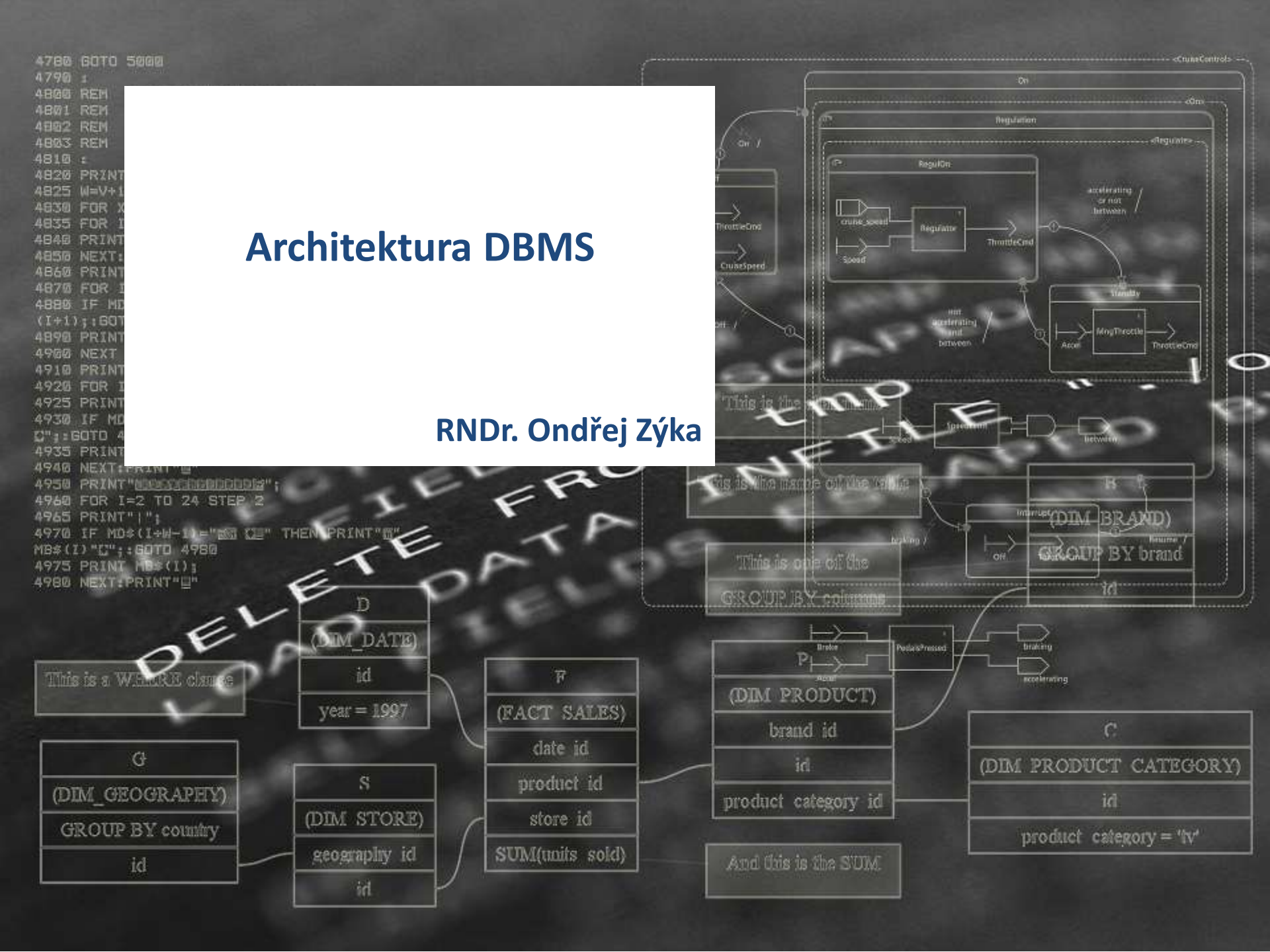
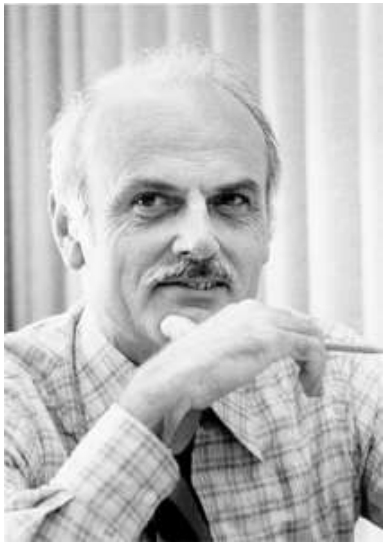
Architektura DBMS

