gdal
- 2237 inscrits à gdal-dev. 2538 messages / 12 derniers mois
- ~ 550 tickets créés / 12 derniers mois (6500 au total). ~600 ouverts
- 1 étudiant GSoC en 2015. 2 en 2016

GDAL/OGR 2.0 en résumé

- V2.0.0: juin 2015
- 10 RFCs implémentées dans le cycle 2.0 dont:
  - Unification de GDAL et OGR au niveau des drivers et datasets
  - Noyaux de ré-échantillonage bilinéaires, bicubiques sans reprojection
  - Géométries curvilignes
  - Gestion des entiers 64 bits
- 11 nouveaux drivers dont:
  - GeoPackage Raster
  - Gestion complète de GeoPackage Vecteur

GDAL/OGR 2.1

- Sortie le 2 mai 2016
- 4461 “commits” (total depuis 1998: 30689)
- 6 RFCs implémentées dans le cycle 2.1
- 7 nouveaux pilotes raster
  - CALS Type 1 : lecture seule. format historique d’archivage
  - IBM DB2 : lecture/écriture. raster tuilés
  - ISCE : lecture seule. format utilisé par le JPL dans osn Interferometric SAR Scientific Computing Environment
  - MRF : Meta Raster Format. lecture/écriture. Développé par NASA Global Imagery Browse Services.
  - SAFE: lecture seule. produits ESA Sentinel-1 (SAR)
  - SENTINEL2: lecture seule. produits ESA Sentinel-2 L1B/L1C/L2A
  - WMTS : lecture seule. client de services de tuiles OGC Web Map Tile Service

GDAL/OGR 2.1

- 5 nouveaux pilotes vecteurs
  - AmigoCloud: lecture/écriture. plateforme “geospatial as a service”
  - IBM DB2: lecture/écriture
  - MongoDB : lecture/écriture. BDD “no-SQL” avec capacités spatiales.
  - netCDF: lecture/écriture. Points et profils verticaux conformément aux convention CF (Climate and Forecast)
  - VDV: lecture/écriture. Modèle et format de données de transport public (concurrent de GTFS). Spécialisation pour le format ITF autrichien.

GDAL/OGR 2.1

- Pilotes existants améliorés :
  - CSV: ajout de l'édition complète (modification & suppressions d' enregistrements, par réécriture du fichier)
  - GeoJSON: édition complète + préservation des extensions
  - ElasticSearch: ajout du mode lecture. Gestion de l'ensemble des géométries en écriture
  - MBTiles: ajout de l'écriture de tuiles raster
  - PDF: ajout d'un nouveau backend. PDFium (license BSD)
  - PLScenes: gestion de l'API V1 de Planet Labs (catalogue de données)
  - VRT(raster): pan-affinage (pan-sharpening) à la volée
  - GeoTIFF: possibilité d'activer la compression parallélisée (utile pour DEFLATE)

GNM

RFC 48: Modélisation de réseaux (GNM / Geographical networks model)

- Création/suppression de réseaux
- Création de la topologie du réseau à partir de données spatiales (manuel/automatique)
- Obtention des connections
- Ajout/retrait de couches et objets au réseau
- Définition d'une logique métier (règles d'autorisation/refus de connection entre objets de différentes couches)
- Différentes méthodes d'analyse de réseau (plus court chemin, connectivité)
- Utilitaires: gnmmanage, gnmanalyze
- Issue des travaux GSoC 2014

GNM

points.csv:

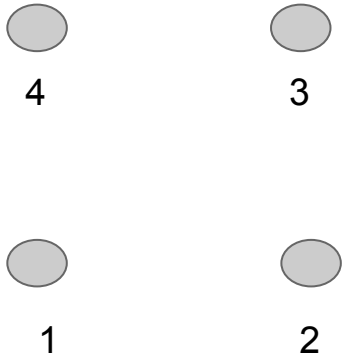
id,WKT

1,"POINT(2 49)"

2,"POINT(2.1 49)"

3,"POINT(2.1 49.1)"

4,"POINT(2 49.1)"



