

# POROVNÁNÍ VÝKONNOSTI DATABÁZÍ POHYBUJÍCÍCH SE OBJEKTŮ

**Tomáš Volf**<sup>1</sup>

<sup>1</sup>*Ústav informačních systémů, Fakulta informačních technologií VUT v Brně,  
Božetěchova 1/2, 61266 Brno, +420 541 141 316, ivolf@fit.vutbr.cz*

## **Abstrakt**

Databáze pohybujících se objektů umožňují uchovávat a dotazovat objekty, které mění svoji polohu v průběhu času nebo mění v průběhu času společně s polohou také svůj tvar v prostoru. Časoprostorová data mohou být reprezentována různými způsoby. Databáze pohybujících se objektů mohou být užitečné pro ukládání dat (například trajektorií pohybů objektů) z různých bezpečnostních a dohledových systémů a jejich pozdější dotazování. Nejen pro takovéto využití je potřeba, aby zpracovávání dat bylo co nejrychlejší. Tento článek se zabývá porovnáním výkonnosti rozšiřitelného databázového systému Secondo, které používá pro správu dat databázi Berkeley DB, a aplikačního programového rozhraní VTApi, které databázi PostgreSQL pro správu dat. Článek podává základní informace o systému Secondo a rozhraní VTApi. Dále uvádí výběr testovacích dat, jejich krátký popis a následné úpravy pro účel testování. Práce představuje vybrané metody pro testování výkonnosti a jejich výsledky. Na závěr jsou výsledky diskutovány a jsou navrženy možnosti dalšího pokračování.

***Klíčová slova:*** *pohybující se objekty; porovnání výkonnosti; Secondo; VTApi.*

## **Úvod**

Databáze pohybujících se objektů umožňují uchovávat a dotazovat objekty, které v průběhu času mění svoji polohu v prostoru, případně kromě své polohy mění také svůj tvar v prostoru. Takovéto databáze nacházejí uplatnění v různých bezpečnostních, dohledových a monitorovacích systémech, například při detekci objektů a jejich trajektorií pohybu v dohledových systémech pro ukládání a dotazování trajektorií těchto objektů.

Pohybující se objekty mohou být v databázi reprezentovány více způsoby. Můžeme zvlášť uložit polohu objektu v prostoru a zvlášť uložit čas, ve kterém se objekt nacházel na této poloze. Jiný způsob reprezentace trajektorie může chápat čas jako další dimenzi. V tomto případě je dvojdimenzionální poloha objektu reprezentována trojdimenzionálně, kde třetím rozměrem je čas, ve kterém se objekt nacházel na dané poloze.

Z pohledu pohybujících se objektů jsou důležité dvě základní abstrakce: pohybující se body a pohybující se oblasti [1]. U pohybujícího se bodu se zajímáme pouze o jeho polohu v čase, kdežto u pohybující se oblasti nás zajímá nejen poloha oblasti v čase, ale také její tvar, který se může v čase měnit.

Tento článek se zabývá porovnáním databázového systému Secondo<sup>1</sup> s aplikačním rozhraním VTApi<sup>2</sup> z pohledu výkonnosti. Oba projekty jsou vyvíjeny v prostředí univerzit, jsou také vyvíjeny s ohledem na multiplatformost (pro operační systémy Linux, Windows a další). Oba jsou poskytovány jako svobodný software s otevřeným zdrojovým kódem.

Secondo je rozšiřitelný databázový systém, který používá pro ukládání dat databázový systém Berkeley DB [2]. Secondo vyvíjí tým z Fakulty matematiky a počítačových věd na univerzitě v Hagenu. Skládá se ze tří hlavních komponent: jádro, optimalizátor a grafické uživatelské rozhraní.

---

<sup>1</sup> <http://dna.fernuni-hagen.de/Secondo.html>

<sup>2</sup> <http://vidte.fit.vutbr.cz/vtapi.html>

Jádro přímo zpracovává dotazy nad množinou modulů algebry, které zapouzdřují veškerou funkcionalitu závislou na datovém modelu [3], a umožňuje pomocí relativně jednoduché syntaxe, zvané prováděcí jazyk, zadat přímo dotazovací plány nebo ekvivalentní výrazy algebry, které následně typově zkontroluje a vyhodnotí. Dotazy se zapisují pomocí vnořených seznamů [4]. Vnořené seznamy jsou používány také pro zápis typových výrazů a pro hodnoty datových typů při jejich externí reprezentaci. Jádro je implementováno v jazyce C++.

Optimalizátor podporuje jazyk SQL, který transformuje do prováděcího jazyka. Hlavní funkcí optimalizátoru je optimalizace spojování dotazů. Optimalizátor je implementován v Prologu a je jej možné jednoduše rozšířit o nová optimalizační pravidla a jejich cenové funkce.