RNDr. Ondřej Zýka



Historie

- Relační model
  - Edgar Frank Codd
  - 1969 - Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks
  - Relační model – matematický model pro ukládání a správu dat
  - Tří hodnotová logika
    - True, False, Unknown
- SQL
  - 1970 - Donald Chamberlin, Raymond F. Boyce **SEQUEL** (**S**tructured **E**nglish **Que**ry **L**anguage) IBM - První návrh
  - 1979 – první komerční implementace Oracle V2 (Relation software)



Historie

1969 – Codd - Relační model

1970 – Chamberlin, Boyce – SQL

1979 – Oracle 2, basic SQL, no transaction

Založení Teradata

1980 – HW - První gigabajtový disk, váha 250 kg, cena $40,000

1983 – Oracle 3 - transaction

1984 – Oracle 4 – read-consistency

1984 – Sybase founded by Mark Hoffman and Bob Epstein in Berkeley

1985 – Oracle 5 – networking, client-server

1986 – HW - Standartizace SCSI

1988 – Oracle 6 – PL/SQL, row level locking, hot backup

1988 – Sybase/Microsoft - sdílení kódu s firmou Microsoft (od roku 86)

Teradata ve spolupráci NCR uvádí databázový počítač

Historie

1991 – HW – 2.5" 100MB disk

1992 – Oracle 7 – referencial integrity, triggers

1993 – Microsoft – Win NT 4.21

1993 – Sybase/Microsoft – ukončení smlouvy

1995 – Microsoft SQL Server 6.0

1998 – Microsoft SQL Server 7.0

1999 – Oracle 8i – java

Teradata- největší zákaznická produkční databáze 130 TB

1999 – HW – IBM 170MB a 340MB disky

2000 – Microsoft SQL Server 2000

2001 – Oracle 9i – XML, RAC

2003 – Oracle 10 – grid computing, flash back

2003 – Windows Server 2003 - 64-bit system - překročení 2GB RAM

Historie

2005 – Sybase 15 – new query-optimalizator, Cluster edition

2005 – Microsoft SQL Server 2005

2005 – HW 500GB disk (Hitachi GST)

2007 – Oracle 11 – Exadata

2007 – HW – 1TB disk (Hitachi GST)

2008 – SQL Server 2008

2009 – HW – SSD – nyní 64 GB 300MB/sec (3000MB/sec.)

2010 – Microsoft SQL Server 2008R2

2010-2013 - Konsolidace trhu

- Oracle kupuje SUN
- SAP kupuje Sybase
- EMC kupuje Greenplum
- IBM kupuje Netezza

2015 – MS a Oracle - Cloud služby pro produkční nasazení

2015 – Gartner hodnotí Mysql a PosgreSQL jako připravené pro produkční nasazení

Cíle a úkoly DBMS

- DBMS – Data Base Management Systém

- Uložit datové struktury
- Zabezpečit data
  - Perzistence uložení
  - Autorizace přístupů
- Realizovat požadavky uživatelů
  - Změna datových struktur
  - Změna dat
  - Dotazy nad daty
  - Monitoring
  - Spouštět kód
- Uživatelský interface

Zdroje DBMS

- Datové prostory
  - Disky
  - Disková pole
  - SAN - Storage area network
  - NAS - Network-attached storage
  - SDD disky
- Paměť
  - RAM
  - Virtual memory
  - SDD disky
- Procesory
  - Univerzální procesory
  - Specializované procesory (Nettezza FPGA)

Zdroje DBMS

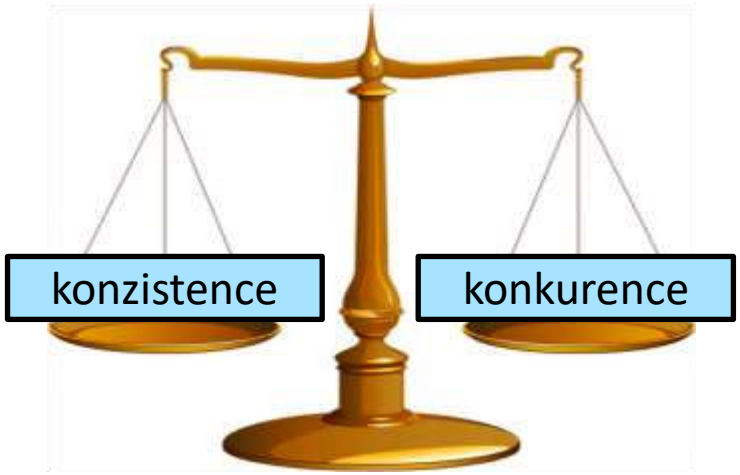
- Sběrnice
  - Komunikace mezi disky a pamětí
  - Komunikace mezi procesory
  - Komunikace mezi node
- Síťové propojení
  - Komunikace s klientem
  - Komunikace mezi geograficky odděleněmi komponentami – například mody clusteru
- Operační systém
  - Správa a přidělování zdrojů
  - Správa procesů a threadů
  - Komunikace s disky
  - Správa paměti
  - Síťová komunikace