$ gnmmanage create -f gnmfile -t_srs EPSG:4326 mon_reseau

$ gnmmanage import points.csv mon_reseau

Layer successfully copied from points.csv and added to the network at mon_reseau

GNM

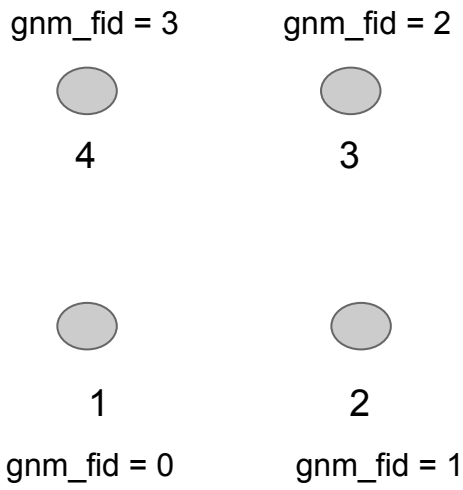
$ ogrinfo mon_reseau points

OGRFeature(points):0
  gnm_fid (Integer64) = 0
  blocked (Integer) = 0
  id (String) = 1
  POINT (2.0 49.0)

OGRFeature(points):1
  gnm_fid (Integer64) = 1
  blocked (Integer) = 0
  id (String) = 2
  POINT (2.1 49.0)

OGRFeature(points):2
  gnm_fid (Integer64) = 2
  blocked (Integer) = 0
  id (String) = 3
  POINT (2.1 49.1)

OGRFeature(points):3
  gnm_fid (Integer64) = 3
  blocked (Integer) = 0
  id (String) = 4
  POINT (2.0 49.1)



GNM

lines.csv:

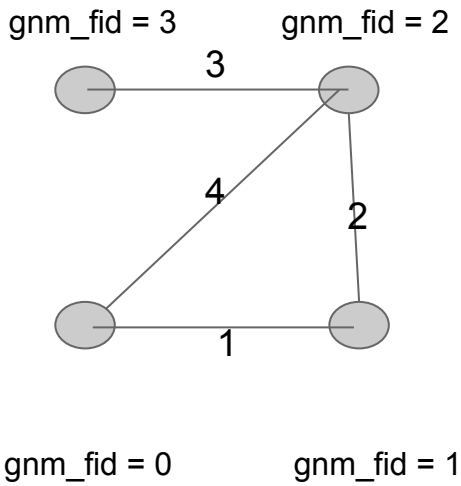
id,WKT

1,"LINESTRING(2 49,2.1 49)"

2,"LINESTRING(2.1 49,2.1 49.1)"

3,"LINESTRING(2.1 49.1,2 49.1)"

4,"LINESTRING(2 49,2.1 49.1)"



$ gnmmanage import lines.csv mon_reseau

Layer successfully copied from lines.csv and added to the network at mon_reseau

GNM

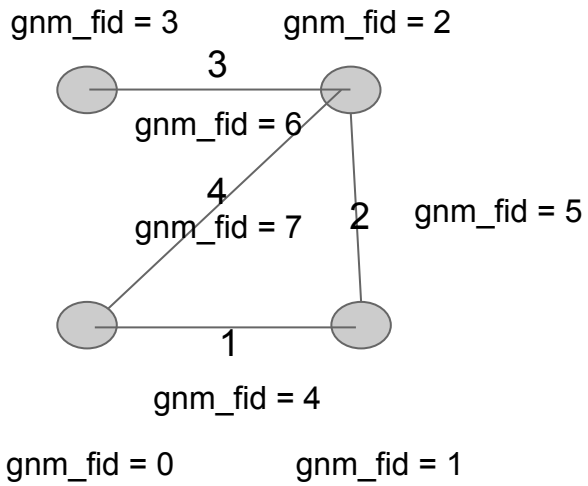
$ ogrinfo mon_reseau lines

OGRFeature(lines):0
  gnm_fid (Integer64) = 4
  blocked (Integer) = 0
  id (String) = 1
  LINESTRING (2 49,2.1 49.0)

OGRFeature(lines):1
  gnm_fid (Integer64) = 5
  blocked (Integer) = 0
  id (String) = 2
  LINESTRING (2.1 49.0,2.1 49.1)

