Indexování u prostorových DB

Z FITwiki
(Přesměrováno z Metody indexování bodových a plošných útvarů)

Prostorové databáze

[1]

Obsah

(http://mistral.in.tum.de/rwork/gg98.pdf)

[2] (http://dna.fernuni-hagen.de/Tutorial-neu.pdf) Pěkný popis základů (resp. v souvislostech). Dušan Kolář ve slajdech čerpal zaručeně odtud...

Tvorba indexu pro prostorové datové typy umožňuje zvýšení efektivity např. u **selekce** a **Join**. V základe existujú dve možnosti pre indexovanie priestorových objektov:

- transformácia nD objektov do 1D bodov a použitie štandartných indexovacích metód (B-tree)
- použitie špeciálnych štruktúr

Obvykle stačí indexovať len body alebo obdĺžniky/hyper-obdĺžniky, ktoré obalujú priestorové objekty, ale existujú aj metódy pre indexovanie polygónov.

Indexovanie 1D

Jedná sa o klasické indexovanie dát, nad ktorými je definované usporiadanie. Pre potreby priestorových dát chýba usporiadanie na viacerých rozmeroch. Pri transformácii do 1D dochádza k strate susednosti. Nemusí být realizovatelná či transformovatelná zpět.

Lineárne/Adaptívne hashovanie

B-strom

(viz Wikipedia B-strom (http://cs.wikipedia.org/wiki/B-strom))

- index-sekvenčná štruktúra
- vyvážený (sám se vyvažuje)
- adaptabilní - môže ľahko vkladať a rušiť rôzne podstromy

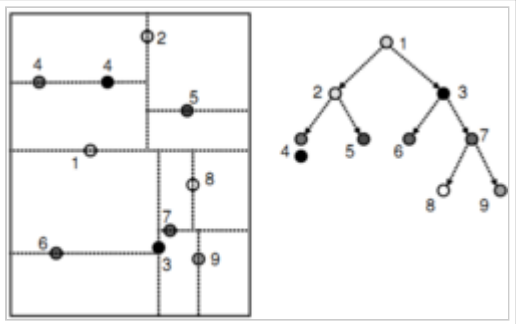
Pozn. Štátnicová otázka: Aká je medz, pri ktorej sa uzol B-stromu pokladá za dostatočne prázdny a zlúči sa? Odpoveď: 50%.

Indexovanie bodov v nD

Stromy

K-D Tree

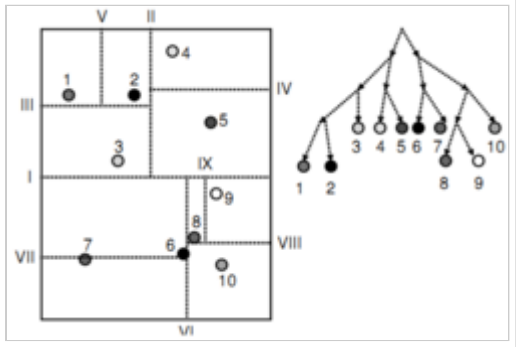
- Binárny strom - dělí prostor hyperplochami (přímka, plocha, ...), které jsou **rovnoběžné s osami** a **musejí obsahovat alespoň 1 bod** . Žádný bod nesmí být současně ve více plochách
- Vhodný pro statická data: rychle vyhledává a přidává, pomalu odstraňuje (pri odstraňovaní bodu v nelistovom uzle treba jeho podstrom zahodiť a znova zaindexovať), špatným přidáváním degraduje.



Ukázka K-D Tree

Adaptivní K-D Tree

- Dělí prostor hyperplochami rovnoběžnými s osami tak, aby **v obou polovinách byl zhruba stejný počet bodů**, přičemž body neprochází.
- **Body jsou v listech** (1 bod na list oproti K-D Tree) - jednoduché rušení
- Nejvhodnější opět pro statická data
- Odstraňuje nevýhodu závislosti na pořadí vkládání - (u K-D tree se vždy dělí právě vkládaným bodem, zde nemusí)



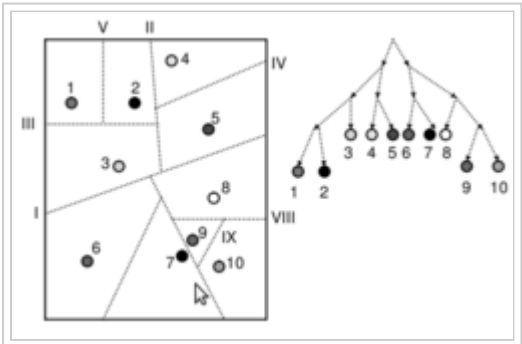
Ukázka adaptivního K-D-tree

Bin tree (Binary Interval)