Grafické rozhraní zobrazuje dotazovaná data a informace o nich. Při zadání dotazu v prováděcím jazyku dotaz pošle ke zpracování jádru. Při zadání dotazu v jazyku SQL grafické rozhraní spolupracuje s optimalizátorem, kterému se tento dotaz předá. Optimalizátor vrátí prováděcí plán grafickému rozhraní, které plán nechá zpracovat jádrem. Grafické rozhraní je implementováno v Javě a dá se rozšířit pomocí prohlížečů.

VTApi je aplikační rozhraní, které si klade za cíl zjednodušit manipulaci s multimediálními daty a jejich metadaty při vývoji aplikací zaměřených na počítačové vidění. VTApi je implementováno v jazyce C++ a pro ukládání dat využívá databázový systém PostgreSQL ve verzi 9.1; pro podporu prostorových a časoprostorových datových typů a operací využívá rozšíření Postgis (ve verzi 1.5.3) a knihovnu GEOS. Rozhraní je vyvíjeno týmem z Fakulty informačních technologií Vysokého učení technického v Brně.

## Materiál a metody

Pro porovnání výkonnosti systému Secondo a VTApi byla použita databáze *berlintest*, která je součástí systému Secondo. Jedná se o vzorovou databázi pro demonstraci práce se systémem Secondo. Z této databáze byla pro účel porovnání vybrána tabulka *Trains*, která popisuje polohu jednotlivých kurzů berlínských linek metra v čase (trajektorie) po celý den, a objekt *tiergarten*, který obsahuje prostorový popis území zoologické zahrady. Následující zápis popisuje schémata tabulky *Trains* a objektu *tiergarten*.

```
Trains(Id: int, Line: int, Up: bool, Trip: mpoint)
tiergarten: region
```

Nejprve bylo potřeba exportovat data ze systému Secondo, následně vytvořit tabulky v databázi PostgreSQL a importovat do nich takto exportovaná data. Při exportu tabulky či objektu ze systému Secondo jsou data zapsána pomocí vnořených seznamů. V těchto vnořených seznamech jsou prostorové a časoprostorové hodnoty popsány pomocí úseků. Časoprostorový úsek obsahuje zvlášť časový úsek a zvlášť prostorový úsek, jak ukazuje následující zjednodušený zápis trajektorie, pro jednoduchost omezené na dva úseky.

```
((("2003-11-20-06:03" "2003-11-20-06:03:52.685")
 (13506.0 11159.0 13336.0 10785.0))
 ("2003-11-20-06:03:52.685" "2003-11-20-06:04:08.127")
 (13336.0 10785.0 13287.0 10675.0)))
```

Ze zápisu je patrné, že se vždy koncový bod předchozího úseku opakuje s prvním bodem úseku následujícího. Pomocí skriptu PHP a poměrně složitých regulárních výrazů byla tato data transformována na skripty SQL, které naplní příslušné tabulky v databázi PostgreSQL. Pro reprezentaci prostorových a časoprostorových hodnot v databázi PostgreSQL je použit datový typ *geometry*. Pohybující se bod je reprezentován jako trojrozměrný řetězec čar, kde čas je třetí

dimenzí, pomocí datového typu *linestring*. Jelikož jsou data v tabulce *trains* pouze z úseku jednoho dne, je možné jej do třetí dimenze transformovat například tak, že zde bude časový údaj vyjádřený v sekundách v rámci jednoho dne, protože údaj o datu v tomto případě není významný. Následující zápis ukazuje transformovaná data z výše uvedeného příkladu zjednodušené trajektorie.

```
LINESTRING(13506.0 11159.0 21780, 13336.0 10785.0 21832.685, 13287.0 10675.0 21848.127)
```

Secondo při zadání dotazu v terminálovém rozhraní vrací pro každý dotaz časový údaj, jak dlouho trvalo zpracování požadovaného dotazu. Pro zjištění informace, jak dlouho trvá zpracování dotazů pomocí VTapi, byl vytvořen speciální program s příslušnými dotazy. Tento program používá knihovnu s rozhraním VTapi, která obsahuje také podporu pro měření času experimentů. Čas se v tomto případě počítá od sestavování výběrového dotazu do získání výsledků. Vytvoření spojení do tohoto času není zahrnuto stejně jako u systému Secondo.

Pro srovnání výkonnosti systému Secondo a rozhraní VTapi byly použity dva dotazy. Prvním dotazem je získání všech dat z tabulky *Trains* a druhým získání údajů o všech vlacích metra, které projíždějí územím, kde se nachází zoologická zahrada. Tyto dotazy byly zapsány do terminálového rozhraní systému Secondo následovně:

```
query Trains; (1. dotaz)
query Trains feed filter [.Trip passes tiergarten] consume; (2. dotaz)
```

Následující zápis odpovídá druhému zmíněnému dotazu zapsanému v jazyku SQL při použití rozšíření Postgis.

```
SELECT T.Id, T.Line, T.Up, ST_AseWKT(T.Trip), O.area
FROM public.trains T, public.objects O
WHERE ST_Intersects(T.Trip, O.area) AND O.tag = 'tiergarten';
```

## Výsledky a diskuse

Výsledky měření doby trvání dotazů se nachází v tabulce 1. Každé měření proběhlo pětkrát a výsledky byly následně zprůměrovány.

