



IOPC2



Petr Čermák
Vedoucí RNDr. Michal Kopecky, Ph.D.

Obsah

- ▶ Cíle práce
- ▶ Vývoj knihovny
- ▶ Design a implementace
- ▶ Porovnání s předchozími verzemi
- ▶ Shrnutí a další vývoj



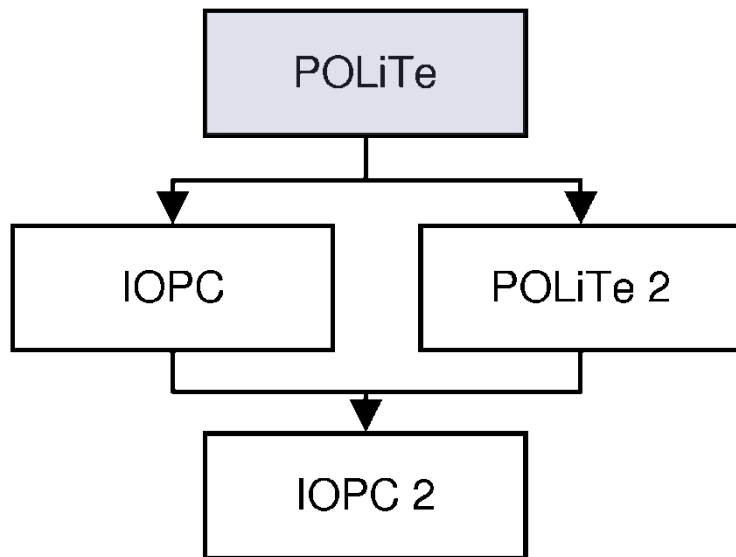
Cíle práce

- ▶ Návrh a implementace knihovny poskytující služby OR mapování v C++
 - ▶ Komfortní, konzistentní rozhraní
 - ▶ Transparentní vývoj uživatelských aplikací
- ▶ Sjednocení vývoje předchozích knihoven
 - ▶ POLiTe, POLiTe2 a IOPC
- ▶ Analýza vlastností ORDBMS a možnosti jejich využití v ORM knihovně.
 - ▶ ADT mapování



Vývoj knihovny

- ▶ Knihovna IOPC 2 vznikla na základě několika knihoven

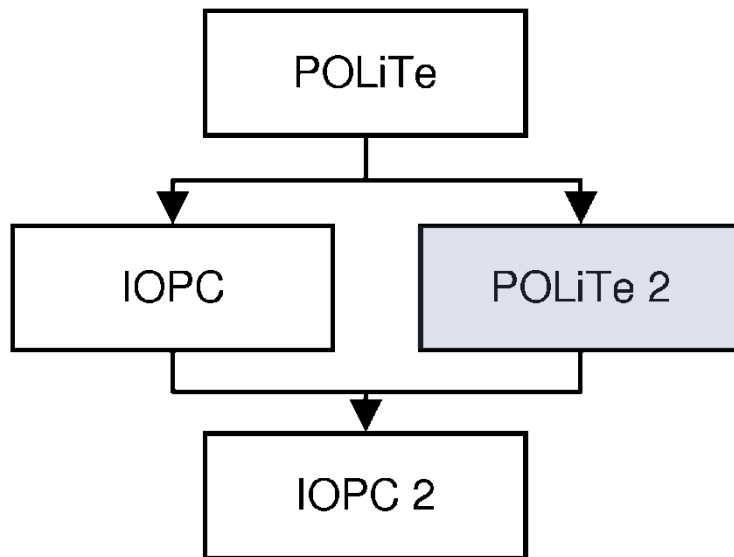


- ▶ Disertační práce RNDr. Michala Kopeckého, Ph.D.
- ▶ Persistence C++ objektů v relačních databázích
- ▶ Třídy popsány pomocí maker
- ▶ Vertikální mapování
- ▶ Persistence asociací mezi objekty
- ▶ Nemodulární architektura



