

# GIS

**Projekty GIS 2012/13**

**Mapování zadaného tématu**

**RG7**

**(Rasters for GRASS 7)**

**Martin Hrubý**

# Mapování

- Sběr dat pro vytvoření mapové vrstvy
  - přímé mapování v terénu
  - průzkum v dostupných datech na Internetu
- Cíl:
  - sestavení vektorové vrstvy (bodové, polygonové)
  - vyplnění atributové tabulky
  - dokumentace dat (původ, metadata)

# Požadavky na obsah a geometrii

- Téma: Instalované/provozované vodní elektrárny nad 10kW inst. Výkonu
  - Průtočné/akumulační.
- Přesnost udání polohy – 1km
- Bodová/polygonová vrstva

# Atributová tabulka

- Ofic. název zdroje
- Provozovatel
- Instalovaný výkon
- Datum kolaudace/zprovoznění
- Datum platnosti dané informace
- Množství dodané elektřiny do sítě [MWh] za rok
- Poznámka – nějaká technická zajímavost (volitelné)

# Metadata a dokumentace

- Zdroje informací (autorská práva)
- Komentář k přesnosti uvedených informací
- Forma předání dat:
  - Shapefile, zpracování v ArcGIS/GRASS
- Dokumentace:
  - Metadata
  - Sazba mapového díla
  - Zasazení do mapového kontextu – dostupná data z dat ArcGIS/veřejné zdroje dat

# Rozdělení práce

- Kraje ČR
- <http://perchta.fit.vutbr.cz:8000/vyuka-gis/49>
-

# RG7: Úvod

- Implementace rastrů a vektorů prehistorická
- Běh GRASSu
- Spuštění GRASSu
  - neexistuje „jádro“ perzistentní v paměti
  - nastavení proměnných shellu
  - GUI – pokusy o moderní GUI
- Spouštění rastrových (všech) analýz
- Redesign GRASSu
  - extrémně konzervativní (rigidní) komunita uživatelů
  - od verze 8 může být dramatictější změna

# Věčné GRASSařské koncepty

- Organizace dat v adresářích
  - Redukce počtu souborů
- Monitor + region
- Analytické moduly + skriptování
- Užitečné knihovny:
  - GDAL
  - OGR
  - PROJ.4



# Budoucí (?) koncepty

- Přejít na uložení dat v SQL-databázích (SQLite3, PostgreSQL, PostGIS)
  - přímý přístup do originálních souborů (bez importu)
- Serverové uložení dat
  - klientské stanice na tabletech
  - dnes prakticky pouze desktopové stanice
- Výkonnost zpracování dat:
  - více využívat paměť počítače – vliv na formáty uložení dat (vektory)
  - multi-threading – jádro, aplikace
  - asynchronní I/O operace - zápis

# Motivace pro RG7

- Experimentální sekundární rastrový sub-systém pro GRASS
- Experimentální systém pro výuku GIS
- **Jádro GISu musí být jednoduché a výkonné**
  - kontinuální práce na zvyšování výkonu jádra
  - zdrojové texty musí být srozumitelné
- Zavedené SW GIS knihovny – gigantické rozměry kódu, nepřehlednost

# Rastrová vrstva

- Z hlediska informatiky – Abstraktní datová struktura
- Informační obsah:
  - matice čísel (typově homogenní, řádká – NULL)
  - metadata – geografické umístění (N,S,W,E,cols,rows), další (reklasifikace, NULLvalue, paleta, ...)
  - mapové projekce minimální – vzdálenost
- Cíl: implementovat efektivní R/W přístup
- Problém:
  - potenciální velikost matice
  - různé formáty

# Způsob přístupu k ADS

- Monitor (g.region) – definuje sub-region a rozlišení
- Analytické nástroje:
  - sekvenční přístup – po řádcích / blocích
  - index-sekvenční
- Zobrazovací monitor
- Množství dat:
  - práce na resamplované podmnožině (ohraničení, rozlišení)
  - práce na originálu dat

# API GRASSu

- Otevři/vytvoř/zavři soubor
- Přečti resamplovaný řádek
- Zapiš (resamplovaný) řádek
- Jádro (knihovna) identifikuje zdrojová data, načte je, provede resampling
  - výsledek uloží do uživatelova bufferu