**Tabulka 1.** Doba trvání výběrových dotazů

Doba trvání výběrových dotazů	Secondo	VTapi	Počet řádků
Všechny vlaky metra	3,9884 s	0,2264 s	562
Vlaky metra projíždějící územím zoologické zahrady	0,924 s	0,0744 s	80

Všechny uvedené testy byly provedeny lokálně na stejném počítači s procesorem Intel Core2 E6600, s 3GB operační pamětí, pod 64-bitovým operačním systémem Windows 7 (oba programy jsou však 32-bitové). Pro dotazování pomocí VTapi byla využita 32-bitová verze databáze PostgreSQL na téže lokálním počítači.

Jak je vidět v tabulce 1, dotaz na všechny vlaky metra v tabulce *trains* v případě systému Secondo trvá několikanásobně déle než v případě VTapi. Stejně tak u dotazu na všechny vlaky metra, které projíždějí územím zoologické zahrady je Secondo o jeden řád pomalejší.

## Závěr

V článku byly krátce představeny rozšiřitelný databázový systém Secondo a aplikační programové rozhraní VTapi. Pro testování výkonnosti byla zvolena databáze *berlintest*, která je

součástí systému Secondo. Vybrané objekty databáze byly po svém exportování transformovány pro import do databáze PostgreSQL. Dále byly vybrány dotazy pro účely porovnání. VTapi zabezpečuje dotazy v porovnání se systémem Secondo znatelně rychleji.

Ačkoliv je pro systém Secondo dostupný vývojový softwarový balík (*software development kit, SDK*) a manuál k instalaci určený pro Windows 7, přesto tento systém nebyl bez problémů přeložitelný pod systémem Windows. Systém Secondo se podařilo zprovoznit po několika zásazích do zdrojových kódů. Při dotazování v systému Secondo je třeba dodržovat velikost písmen. Z mého pohledu je velmi nevhodné používat různou konvenci názvů, která značně omezuje a přináší zmatek a chyby při dotazování.

Existují další databázové systémy, které mají podporu pro ukládání prostorových a časoprostorových dat například v podobě rozšíření. Bylo by tedy možné dále porovnávat výkonost s rozšířeními pro databázový systém Oracle, MySQL, Microsoft SQL a další.

### **Poděkování**

Tato práce byla částečně podpořena projektem Nástroje a metody zpracování videa a obrazu pro boj s terorismem (VG20102015006), výzkumným záměrem MSM0021630528, specifickým výzkumem FIT-S-11-2 a Centrem excelence IT4Innovations CZ.1.05/1.1.00/02.0070.

### **Literatura**

- [1.] GÜTING, R. H., BÖHLEN, M. H., ERWING, M., JENSEN, C. S., LORENTZOS, N. A., SCHNEIDER, M., VAZIRGIANNIS, M. *A Foundation for Representing and Querying Moving Objects*. ACM Transactions on Database Systems, 2000, roč. 25, č. 1, s. 1-42.
- [2.] GÜTING, R. H., BEHR, T., DÜNTGEN, C. *SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations*. IEEE Data Engineering Bulletin, 2010, roč. 33, č. 2, s. 56-63.
- [3.] GÜTING, R. H. *How to Build Your Own Moving Objects Database System*. In Proceedings of the 2007 International Conference on Mobile Data Management (MDM '07). IEEE Computer Society, Washington, DC, USA, 2007, s. 1-2.
- [4.] GÜTING, R. H., DIEKER, S., FREUNDORDER, C., BECKER, L., SCHENK, H. *SECONDO/QP: Implementation of a Generic Query Processor*. In Proceedings of the 10th International Conference on Database and Expert Systems Applications (DEXA '99). Springer-Verlag, London, UK, s. 66-87.

### **Abstract**

Moving object databases allow to store and query the objects, which change their position in time. They also may change their position and their extent in time. The time-dependent data can be represented in several different ways. The moving object databases may be useful for storing data (for example, trajectories of movements of objects) from different security and surveillance systems. Data may be also queried later. Not only for these cases, it is needed to process data as quickly as possible. This paper deals with comparing the performance of an extensible database system named Secondo, which uses Berkeley DB as a storage manager, and an application programming interface named VTapi, which uses database system PostgreSQL as a storage manager. The paper provides some basic information about Secondo system and about VTapi interface. It also presents a selection of test data, describes them briefly and provides information about their adjustments for testing purposes. Paper presents selected methods for performance testing and their results. Finally, the results are discussed and the proposals of future directions are mentioned.