Vývoj knihovny

- ▶ Knihovna IOPC 2 vznikla na základě několika knihoven

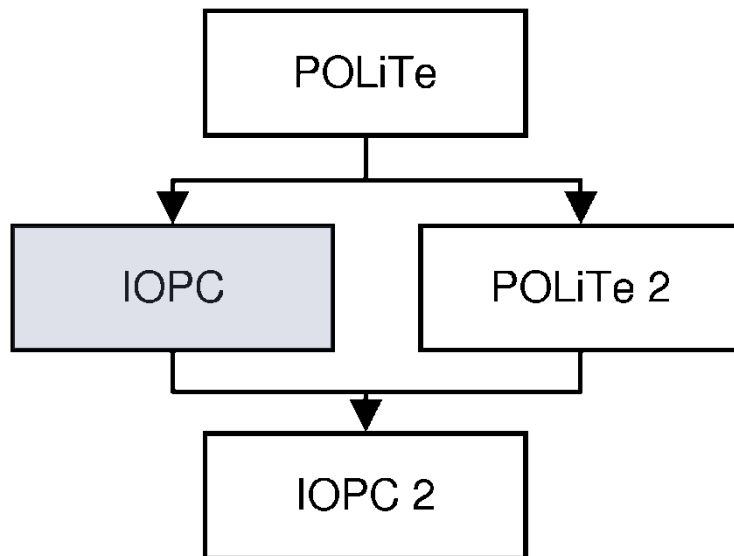


- ▶ Diplomová práce Mgr. Jana Hadravy
- ▶ Zaměřena na zlepšení výkonu knihovny POLiTe
- ▶ Rozšiřitelná cacheovací vrstva
- ▶ Multithreaded prostředí
- ▶ Ostatní vlastnosti stejné jako POLiTe



Vývoj knihovny

- ▶ Knihovna IOPC 2 vznikla na základě několika knihoven



- ▶ Diplomová práce Mgr. Josefa Trocha
- ▶ Zaměřena na transparentní vývoj uživatelských aplikací
 - ▶ Používá source-to-source translator OpenC++
- ▶ Vertikální, horizontální a filtrované mapování
- ▶ Dvě rozhraní - POLiTe a IOPC
- ▶ Mnoho problémů



Design a implementace - cíle

- ▶ **Použít to nejlepší z předchozích knihoven**
 - ▶ Databázová vrstva (POLiTe)
 - ▶ Transparentní vývoj uživatelských aplikací (IOPC)
 - ▶ Cacheovací vrstva (POLiTe 2)
 - ▶ Různé druhy mapování (POLiTe, IOPC)
- ▶ **Přidat další vlastnosti**
 - ▶ Modulární architektura
 - ▶ Reflexe v C++
 - ▶ GCCXML místo OpenC++
 - ▶ OR databáze – ADT mapování
 - ▶ Oracle 10g