- **rekurzivně dělí podprostor na hyperkrychle stejné velikosti až každá obsahuje max. jeden bod**

BSP tree (Binary Space Partitioning)

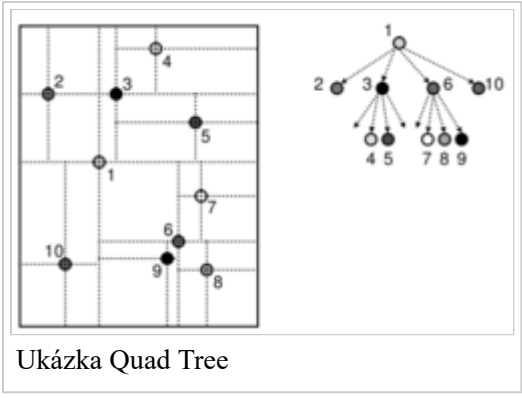
- Pracuje stejně jako Adaptivní K-D Tree, ale **hyperplochy nejsou rovnoběžné s osami**
- Dělení se provádí tak dlouho, dokud počet bodů v podprostorech neklesne pod limit (nemusí být 1 jako pro adaptivní K-D).
- Nepřináší nějaké zlepšení, spíše vyšší nároky na paměť



Ukázka BSP Tree

Quad tree

- Odpovídá K-D Tree, ale nedělí na dvě poloroviny, kdežto na 2^n podstromů, kde n je dimenze prostoru. *Quad-tree dělí na 4 oblasti. Pro 8 oblastí je zde octree a pro obecně n oblasti n dimensional tree.*
- Některé větve nemusejí obsahovat data.
- Existuje varianta dělící v bodech (**point QT**) a na stejné části (**region QT**)
- *Například nelistový uzel bodového kvadrantového stromu pro čtyřdimenzionální prostor bude mít **až** 16 následníků*



Ukázka Quad Tree

Hashovacie algoritmy

1) Vícerozměrné lineární hashování

Robí sa pomocou kriviek vyplňujúcich priestor:

- Z-krivka (http://en.wikipedia.org/wiki/Z-order_(curve))
- N krivka (Lebesgue Curve 2D (http://www.robertdickau.com/lebesgue2d.html) 3D (http://www.robertdickau.com/lebesgue3d.html))
- Hilbertova krivka (http://en.wikipedia.org/wiki/Hilbert_curve)
- ...

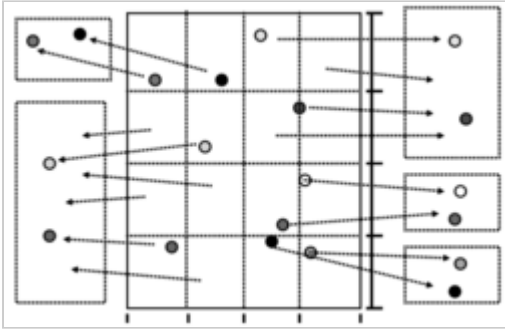
používa sa aj pre viacrozmerné objekty

2) Adaptívní hashování

Grid file

- Prostor rozdělený n-rozměrnou mrížkou (ne nutně pravidelnou)
- Adresář (uložený na disku) priřazuje každou bunku k nějaké dátové jednotky (bucket)
- Dátové jednotky obsahují informáce o konkrétních bodech, jejich obsah sa prochází už sekvenčně
- Dátová jednotka nesmí byť prázdna; při přeplnění dochádzí k rozdělení mrížky daľší hyperplochou, což může být **nelokální** změna (nežádoucí nárust adresáře)

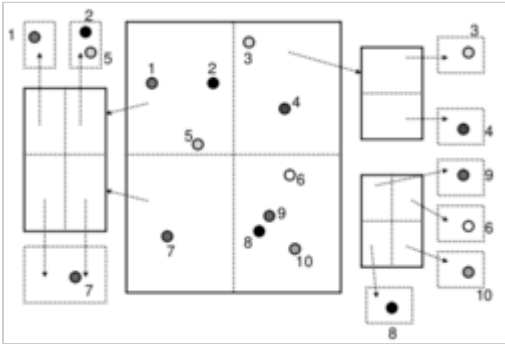
- Mazaní též není lokální. Při vyprázdnění dátové jednotky dochází k jejímu slučování s jinou a je nutné ověřit odstranění hyperplochy, případně převzorkovat prostor, do kterého zasahovala.
- Využití prostoru: 69%