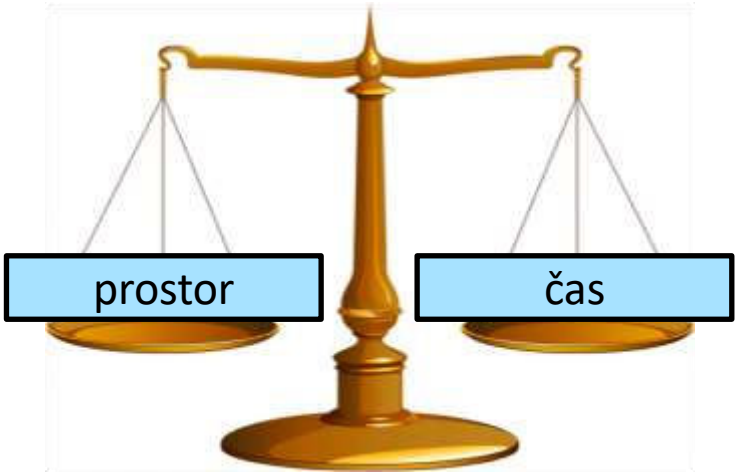
Důležitá čísla

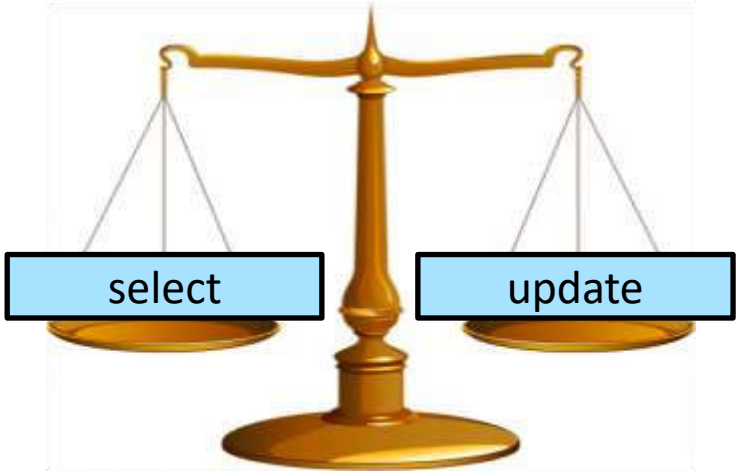
Tick of 3GHz procesor	0,3 ns
L1 cache reference	0,5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	100 ns
Main memory reference	100 ns
Compress 1K butes with Zippy	10 000 ns
Send 2K bytes over 1 Gbps network	20 000 ns
Read 1MB sequentially from memory	250 000 ns
Round trip within same datacenter	500 000 ns
Disk seek	10 000 000 ns
Read 1MB sequentially from network	10 000 000 ns
Read 1MB sequentially from disk	30 000 000 ns
Send packet CA -> Netherland-> CA	150 000 000 ns