OGRFeature(lines):2
  gnm_fid (Integer64) = 6
  blocked (Integer) = 0
  id (String) = 3
  LINESTRING (2.1 49.1,2.0 49.1)

OGRFeature(lines):3
  gnm_fid (Integer64) = 7
  blocked (Integer) = 0
  id (String) = 4
  LINESTRING (2 49,2.1 49.1)



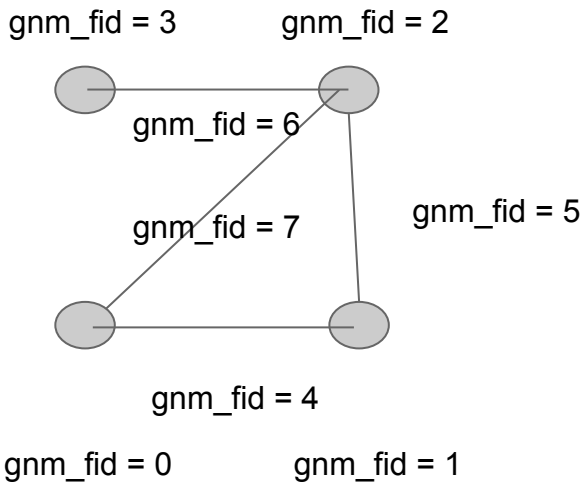
GNM

$ gnmmanage autoconnect 0.01 mon_reseau

No layers provided. Use all layers of network:

1. points
2. lines

Features connected successfully



$ ogrinfo mon_reseau _gnm_graph

OGRFeature(_gnm_graph):0
- source (Integer64) = 0
- target (Integer64) = 1
- connector (Integer64) = 4
- cost (Real) = 1.0
- inv_cost (Real) = 1.0
- direction (Integer) = 0
- blocked (Integer) = 0

OGRFeature(_gnm_graph):1
- source (Integer64) = 1
- target (Integer64) = 2
- connector (Integer64) = 5
- cost (Real) = 1.0
- inv_cost (Real) = 1.0
- direction (Integer) = 0
- blocked (Integer) = 0

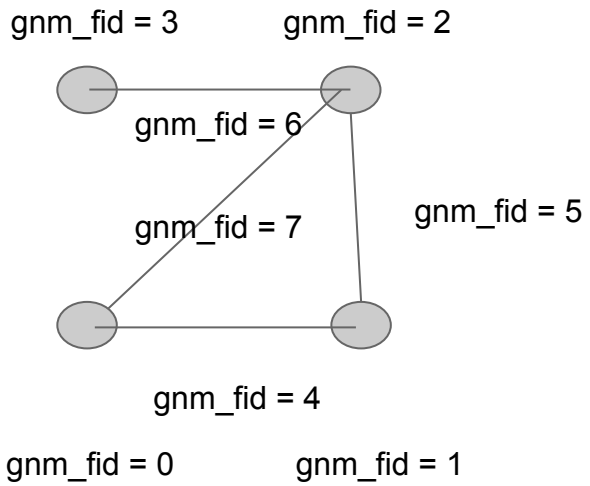
OGRFeature(_gnm_graph):2
- source (Integer64) = 2
- target (Integer64) = 3
- connector (Integer64) = 6
- cost (Real) = 1.0
- inv_cost (Real) = 1.0
- direction (Integer) = 0
- blocked (Integer) = 0

OGRFeature(_gnm_graph):3
- source (Integer64) = 0
- target (Integer64) = 2
- connector (Integer64) = 7
- cost (Real) = 1.0
- inv_cost (Real) = 1.0
- direction (Integer) = 0
- blocked (Integer) = 0