Ukázka Grid file

Two Level Grid file

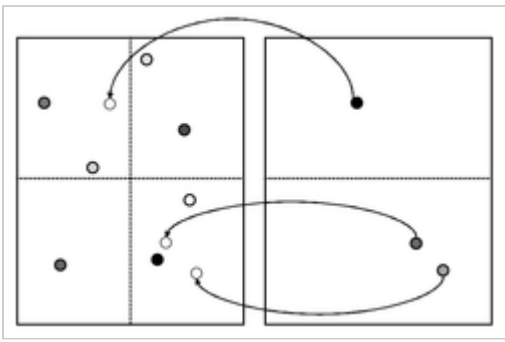
- Grid file na vrchní úrovni (kořenový adresář) adresuje do grid files na druhej úrovni (pod-adresář).
- Zmeny jsou časteji lokální, ale k nelokálním může stále docházet



Ukázka Two Level Grid file

Twin Grid file

- Dva rovnocenné grid files (primární a přetokový) bez hierarchického vztahu
- Body se umisťují do primárního, pokud je plný (muselo by se dělit) dávají se do přetokového
- Rovnoměrné rozložení dat
- Využitie priestoru: až 90% (zlepšení) a bez zpomalení



Ukázka Twin Grid file

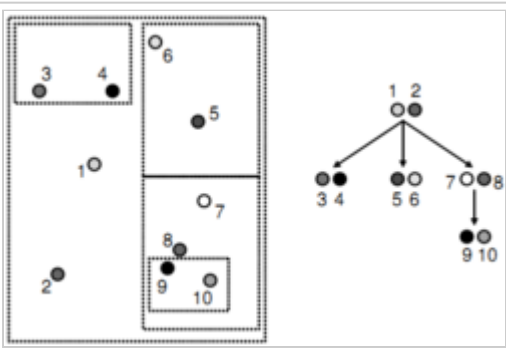
EXCELL

- Jako grid file
- dělí na jednotky stejné velikosti
- dělení je plošné
  - velký adresář
  - později hierarchie
  - přetokové stránky

Hybridné algoritmy (hashování a stromy)

BANG file (Balanced and Nested Grid file)

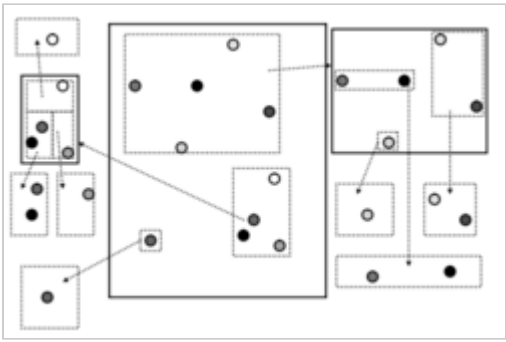
- Bunka tvorí také dátovú jednotku
- Dátové jednotky sa prekrývajú
- Odstraňuje problém exponenciálního nárůstu adresáře
- Překrývající se a vnořené buňky (hierarchické)
- Datové jednotky jsou uloženy ve vyváženém stromě



Ukázka BANG File

Buddy tree

- Základem je hashování, adresář je ale stromová struktura s minimálně dvěma položkami
- Dělení hyperplochami rovnoběžnými s osami – dále se prostor omezí na minimální obvodový obdélník
- Dělení tak dlouho, než počet klesne pod určitou mez
- Jediná plně hybridní struktura
- Ukazatele na datovou jednotku až v listech



Buddy tree

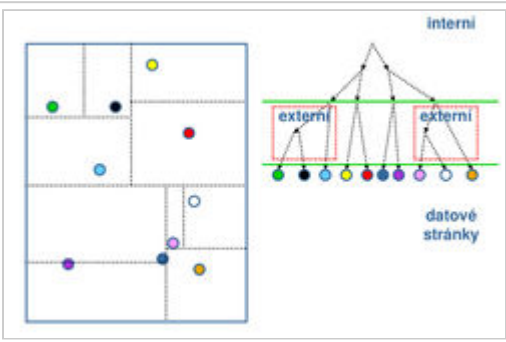
Přístup k bodům

K-D-B-Tree

- podobné Adaptivni K-D Tree, ale listy jsou datové jednotky namísto bodů (body v datových jednotkách jsou prohledávány sekvenčně)
- balancovan pomoci B-Tree
- vkladani: muze zpusobit rozdeleni (heuristiky na optimalni rozdeleni; propagace stromem)
- mazani: slucovani pri podteceni

LSD Tree

- Adaptivní K-D Tree
- Výškově vyvážený strom
- Externí adresářová stránka
- Nejen pro prostorová data