# Co lze zlepšit?

- API – zavedení pohyblivého „okna“ s automatizovaným zápisem a čtením
  - abstrakce nad současným API
  - abstrahované vytváření cache nad daty
  - zvýšení komfortu pro tvůrce analýz
  - přepsání analytických nástrojů???
  - zavedení standardu „okenního“ API, tj. doporučení pro budoucnost?
- Způsob uložení/organizace dat
- Řízení přístupu k datům

# Způsob uložení dat

- GRASS formát – jeden velký soubor + NULL data
  - import/export – ASCII zdroj, GDAL zdroj, geokoordinace
- RG7:
  - relativně malé 2D bloky – dlaždice (2D, 1D)
  - virtualita formátu
  - Operace nad dlaždicí – save, load, test
- Nad tímto konceptem lze definovat vyšší vrstvy
  - uživatelské API, serverové uložení
  - 3D rastr, pyramidy

# Řízení přístupu k dlaždicím

- Virtuální dlaždicový prostor
- Na uživatelskou žádost (buňka/blok rastru):
  - překlad jeho požadavku do indentifikace dlaždic
  - zpřístupnění dlaždic v paměti
  - resampling, předání výsledků
- Důvody pro ponechání dlaždic v cache:
  - GRASS spoléhá na buffery OS
  - styl analytických algoritmů, opakovaný přístup k vrstvě
  - serverové poskytování dat
  - asynchronní čtení/zápis



# Terminologie

- struct Region
  - LB, TR – left bottom, top right – Coord
  - size – lcoord
- Xcoord – x, y
- Region rastru – jeho faktický rozměr.
- Minimum region (Tiling region).
- definice: basics.h

# Způsob rozdělení do dlaždic

- 4Tree – myšlenka byla opuštěna.
- pravidelné rozdělení + okraje
- logická souřadnice dlaždice (tile\_xy, pojmenování)
- logická souřadnice buňky
- Region, minimum\_region
- Předpokládá se
  - obecně různá velikost dlaždic
  - **různý způsob rozkladu** do dlaždic pro každý soubor

# Moduly RG7 (vrstvy abstrakce)

- Fyzické uložení dat, třídy:
  - Ras\_phtile – virtualizovatelný obsah konkrétní jedné dlaždice
  - Ras\_phys\_file – řídí přístup k dlaždicím
  - Ras\_file\_interface (virt.) - rozhraní k uložení obsahu dlaždic
  - Ras\_metadata – standardizované uložení metadat
  - Ras\_tiling – překlad Coord na log. adresu dlaždice, určení regionu každé dlaždice

# Vazby mezi objekty

- **Ras\_phys\_file**
  - seznam Ras\_phtile (známé, alokované)
  - iface – ref. na Ras\_file\_interface
  - tiling – ref., prostor pro experimentování
  - atribut: max\_available\_tiles – velikost cache
    - zřejmě musí být cache pružná ve velikosti
- **Ras\_file\_interface**
  - Ras\_metadata

# Moduly interfaceXXX

- Ras\_sqlite3\_interface
- Ras\_grass\_interface
- Ras\_wms\_interface
- Další databázové a souborové interface
  - včetně systému dlaždice=soubor
- Síťový interface – zatím ???
  - Ras\_phys\_file lokální
  - Ras\_xxxnet\_interface čte vzdálená data

# Rozhraní Ras\_file\_interface

- open/close/create/drop
- write\_metadata/load\_metadata
- transaction(BOOL start/end)
- int swap\_tile(Ras\_phtile \*)
- int load\_swapped\_tile(Ras\_phtile \*)
- int swapped\_tile\_available(Ras\_phtile \*)
- Returns:
  - tech failed
  - ok
  - ok, no data

# Rozhraní Ras\_phys\_file

- Konstrukce objektu – vstup – interface objekt
  - vytvoří tiling, pooly, buffery
- wgeo/rgeo (Coord)
- wcell/rcell (Icoord)
- Region &region()
- Ras\_metadata &metadata()
- Zbytek private...
- Problém k řešení:
  - efektivní R/W pro sub-regiony

# Přístup na vrstvu Ras\_phys\_file

- Expertní použití
- Výkonostní testování
- Očekává se spíše přístup přes „okna“
  - Ras\_window – opět rastrová ADS
- Strategie průchodu rastrem



# Vrstva oken

- Ras\_win\_abstraction – resampling do rozlišení monitoru, ořezání rastru (okraje)
  - Maskování
  - Reklasifikace – úroveň (physfile, window) ???
- Ras\_window
  - nastavená velikost okna (rozlišení monitoru)
  - operace – umístění okna/posun okna
  - abstraktní přístup – zápis/čtení
- Ras\_multilayer\_window
  - vrstva == jeden rastrový soubor

# Vrstva RG7\_system

- Uživatelské rozhraní na mapové vrstvy v rámci koncepce lokací a mapsetů
- Obsahuje informace o otevřených rastrech
- Vytváří na požádání Ras\_window \*
- Zatím „desktop“ pojetí
  - serverové pojetí: RG7\_system na serveru + API + lokální podoba RG7\_system
- Výhledově: sjednocení cache pro všechny soubory (??)

