Engineer

GDAL

ett bibliotek för att läsa och konvertera geodata

Vad det är

GDAL, förkortning för Geospatial Data Abstraction Library och ofta skrivet GDAL/OGR, är mjukvara med öppen källkod som öppnar kartdata i ett format och skriver ut dem i ett annat, och som projicerar om dem på vägen när man ber om det. Projektets egen dokumentation kallar det ett översättarbibliotek.

Geodata kommer i två former. Rasterdata är ett rutnät av celler: en satellitbild, ett flygfoto, en höjdmodell. Vektordata är punkter, linjer och polygoner med attribut kopplade till sig: adresser, vägar, fastighetsgränser. GDAL ger programmet som anropar det en abstrakt modell för varje form, och bakom modellen sitter en drivrutin per format. GeoTIFF, JPEG 2000 och NetCDF är drivrutiner på rastersidan. Shapefile, GeoJSON, GeoPackage, KML och tabeller i PostgreSQL är drivrutiner på vektorsidan, och det finns många fler av båda slagen. Ett program som skrivs mot GDAL läser alla via samma anrop.

Vektorhalvan hette länge OGR, och de två halvorna hade separata strukturer för drivrutiner tills GDAL 2.0 slog ihop dem. Den historien förklarar varför vektorprogrammen fortfarande bär det gamla prefixet: `ogrinfo` beskriver ett vektordataset och `ogr2ogr` konverterar ett. Arbetet med biblioteket började i slutet av 1998 med Frank Warmerdam. Det släpps under en licens i MIT-stil och är ett projekt inom Open Source Geospatial Foundation, OSGeo.

Det finns tre sätt att använda det. Det första är kommandoraden, med program som `gdalinfo`, `gdal_translate` och `gdalwarp` för raster och `ogrinfo` och `ogr2ogr` för vektorer. Sedan version 3.11 finns också ett samlat kommando, `gdal`, med underkommandon som `gdal raster` och `gdal vector`, som projektet fortfarande märker som preliminärt. Det andra är kod, via bindningar för Python och andra språk. Det tredje är inuti annan mjukvara: QGIS läser t.ex. sina vektorformat genom GDAL:s drivrutiner och erbjuder GDAL-programmen i

sin verktygslåda för Processing. En typisk konvertering ryms på en rad. Här blir en shapefil ett GeoPackage som samtidigt projiceras om till den UTM-zon som används i hela Danmark:

```
ogr2ogr -f GPKG -t_srs EPSG:25832 parcels.gpkg parcels.shp
```

I skrivande stund är den aktuella versionen 3.13.3. Få verktyg konkurrerar på den här nivån. ArcGIS har egna konverteringsverktyg och enskilda format har små egna läsare, men ett team som måste ta emot vad en kund än skickar brukar till slut ha GDAL i sin stack, ofta via ett bibliotek som är beroende av det.

Använd begreppet i ett beslut

Använd ”Vad är GDAL?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Allt som händer på en plats — fartyg, fordon, tomter, sensorer — på en karta du kan söka i, mäta på och dela.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Gjorde pipelinen för geodata 50 gånger snabbare — 50x snabbare datapipelines för geodata

Gjorde pipelinen för geodata 50 gånger snabbare Hör av er →

Varför det spelar roll

Varje geospatialt system har en gräns där data utifrån kommer in: en kommuns shapefiler, en lantmätares GeoTIFF-filer, en partners export i GeoJSON, en KML-fil som någon har ritat i en webbläsare. Beslutet som GDAL påverkar är vem som står vid den gränsen. Det kan vara en person som klickar sig igenom ett skrivbordsverktyg, eller ett skript som gör exakt samma sak varje gång. GDAL är det som gör skriptet billigt att skriva.

En konvertering som är skriven som ett `ogr2ogr` - eller `gdalwarp` -kommando kan läsas vid kodgranskning, ligga under versionshantering och köras igen när nästa leverans kommer. En konvertering som görs för hand görs en gång, av den som råkar minnas stegen. För data som kommer i omgångar, månad efter månad, avgör den skillnaden om kvaliteten håller.

Licensen spelar också in. Det finns ingen avgift per användare och inget godkännande från en leverantör, och samma verktyg körs på en laptop, på en byggserver och i en container.

Det tredje skälet är räckvidden in i lagring. GDAL:s virtuella filsystem låter en drivrutin läsa data där de ligger. Med prefixet `/vsicurl/` läser den en fil över HTTP genom att be om byteintervall, så att de flesta drivrutiner kan konsumera ett fjärrdataset bit för bit utan att ladda ner allt, och `/vsis3/` gör samma sak mot Amazon S3. Kombinera det med en Cloud Optimized GeoTIFF, som är indelad i rutor, bär sina egna översikter och är upplagd för att hålla antalet anrop nere, så kan ett raster på många gigabyte ligga i objektlagring och ändå läsas ett fönster i taget.

Det finns också verkliga kostnader. GDAL är ett stort bibliotek i maskinkod med många valfria beroenden, och vilka drivrutiner en viss build innehåller beror på hur den kompilerades. Två maskiner som båda ”har GDAL” kan alltså öppna olika format. Och verktyget översätter troget, vilket innebär att det troget för över varje begränsning i det format det ombeds skriva.