Simple example of an LSD-tree

Indexovaní vícerozměrných objektů

Indexace bodů je sice převážná část indexování, ale nestačí pro aplikační nasazení. Pro vícerozměrná data se používají stromové struktury a hashování (PLOP, Multi-Layer Grid File, R-File)

Transformace (mapování)

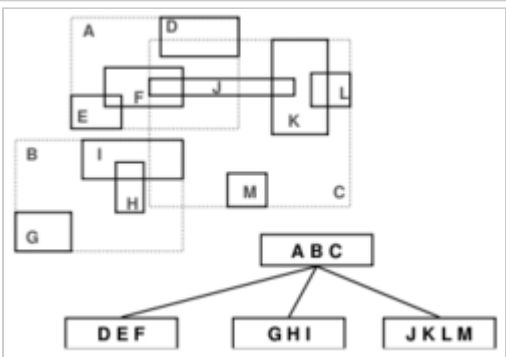
Mapuje objekty popsané *k* body v *n*D prostoru na body v *k*n*D prostoru (např. obdélník ve 2D je bod ve 4D). Některé dotazy nejsou realizovatelné, mapování složité, někdy nemožné, problém zpětné transformace výsledku.

Překrývání (vymezování)

Buňky se překrývají a datové jednotky (buckets) většinou také, vzniká více možných cest prohledání, ale neví se kolik (problém při paralelizaci).

R-Tree [3] (http://cs.wikipedia.org/wiki/R-strom)

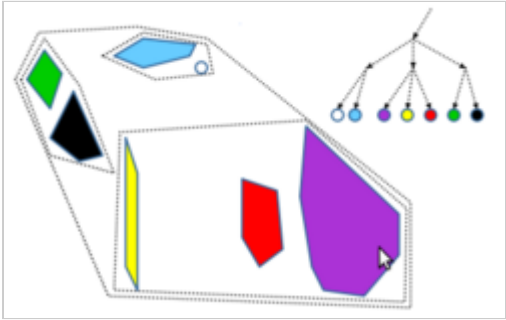
- Podobné B-stromům - data v listech
- Zástupce metody **překrývání** - objekt sa nachádza len v jednom liste, no môže existovať viac prehladávaných vetví.
- Obaluje objekty bounding boxy (pokud obsahují moc bodů, rekurzivně dělí), vždy celý objekt -> mohou se překrývat a indexované objekty mohou zasahovat do více buněk



Simple example of an R-tree for 2D rectangles

P-Tree

- Indexuje body aj polygony
- Založeny na adaptívnom K-D Tree
- Používa konvexné obálky (polygony)
- Data v listech

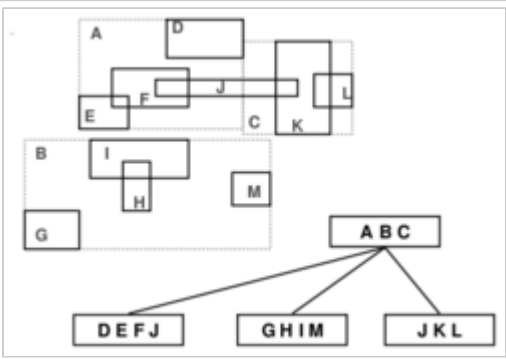

P-tree

Ořezávání (duplikace objektů)

Buňky se nesmějí překrývat, jediné řešení je pak rozsekat objekty na části podle hranic dotýkajících se bounding boxů. Zpětné shlukování je problém. Při rozšiřování prostoru datových jednotek může dojít k deadlocku, když už se nedá dělit.

R^+-Tree

- Zástupce metody **ořezávání**.
- Je podobný R-Tree, ale namísto překrytí dělí objekty na části
- Pokud objekt zasahuje do více buněk, které nesousedí, je třeba vložit buňku pro střední část nebo rozšířit jednu z buněk (může nastat problém).



Simple example of an R+-tree for 2D rectangles

Vícerozměrné lineární hashování

- Vychází z indexovaní bodů: křivky vyplňující prostor
- **Multi-Layer Grid File**, **R-File**, **Z-hashing**

Externí odkazy

- Wikipedia: Spatial index (http://en.wikipedia.org/wiki/Spatial_index)
- Spatial Index Demos (http://donar.umiacs.umd.edu/quadtree/) - kopec interaktívnych appletov
- Wikipedia: kd-tree (http://en.wikipedia.org/wiki/Kd-tree)
- Wikipedia: R-Tree (http://en.wikipedia.org/wiki/R-tree)
- PDB opora od strany 39

Kategorie: Pokročilé databázové systémy

- Stránka byla naposledy editována 15. 6. 2017 v 09:11.