# Otevření existujícího rastru

- Vznikne objekt `Ras_file_interface`
  - načte metadata (`Region`, `minimum_region`)
- Vznikne objekt `Ras_phys_file`
  - obdrží iface
  - vytvoří tiling
- Dotaz ve formě `r/wgeo` nebo `r/wcell`
  - identifikace dlaždice
  - zajištění obsahu dat pro dlaždici
  - předání požadavku dlaždici

# Obsluha rgeo požadavku

- překlad Coord → Icoord
- překlad Icoord → tile\_xy (objekt tiling)
- locate\_tile\_xy(ICoord tile\_xy) → Ras\_phtile
  - již zavedena, pak known\_tiles.get(tile\_xy)
  - ne:
    - vytvoření Ras\_phtile (tiling.region\_...(tile\_xy))
    - výpočet ilb pro dlaždici
    - zavedení mezi known\_tiles
- rw\_access\_tile(Ras\_phtile \*)

# rw\_access\_tile(Ras\_phtile \*tile):

- tile->allocated() → true
- make\_tile\_allocated(tile)
  - if (needs\_reallocate), reallocate (samostatný problém)
  - přiděl 2D buffer (pool\_of\_buffers)
  - zapiš do allocated\_tiles
- iface.swapped\_tile\_available(tile)
  - je, iface.load\_swapped\_tile(tile)
  - není, vyplň obsah tile NULLvalue

# make\_tile\_deallocated(Ras\_phtile

- `iface.swap_tile(tile)`
  - podmíněno `tile->needs_flush()`
- vrácení bufferu do `pool_of_buffers`
- `allocated_tiles.erase(tile)`
- Na zvažení: `deregister_tile(tile)`:
  - `known_tiles.erase(tile)`
  - delete tile

# Realokace v `allocated_tiles`

- Lze se inspirovat algoritmy v OS pro stránkování paměti?
- Myšlenka:
  - cache je omezená (`max_available_tiles`)
  - je potřeba alokovat dlaždici
  - nějaká (jedna, víc) alokovaných dlaždic musí dealokovat
- Která?
  - statistiky přístupů, vyhodit nejméně přistupovanou
  - rada od uživatele (které chce používat?)
  - chybně navržený mechanismus sníží výkonnost RG7

# Realokace: vliv uživatele

- Uživatel v rastru umístí přístupová okna
- `max_available_tiles` se musí přizpůsobit
- Areas of Interest (AOI):
  - specifikace regionů v rastru, které budou (např. v budoucnu) přístupovány
  - tyto regiony obsáhnou dlaždice, priorita dlaždice
  - v okamžiku `needs_reallocate()` v uživatelském vláknu:
    - statistika priorit dlaždic
    - potřebný počet dlaždic s nejnižší prioritou - `make_tile_deallocated(...)`



# Použití AOI

- Umístění `Ras_window` – definuje jednu AOI
- Definice AOI: `Ras_area_of_interest`
  - pouze zvýší prioritu množiny dlaždic, není zatím `allocated()`, načtena, resampling
- Je jasné, že suma prioritních dlaždic může být větší než `max_available_tiles` (není dogma)
  - pak je nutné přeformulovat:
    - `needs_reallocate`
    - zavést dva ukazatele: ideální cache, reálná cache

# Ideál AOI – background thread

- RG7\_system spustí vlákno:
  - na určitý podnět se aktivuje
  - projde všechny Ras\_phys\_file
  - načte nealokované dlaždice s priority>0
  - dealokuje dlaždice s priority==0, pokud je třeba
- Thread-safe Ras\_phys\_file (analýza kódu)
- Možná zámky na dlaždicích (zdůvodnit)
- Smysl:
  - odlehčení uživatelskému vláknu
  - asynchronní (odložený) zápis, dopředné čtení