Zdroj: Linux Magazine

Kompromisy DBMS







Disky a disková pole

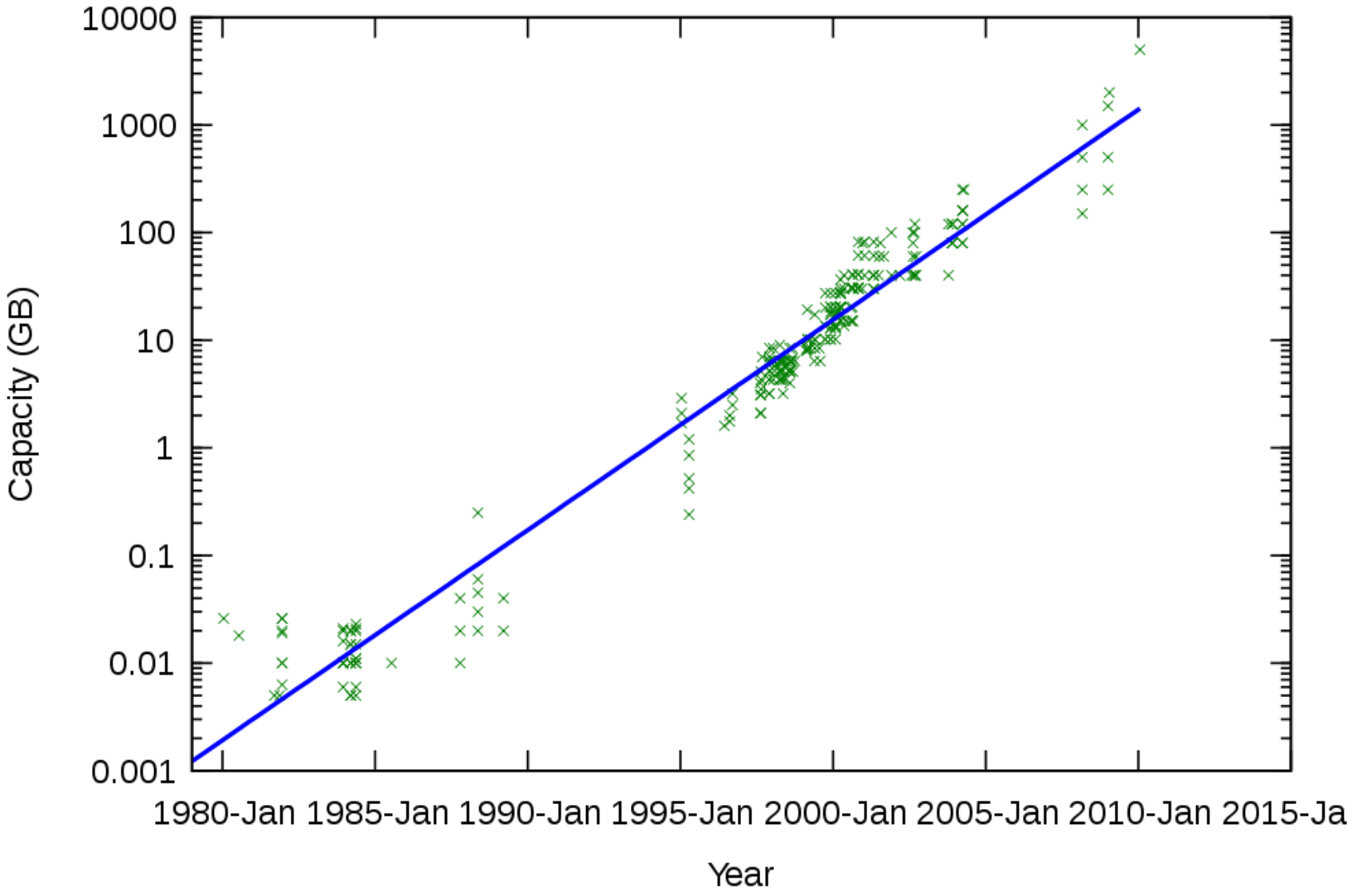
- Dnes kapacita i více než 1TB
  - Stále jenom cca 200 pohybů hlavičky za sekundu
  - Čím více malých disků, tím lépe
- Disková pole
  - Veliké keše
  - Složitý operační systém
  - Mirroring
  - Striping
  - Přenosová rychlost - SATA teoreticky 6GB/sec
- Těžko dohledatelné přesné umístění dat při analýze vzájemného ovlivňování výkonu
  - Diskové pole
  - Cloud

Data write stack

- Je složité (nemožné) identifikovat příčinu chybného zápisu na médium.