GNM

$ gnmanalyse dijkstra 0 3 -f csv \
-ds /vsistdout/ mon_reseau

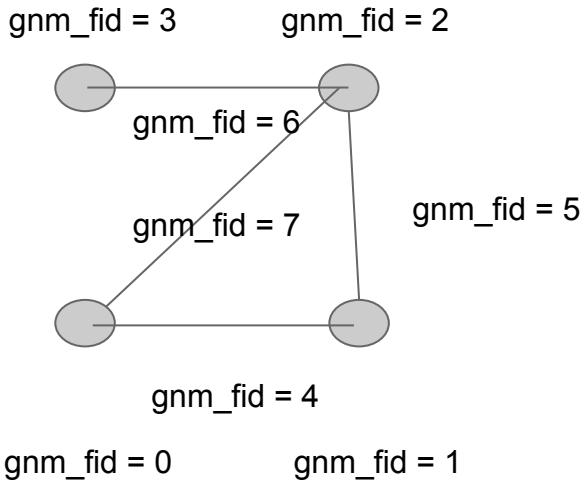
gnm_fid	ogrlayer	path_num	ftype	blocked	id
0	points	1	VERTEX	0	1
2	points	1	VERTEX	0	3
7	lines	1	EDGE	0	4
3	points	1	VERTEX	0	4
6	lines	1	EDGE	0	3



GNM

```
$ gnmanalyse kpaths 0 3 2 -f csv -ds /vsistdout/ \
        -alo "fetch_vertex=OFF" mon_reseau
```

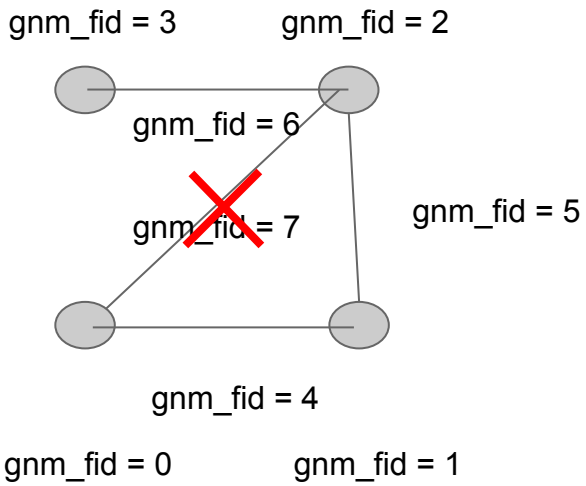
gnm_fid	ogrlayer	path_num	ftype	blocked	id
7	lines	1	EDGE	0	4
6	lines	1	EDGE	0	3
4	lines	2	EDGE	0	1
5	lines	2	VERTEX	0	2
6	lines	2	EDGE	0	3



GNM

$ gnmmanage disconnect 0 2 7 mon_reseau

Features disconnected successfully



$ ogrinfo mon_reseau _gnm_graph

OGRFeature(_gnm_graph):0
  source (Integer64) = 0
  target (Integer64) = 1
  connector (Integer64) = 4
  cost (Real) = 1.0
  inv_cost (Real) = 1.0
  direction (Integer) = 0
  blocked (Integer) = 0

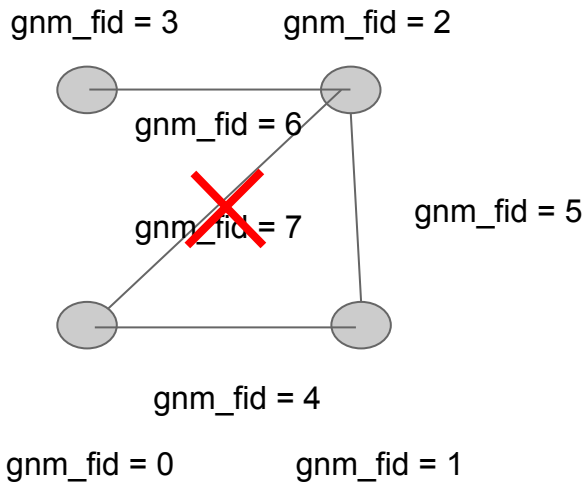
OGRFeature(_gnm_graph):1
  source (Integer64) = 1
  target (Integer64) = 2
  connector (Integer64) = 5
  cost (Real) = 1.0
  inv_cost (Real) = 1.0
  direction (Integer) = 0
  blocked (Integer) = 0

OGRFeature(_gnm_graph):2
  source (Integer64) = 2
  target (Integer64) = 3
  connector (Integer64) = 6
  cost (Real) = 1.0
  inv_cost (Real) = 1.0
  direction (Integer) = 0
  blocked (Integer) = 0