# Způsob formulace AOI

- Uživatel požádá o načtení řádku 1
- Označí řádek 2 za budoucí „zájem“
- Zpracovává řádek 1 (vlákno AOI načítá ř. 2)
- Uživatel chce řádek 2. Ten už je v paměti.
- Zřejmý problém:
  - načasování, synchronizace
  - může toto uživatel formulovat efektivně? Abstraktní pojetí API.
  - RG7 toto smí někdy ignorovat (režie může převýšit nad užitkem)

# Rozbor okruhů pro řešení

- 1) ADS Tile\_storage, Pool\_of\_phtile\_buffers
- Tile\_storage:
  - aktuálně `std::map<(Int,Int), Ras_phtile *>` -  $O(\log)$
  - konst. přístup (vkládání, vyhledávání, mazání)
- Pool\_of\_phtile\_buffers (RG7\_system)
  - může nekonečně růst
  - každá dlaždice může mít různou velikost
  - přidělovací strategie
  - ADS pro roztřídění seznamů volných bufferů dle velikosti (délka, je to 1D)

## 2) Iface pro GRASS

Minimálně operace pro čtení

# 3) Iface pro WMS

- Vyžaduje networking
- Není aktuálně kritické
- Smysl dává:
  - open/close
  - load\_metadata
  - load/test tile

## 4) cmd\_inputs

- Konceptní řešení zadávání parametrů z příkazové řádky
- Inspirace GRASS

# 5) compressed\_ba

- Implementace metod `compress()/decompress()`
- Kompresi bude volitelná (nastavení v metadatech)
- Uvažuje se operace nad jednou dlaždicí
- Různé průchody dlaždicí
- Kam umístit volání `compress()/decompress()` ?
  - operace v `Ras_phtile`
  - operace v `Ras_file_interface`



## 6-7) Ras\_metadata

- Zřejmě nutnost spojit s load/write\_metadata
- Zavedení reklasifikační tabulky + reference
- Kategorie
- Barevná paleta

# 8) blokové operace Ras\_phys\_file

- Návaznost na Ras\_window
  - load\_window\_contents()
  - flush()
  - nahradit 2D cyklus blokovým voláním Ras\_phys\_file
- Identifikace dlaždic + zpřístupnění dlaždic
- Resampling do zadaného bufferu
  - vstup: geo. krok - Coord

# 9) Experiment: řádkové dlaždice

- Celé řádky nebo řádkové úseky
- Vytvoření alternativního Ras\_tiling
- Nalezení optimální velikosti 1D dlaždice
  - vyšší počet dlaždic → vyšší režie Ras\_phys\_file
- Všechny algoritmy RG7 iterují přes řádky
- Řádkový přístup převládá v GRASS analýzách

# 10) Thread-safety (analýza)

- Návaznost na koncept AOI:
  - vlákno uživatele – okna, přístup na RG7\_system
  - vlákno RG7\_system

# 11) Optimalizace SQLite3

- Hlubší prostudování/znalost SQLite3
- Nastavení velikosti bufferů
- Asynchronní operace + možné problémy

# 12) PostgreSQL iface

- Obdoba Ras\_sqlite3\_interface
- Výkonnostní srovnání
- Způsob uložení dat v PostgreSQL nevyhovuje cílům RG7. Aplikace nejspíš pro serverové provedení.
- Provéřit, zda-li to má přínos

# 13) Barevné palety

- Systém předdefinovaných palet
  - typizované: DEM, rainbow, šedé
- Uživatelské palety
- Palety:
  - diskrétní
  - intervaly
- Činnost:
  - vrát' RGB pro vstupní hodnotu

# 14) Ras\_multilayer\_window

- Ras\_window pro více vrstev (společný region)
- Alternativně: optimalizace Ras\_window
  - posuny okna



# Pro velmi odvážné

- Návrh vektorového sub-systému ve stylu RG7
  - Uložení v SQLite3.
  - Prostorový index.
  - Zatím bez nároků na topologii.
  - Fakticky vzato SHP formát.

# Závěr

- Zopakování motivace pro RG7:
  - alternativa rastrů v GRASSu – prověření smyslu.
  - základ pro vlastní FIT-GIS (informatické pojetí)
- Motivacím odpovídají i okruhy projektů
  - některé vědomě duplikují části GRASSu
- Koncepce RG7 a budoucí výhledy zahrnují velkou částí IT
  - cvičná platforma pro studenty informatiky
  - hledání ideálu koncepce a výkonnosti
  - je třeba překonat „počáteční fázi“ projektu