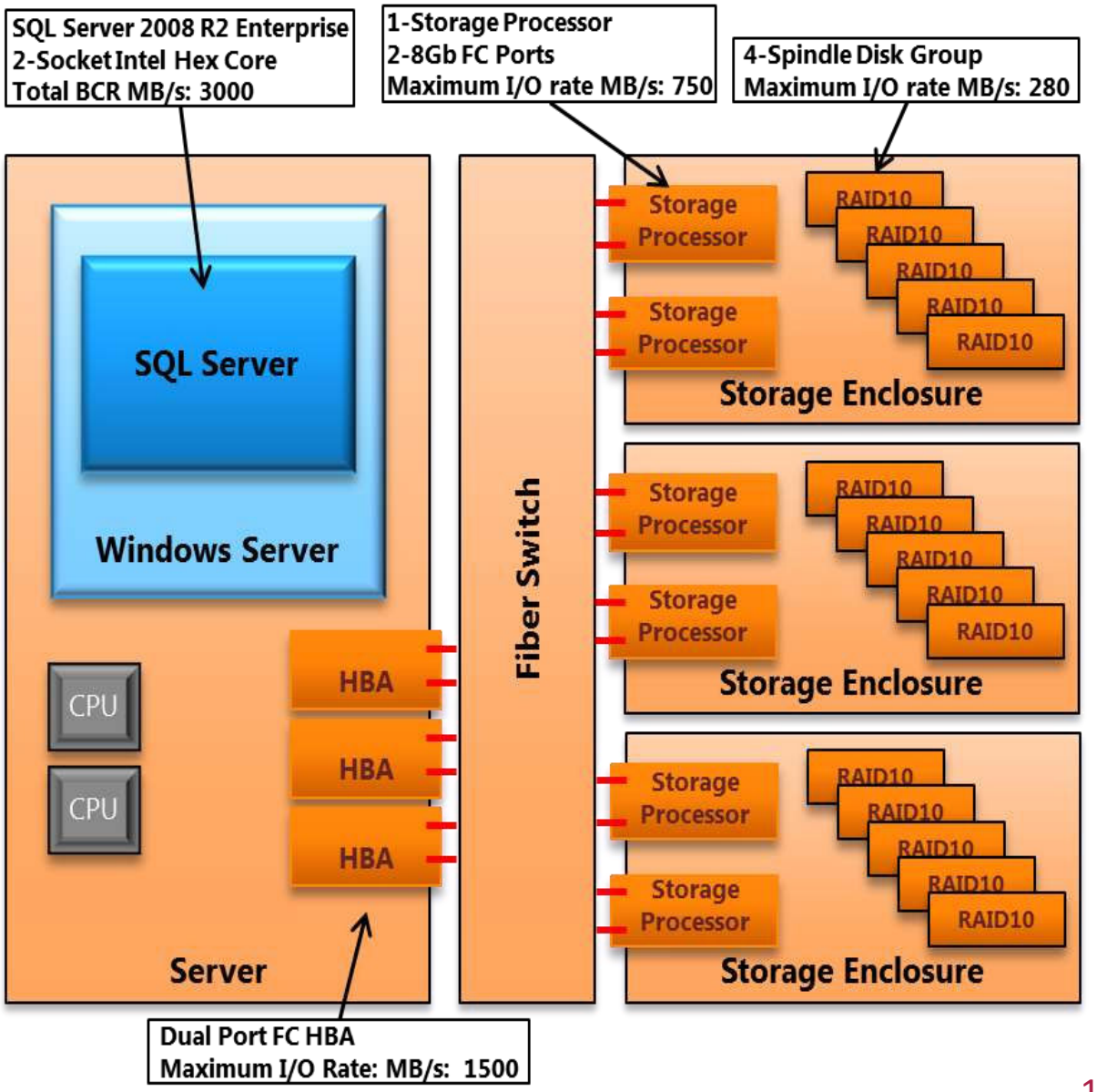
- Datový server dá příkaz zápisu na disk
  - Datový server –> operační systém
  - Operační systém –> souborový systém
  - Souborový systém –> volume manager
  - Volume manager –> device driver
  - Device driver –> Host-Bus-Adapter
  - Host-Bus-Adapter –> Storage controler
  - Storage controler –> Disk
  - Disk OS –> medium
- Uložení dat na NAS pro zjednodušení pomineme

Diskové kapacity (Wikipedia)



Banchmark – Microsoft SQL server 2008 R2

- 2 sokety
- 12 jader
- 3 8Gbps dual-port HBA cards,
- 12 4-disk RAID1+0 primary data LUN



- Více viz Microsoft. Fast Track Data Warehouse 3.0 Reference Guide Published: 4 February 2011