Så ser det ut i praktiken

GDAL förekommer i två projekt på den här webbplatsen, i två olika roller.

Mellan 2015 och 2022, som teamledare för utveckling av GIS-mjukvara, gick leveransen av mer än trettio GIS-projekt, från utveckling via driftsättning till support genom den rollen. I botten låg PostgreSQL med PostGIS för geodatan, och GDAL/OGR var verktyget för att flytta den mellan format: shapefiler, GeoJSON, GeoTIFF, KML och vektortiles. För dataarbetet i övrigt fanns QGIS och JOSM, och kartorna ovanpå byggdes med Mapbox GL och Leaflet. Arbetet bakom den listan av format spände från digital kartläggning i 2D och 3D via LiDAR-bearbetning, georeferering och vektorisering till inomhuskartläggning och navigering, och varje projekt fick support i produktion efter leveransen.

Mellan 2022 och 2024, i en roll som dataingenjör i ett storskaligt geospatialt analysprojekt, var GDAL ett av verktygen för att hitta och rätta fel. Spatiala fel diagnostiserades och rättades med GDAL, QGIS och ArcGIS, och eftersom samma rensningsuppgifter kom tillbaka gång på gång automatiserades de i Bash-skript. PostGIS bar de spatiala frågorna och indexen, procedurer i PL/pgSQL och Transact-SQL transformerade data inne i PostgreSQL och SQL Server, och resultatet efter rensningen matade interaktiva visualiseringar byggda med Mapbox. Över 1 000 problem åtgärdades, och arbetet bidrog till 35 % kortare körtider för spatiala frågor.

Var det går fel

GDAL gör exakt det man säger åt det. De flesta problem börjar när man säger något som skiljer sig en aning från det man menade.

Punkterna hamnar på fel kontinent. Sedan GDAL 3.0 följer biblioteket den axelordning som myndigheten bakom ett koordinatsystem har bestämt, och för EPSG:4326, det välbekanta systemet med latitud och longitud, kommer latituden först. Kod som skrevs för GDAL 2 skickar longituden först, och det gör GeoJSON också. Blandar man de två byter varje par plats: en punkt i Köpenhamn dyker upp i Arabiska havet. Bestäm en ordning vid programmets kant och skriv ut den. I kod som vill ha longituden först är den dokumenterade växeln `SetAxisMappingStrategy(OAMS_TRADITIONAL_GIS_ORDER)` på referenssystemet.

Kolumnnamn kortas av och datum tappar klockslaget. En shapefil tillåter attributnamn på högst 10 tecken. GDAL kortar längre namn när det skriver, och när två avkortade namn krockar kapar det dem till 8 tecken och lägger till ett löpnummer, så att `abcdefghijkl` blir `abcdefgh_1`. Formatet lagrar datum utan klockslag. GDAL rekommenderar att `.shp`- och `.dbf`-filer hålls på högst 2 GB för kompatibilitet med annan programvara. Å, ä, ö eller kyrillisk text kan bli till brus om kodningen feltolkas när `.cpg`-filen som anger teckentabellen saknas. Skriv GeoPackage där mottagaren tillåter det. Där en shapefil krävs, byt namn på kolumnerna med avsikt före konverteringen.

Ett raster på en fjärrserver laddas ner i sin helhet. Får GDAL en `https://`-adress utan prefix väljer det sin pseudodrivrutin för HTTP, som drar in hela filen i minnet innan någon drivrutin läser den. Det passar en liten fil och är fel val för ett rikstäckande ortofoto. Sätt `/vsicurl/` framför adressen så att drivrutinen bara hämtar de byteintervall den behöver, och lagra stora raster som Cloud Optimized GeoTIFF så att intervallen blir få. Låt sedan filen vara: att redigera en COG på plats förstör dess upplägg, och sedan GDAL 3.8 vägras ett försök att öppna en COG för uppdatering om inte det uttryckligen tillåts.

Talen i en CSV kommer fram som text. CSV-drivrutinen läser varje fält som en sträng om den inte får andra instruktioner, så en kolumn med befolkningstal sorteras alfabetiskt och en summering längre fram blir ett typfel. Sätt öppningsalternativet `AUTODETECT_TYPE=YES`, eller lägg en `.csvt`-fil bredvid CSV-filen som anger varje kolumns typ.

En uppgradering slår sönder importen i Python. Paketet `gdal` på PyPI byggs mot den GDAL som redan är installerad på maskinen, så de två versionerna måste stämma överens. Projektets installationsinstruktioner låser den ena vid den andra:

```
pip install gdal[numpy]=="$(gdal-config --version).*"
```

Läs mer

- GDAL documentation — projektets egen beskrivning av översättarbiblioteket, dess drivrutiner och dess licens.
- Programs — alla kommandoradsverktyg, från `gdalinfo` och `ogr2ogr` till det nyare kommandot `gdal`.
- GDAL Virtual File Systems — `/vsicurl/`, `/vsis3/` och de andra prefixen för att läsa data där de ligger.
- OGR Coordinate Reference Systems and Coordinate Transformation tutorial — med avsnittet om axelordning och vad som ändrades i GDAL 3.0.
- COG – Cloud Optimized GeoTIFF generator — filens upplägg, dess översikter och varför redigering på plats gör dem verkningslösa.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är GDAL?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/gdal/