Architektura IOPC 2 - konfigurace

iopc common – sdílené služby

iopc meta - reflexe

iopc db – přístup k databázi

Oracle 10g

Oracle 10g OR
rozšíření

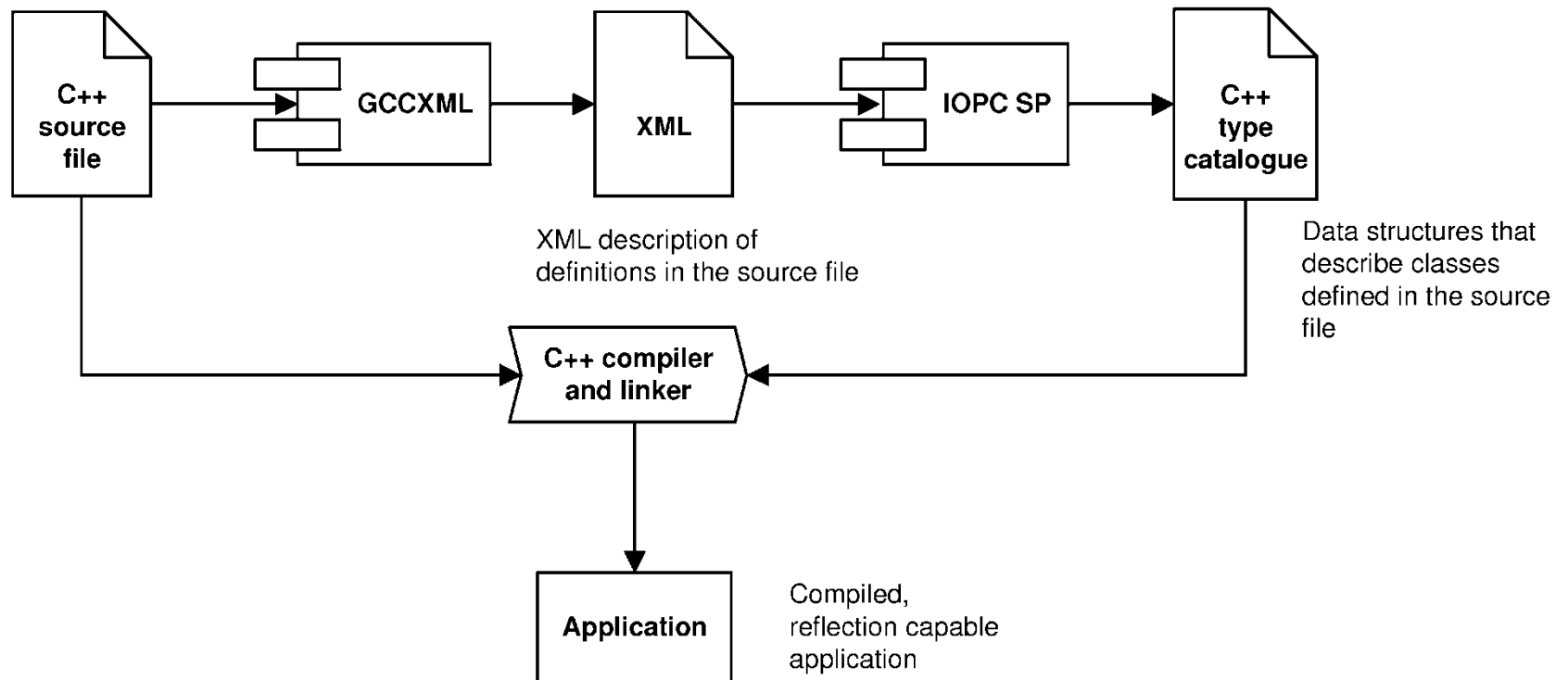
iopc lib – objektově relační mapování

Uživatelská aplikace

1. **Různé služby**
-logování, zámky...
2. **Reflexe**
3. **Databázový přístup**
4. **OR mapování**



Architektura IOPC 2 - workflow



Porovnání s předchozími verzemi

	POLiTe	POLiTe 2	IOPC	IOPC 2
Transparence	- makra	- makra	+ OpenC++	++ GCCXML
Typy mapování	Vertikální	Vertikální	Vertikální, horizontální, filtrované	Vertikální, horizontální, filtrované, ADT
Cachování	Jednoduché	Pokročilé	Jednoduché (chybná impl)	Pokročilé – z POLiTe 2
R/O přístup	+	+	-	+
Architektura	-	-	-	+ Různé konfigurace
Multithreading	-	+	-	+
Perzistence asociací	+ Různé druhy	+ Různé druhy	- Jednoduchá reference	- Jednoduchá reference
Reflexe	-	-	-	+
Podporované databáze	Oracle 7	Oracle 7	Oracle 8i	Oracle 10g