SSD disky

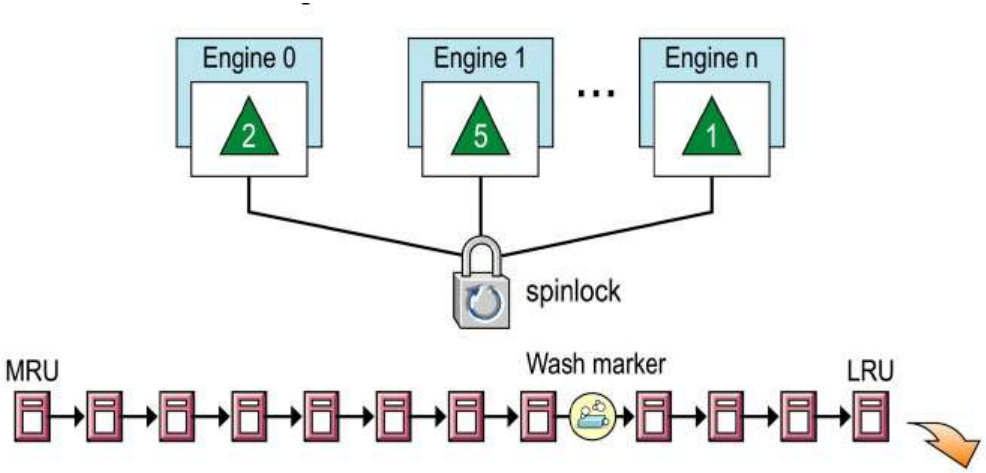
	SSD disk	Disk
Přístupová doba (random access)	0,1 ms	5-10 ms
Přenosová rychlost	300 - 500 MB/s	100 - 160 MB/s
IOPS	20000-10000	75-100
Cena (2011)	1-2 $/GB	0.05-0.1 $/GB
Cena (2016)	0,1 $/GB (1TB Unit)	0.06 $/GB (4GB unit)
Kapacita 2011 (2016)	256GB (2TB)	1TB (4TB)

- Pro standardní relační databáze je výhodnější použít SSD Disky jako velké datové keše než pro uložen dat

Paměť

- Systémové struktury
  - Nastavení
  - Buffery pro třídění dat
  - Buffery pro kód (java)
- Datové buffery
  - Čtení z paměti je mnohonásobně rychlejší
  - Požadovaná data se dají předpovědět na základě struktury uložení dat nebo dotazu
    - Asynchronní načítání dat
    - Načítání celých datových bloků
  - Do bufferů se ukládají
    - Často používaná data
    - Naposledy použitá data
    - Změněná data
  - LRU algoritmus
- Buffer výsledků
- Globální buffery
- Lokální buffery

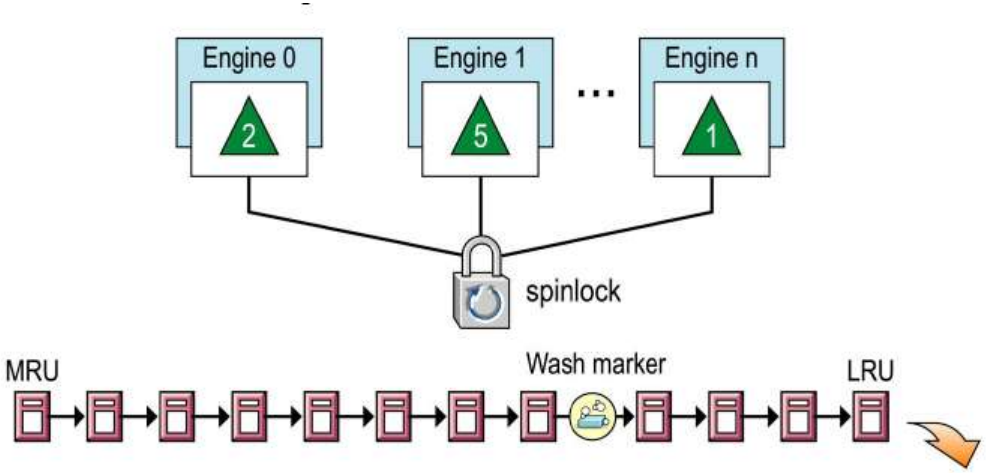
LRU algoritmus



Datová keš řízená LRU algoritmem

- Pro práci mnoha uživatelů s menším množstvím dat
- Eliminace zbytečných zápisu datových bloků na disk
- Stránky uloženy ve spojovém seznamu
- Při přístupu na stránku se stránka přesune na začátek seznamu
- Žádné kopírování stránek, pouze udržování spojového seznamu
- Udržování třech pointerů – Začátek seznamu, konec seznamu a WashMaker

LRU algoritmus



Datová keš řízená LRU algoritmem

- Identifikace zda stránka je v paměti pomocí hash table
- Pokud stránka není, nahradí se stránka na konci seznamu a přesune se na začátek
- Eliminace zápisů pozměněných stránek (dirty pages) na konci seznamu
  - Asynchronní zápis dirty pages pokud překročí Wash maker na disk
- Pro zajištění unikátního přístupu na stránku pro jednotlivé uživatele se používá Spinlock mechanismu
- Eliminace zahlcení keše jedním uživatelem – optimalizator určí že načtené stránky se zařadí za Wash Maker, nikoliv na začátek keše

Procesy

- Procesy vykonávající požadavky klientů

- Další podpůrné procesy
  - Listener
  - Správa diskových prostorů
  - Checkoint process
  - Sběr statistik
  - Monitoring
  - Backup procesy