GNM

```
$ gnmanalyse 0 3 -f csv -ds /vsistdout/ \
        -alo "fetch_vertex=OFF" mon_reseau
```

gnm_fid	ogrlayer	path_num	ftype	blocked	id
4	lines	1	EDGE	0	1
5	lines	1	VERTEX	0	2
6	lines	1	EDGE	0	3



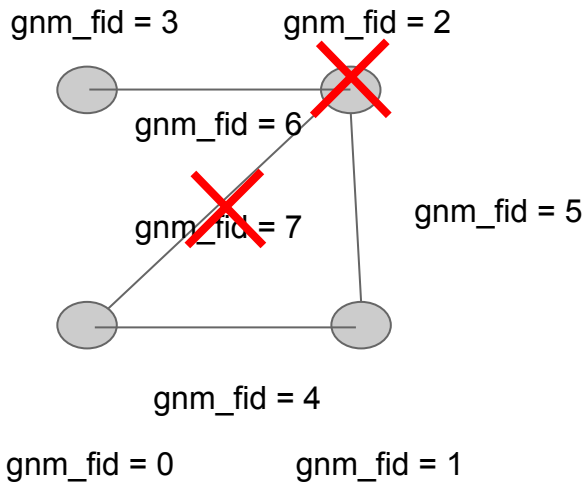
GNM

$ gnmmanage change -bl 2 mon_reseau

Change block state successfully

$ ogrinfo mon_reseau points

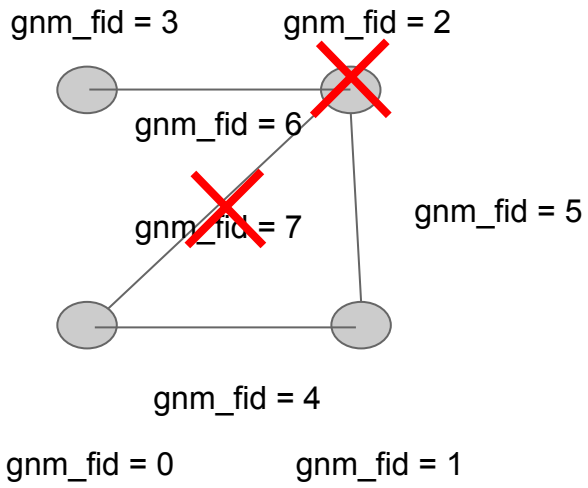
OGRFeature(points):2
  gnm_fid (Integer64) = 2
  blocked (Integer) = **7**
  id (String) = 3
  POINT (2.1 49.1)



GNM

```
$ gnmanalyse 0 3 -f csv -ds /vsistdout/ \
        -alo "fetch_vertex=OFF" mon_reseau
```

gnm_fid	ogrlayer	path_num	ftype	blocked	id



GNM

- Encore un peu expérimental
- Compilé optionnellement
- Stockage limité à shapefile ou PostgreSQL pour l'instant (limitation pouvant être aisément levée)
- Les algorithmes nécessitent l'ingestion complète du graphe en mémoire

- http://gsoc2014gnm.blogspot.fr/
- http://gdal.org/gnm_arch.html
- http://gdal.org/gnm_tut.html

RFC 59.1: Utilitaires utilisables en tant que fonctions

- Fini les :
  - os.system(“gdal_translate src.tif dst.tif”)
- Maintenant :
  - gdal.Translate(‘dst.tif’, ‘src.tif’)
  - gdal.Translate(‘dst.tif’, src_dataset)
  - gdal.Translate(‘dest_mem’, src_dataset, format = ‘MEM’)
  - gdal.Translate(dst, src, callback = my_progress_func)
- Disponible pour gdal_translate, gdalwarp, gdalinfo, ogr2ogr, gdaldem, gdalbuildvrt, nearblack, gdalgrid, gdal_rasterize
- Disponible en C/C++, Python, Java, Perl, (C#)
- Avantages :
  - Accepte des jeux de donnée en mémoire. Utile pour chainage de traitements sans devoir sérialiser/désérialiser sur disque
  - Progression / annulation
  - Plus de soucis avec le chemin des utilitaires

Autres RFCs:

- RFC 26: utilisation de table de hachage pour l'indexation des blocs en cache.
  - Rend possible l'utilisation de très grands dataset (jusqu'à 2e9 x 2e9 pixels), pour WMS/WMTS
- RFC 58: DeleteNoDataValue()
  - Suppression de la métadonné valeur nulle dans un jeu de donnée existant
  - Implémenté pour GeoTIFF, MEM, VRT, et .aux.xml
- RFC 60: Amélioration de la préservation des données lors des conversions entre format OGR identique
  - Notion de donnée native attachée à un objet
  - Pour GeoJSON
- RFC 61: Gestion des géométries mesurées (dimension M)
  - Par ex: POINT M (1 2 3), POINT ZM (1 2 3 4)
  - Shapefile, PostgreSQL/PostGIS, PGDump, MEM, SQLite, GeoPackage, FileGDB, OpenFileGDB, CSV, VRT

Autres changements

- Mise à jour vers la V8.8 de la BDD EPSG
- Algorithme d'interpolation linéaire (basé sur libqhull) dans gdal_grid (très rapide et "beaux" résultats)
- Systèmes de fichiers virtuels /vsis3/, /vsis3_streaming/, /vsicrypt/
- Script de complétion Bash pour les utilitaires en ligne de commande
- Changements internes :
  - Passe globale d'assainissement/nettoyage du code,
  - compilation sans warnings,
  - résolution d'anomalies remontés par des analyseurs statiques de code
  - corrections de nombreuses anomalies de sécurité dans des dizaines de drivers vis à vis de fichiers hostiles/corrompus

Travaux GSoC 2016 en cours

- Pilote DWG
  - Basé sur une nouvelle bibliothèque libopencad (licence X/MIT)
  - Vise compatibilité DWG R2000, R13/R14
  - https://trac.osgeo.org/gdal/wiki/DWG_driver

- Modèle géométrique complètement à niveau avec ISO SQL/MM Part 3
  - Ajout des types Triangle, TIN (Triangulated Irregular Networks), PolyhedralSurface
  - Gestion dans les pilotes Shape, FileGDB, PostGIS
  - Lien potential avec SFCGAL pour les opérations 3D

Directions futures potentielles

- Système de construction CMake
  - Construction en dehors de l'arborescence, gestion correcte des dépendances.
  - Quand complet et stable, système de construction unifié pour Unix&Windows
  - Effort en cours mené par Dmitriy Baryshnikov: https://github.com/nextgis/gdal_svn/tree/rsmd-reader-dev

- Cache de bloc raster par dataset:
  - Pour obtenir une parallélisation sans aucun verrou
  - Résoudre les problèmes d'écriture en parallèle de plusieurs dataset
  - https://trac.osgeo.org/gdal/wiki/rfc47_dataset_caching (Blake Thompson)

Directions futures potentielles

- GNM
  - Conversions entre formats de réseau (PGRouting, Spatialite, …)
  - Robustification

- Topologie planaire:
  - Nouvelle abstraction basée sur la modélisation ISO SQL/MM Part3
  - Primitives topologiques: noeuds, arêtes, faces
  - Topo-géométrie batie à partir de primitives / Topo-géométrie hiérarchique
  - Construction de topologie à partie de géométrie
  - Conversion géométrie ←→ Topo-géométrie conversions
  - Interface avec PostGIS, GRASS, Oracle, GML, Spatialite, TopoJSON
  - Conversion : topo2topo

Directions futures potentielles

- Algèbre de carte raster (raster map algebra)
- Gestion des schémas d'application GML / GML Complex Features (jeux de données INSPIRE…)
- Gestion en écriture OpenFileGDB
- Pilote GeoJSON compatible de fichiers de taille arbitraire
- Amélioration de gestion des systèmes de référence spatiaux: guessing EPSG codes, proposing appropriate datum shifts according to location, …
- Gestion du standard CRS WKT 2 / ISO 19162
- Utilisation d'une bibliothèque de traitements géométriques : Boost::Geometry
- Nouveaux pilotes, amélioration de performances, ...

Questions?

Liens:

http://www.gdal.org/

https://trac.osgeo.org/gdal/wiki/RfcList

Contact: even.rouault@spatialys.com