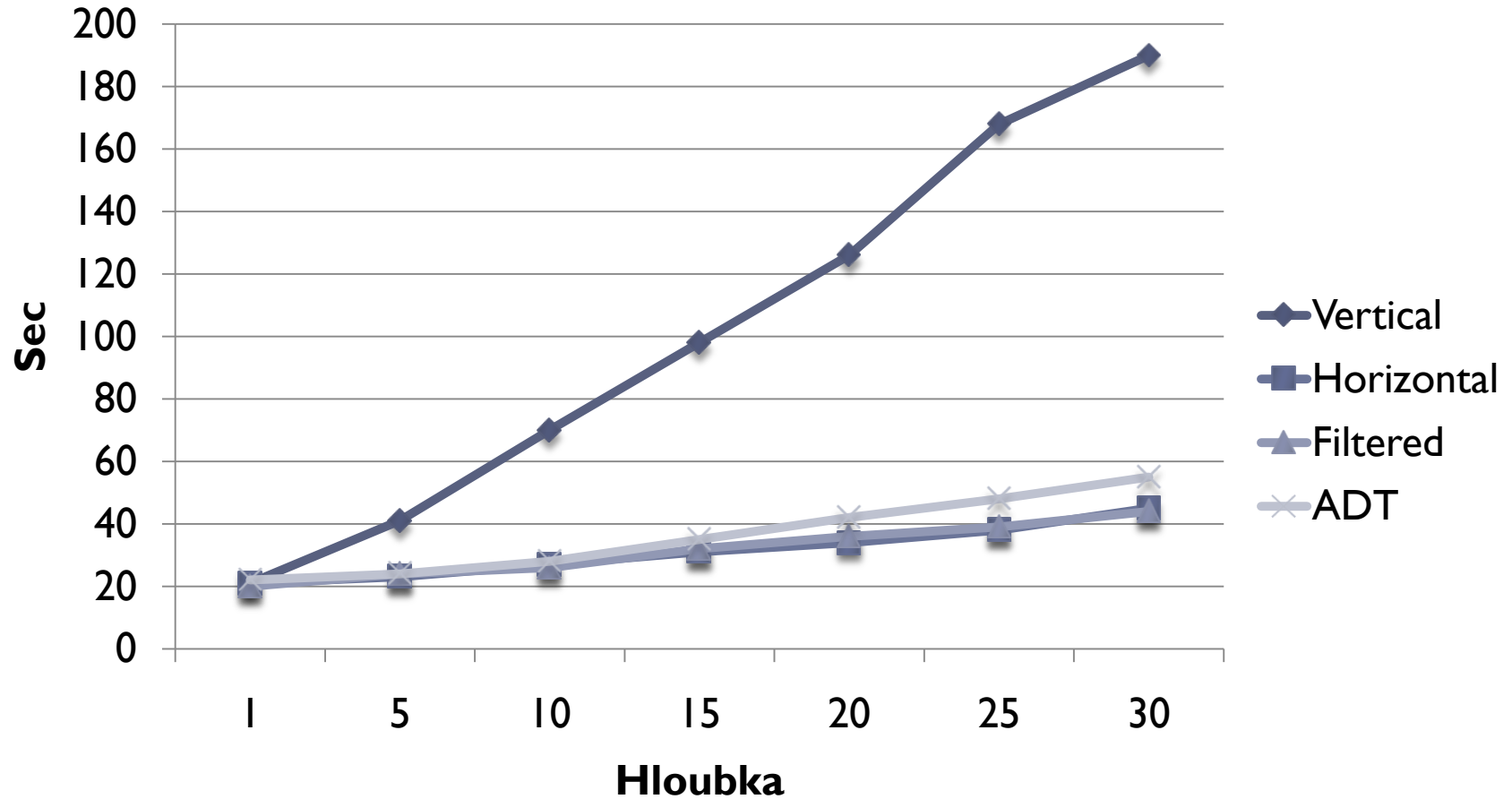
Výhody IOPC 2 a další vývoj

- ▶ Platforma pro jednoduchý a rychlý vývoj db aplikací v C++
- ▶ Transparentní použití
 - ▶ Možnost integrace s IDE
- ▶ Široká konfigurovatelnost
- ▶ Další vývoj
 - ▶ Implementace dalších DB driverů
 - ▶ Rozšíření persistence asociací
 - ▶ Rozšíření reflexe – metody, další typy, ...
 - ▶ Více ORDBMS vlastností do ADT mapování
 - ▶ Vytvoření reálné aplikace používající IOPC2 -> další požadavky
 - ▶ Mnoho dalšího