Implementace procesů

- Procesy operačního systému
  - Řízené operačním systémem
  - Přepínání kontextu na úrovni operačního systému
  - Každý proces vlastní adresní prostor

- Thready operačního systému
  - Jednotný adresní prostor
  - Řízeno operačním systémem

- Vlastní řešení procesů
  - Jeden proces, jednotlivé procesy jsou v něm implementovány interně
  - Jednotný adresní prostor
  - Kontrola nad správou procesů bez nutnosti zásahu OS – nezávislost nad OS

Implementace procesů

- Procesy nebo thready v operačním systému mají větší náročnost při změně kontextu (provádí operační systém)
- Větší závislost na verzi operačním systému – ladění parametrů na úrovni operačního systému
- Při použití vlastního řízení každá synchronní IO operace nebo volání jádra blokuje i ostatní procesy
- Každá chyba v software ovlivní i další procesy

- Výhodné použití databázový server i operační systém od stejného dodavatele

Implementace write-ahead transakčního logu

- Implementace zaručuje durabilitu transakcí - schopnost zotavení při výpadku OS nebo hardware.

- Implementace optimalizuje počet zápisů na disk
  - Log se zapisuje při commit transakce (zajištění durability)
  - Data se zapisují pouze pokud to je nezbytně nutné (viz algoritmus LRU keše)

- Transakční log – datová struktura na úrovni databáze

- Do transakčního logu se zapisuje
  - Začátek a konec transakce (begin tran, rollback, savepoint, commit)
  - Změny v datech - stav před změnou i po změně, místo změny (datová stránka, řádek, …)
  - Změny struktury databáze (DDL)
  - Alokace a dealokace datových stránek – změny interní struktury databáze
  - Záznamy o Checkpoint operaci společně se začátkem nejstarší otevřené transakce v okamžiku checkpointu