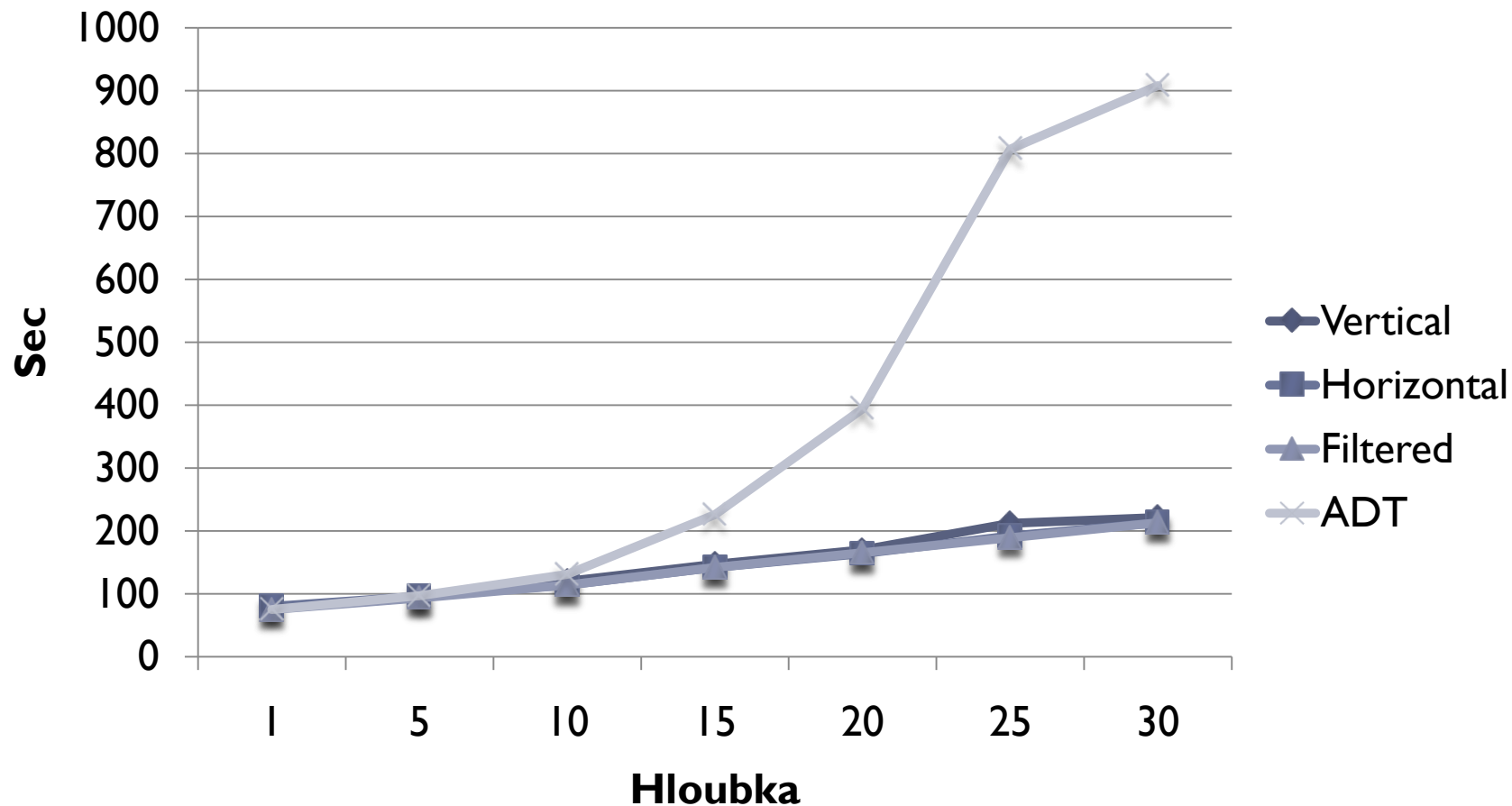
Porovnání výkonu ADT mapování

Insert 10 000 objektů max hloubky



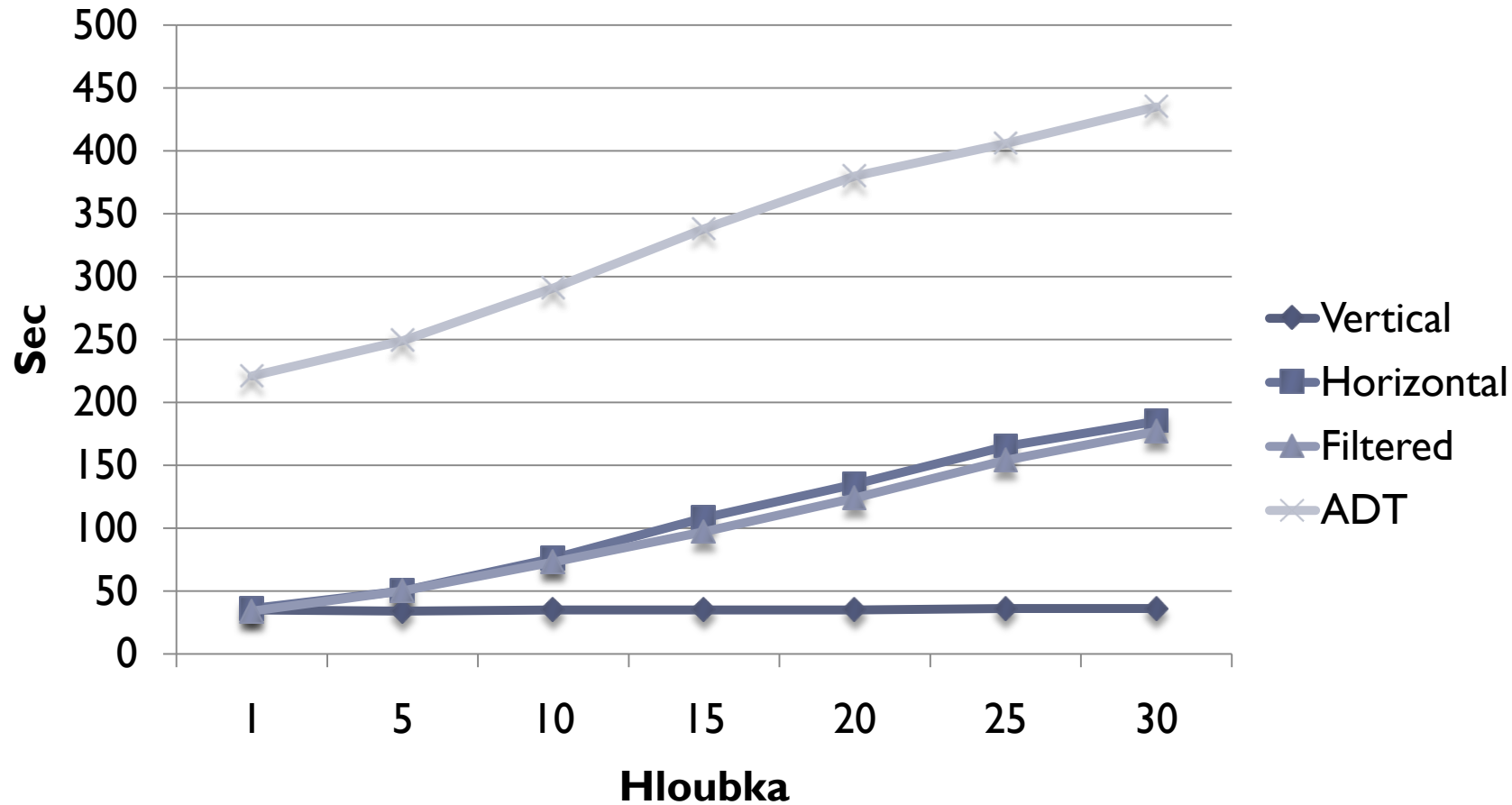
Porovnání výkonu ADT mapování

Load 30 000 objektů po OID



Porovnání výkonu ADT mapování

20 000 polymorfních selectů



Obdobné projekty v C++

- ▶ **LiteSQL** <http://apps.sourceforge.net/trac/litesql/>
 - ▶ XML model -> třídy, schéma
 - ▶ Pouze O/R mapování, žádné cache apod.
- ▶ **Debea Database Access Library** <http://debea.net/trac>
 - ▶ Popis pomocí maker
 - ▶ Opět pouze O/R mapování
- ▶ **Database Template Library** <http://dtemplatelib.sourceforge.net/>
 - ▶ Nutno psát (de)serializační kód
 - ▶ Žádná dědičnost, spíše snadnější způsob jak dostat data z db
- ▶ **HiberLite** <http://code.google.com/p/hiberlite/>
 - ▶ Popis pomocí maker / bindování, základní persistence
- ▶ **SOCI** <http://soci.sourceforge.net/doc/index.html>
 - ▶ Velmi dobrá knihovna pro přístup k DB, podporuje různé databáze
 - ▶ Pro persistenci objektů je třeba psát konverzní můstky, žádná dědičnost
- ▶ Seznam přejet z http://en.wikipedia.org/wiki/List_of_object-relational_mapping_software



Obdobné projekty v C++ - komerční

▶ Objektové databáze

▶ Versant Object Database

http://www.versant.com/en_US/products/objectdatabase

▶ Transparentní persistence

▶ Progress Software ObjectStore

<http://www.progress.com/objectstore/index.ssp>

▶ Transparentní persistence

▶ Matisse <http://www.matisse.com/>

▶ ODL definice tříd

▶ A další...