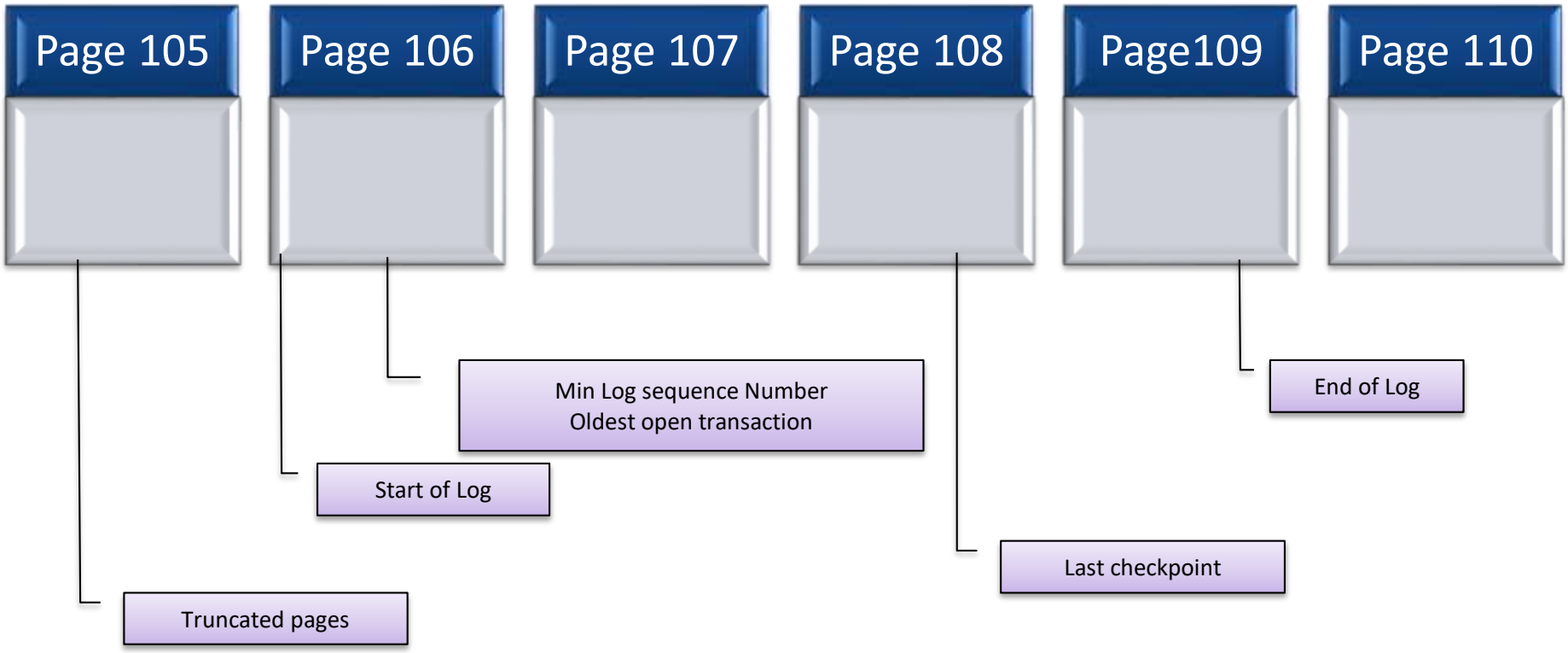
Implementace transakčního logu

- Záznamy do transakčního logu
  - V okamžiku commit transakce musí být zapsán celý obsah transakce, do té doby se změny mohou uchovávat v keších transakčního logu.
  - Změny v transakci musí být zapsány ve správném pořadí.
  - Nemusí být zachováno pořadí změn v rozdílných transakcích.
  - Musí se zachovat pořadí operace Commit

- Příkaz CHECKPOINT
  - Změny v datech se nezapisují okamžitě (Dirty pages) nemusí být zapsány na disk po ukončení transakce.
  - Příkaz Checkpoint vynucuje zápis všech změněných stránek ze všech datových keší

- Rollback
  - V transakčním logu jsou všechny informace nutné k obnově dat do původního stavu.
  - Při rollbacku nemusí být do transakčního logu zapsány všechny změny,
  - Je nutné ale zapsat takové změny, aby byl transakční log konzistetní

Implementace transakčního logu



Implementace transakčního logu

- Postup Zotavení databáze po výpadku - Recovery
  - Najde se poslední Checkpoint
    - Info o Checkointu obsahuje začátek nejstarší otevření transakce v době checkpointu – bod začátku zpracování transakčního logu při Recovery.
- Prochází se log a pro každou transakci v logu (podle času ukončení transakce) se provede:
  - Transakce má commit – ověří se, že data odpovídají konci transakce
    - Pokud ne, nastaví se na koncovou hodnotu transakce.
  - Transakce má rollback – ověří se, že data odpovídají začátku transakce
    - Pokud ne, nastaví se na počáteční hodnotu transakce.
  - Při nalezení konci logu se nenajde konec transakce
    - K transakci se doplní rollback a transakční log se upraví do konzistetního stavu.

Další kompromisy architektury

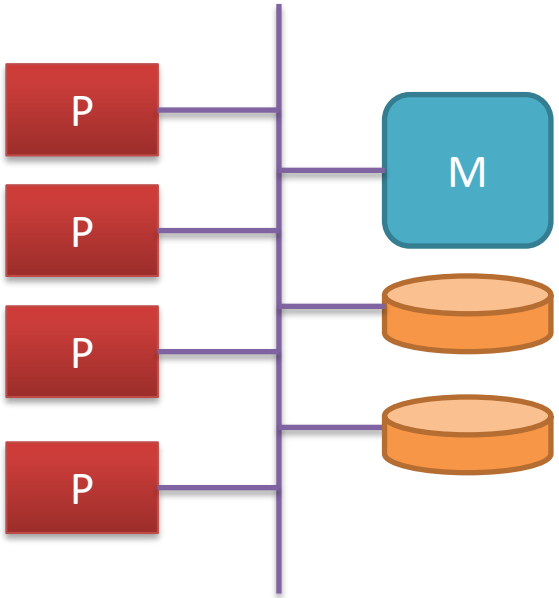
- Operační paměť a IO operace
  - Pokud je jí málo, je vhodné použít malé IO operace
  - Pokud rostou data, je velké množství malých operací limitující
- Operační paměť a počet klientů
  - Pokud je klientů málo, je vhodné maximum informací držet ve vlastním prostoru klienta
  - Pokud je klientů hodně, je nutné maximum informací držet v globálním prostoru serveru
- Operační paměť a rychlost procesoru
  - 4.77 MHz (1981)
  - 3.60 GHz (2009), více úrovní keší
  - Pomalý procesor
    - 1I/O operace odpovídá zpracování 20 stránek v paměti
  - Rychlý procesor přístup na disk 1000-krát „dražší“
    - Vyplatí se udržovat data v paměti
    - In-memory database

Další kompromisy architektury

- Diskové operace a uložení dat
  - Pokud mají být data zpracovány, musí projít procesorem
  - Komprimace dat rychlejší přenos dat z disku do paměti, je náročnější na výkon procesoru
  - Přenesení rozhodování o datech z procesoru na úroveň řadiče disků
  - Požadavek na maximální paralelizaci diskového subsystému

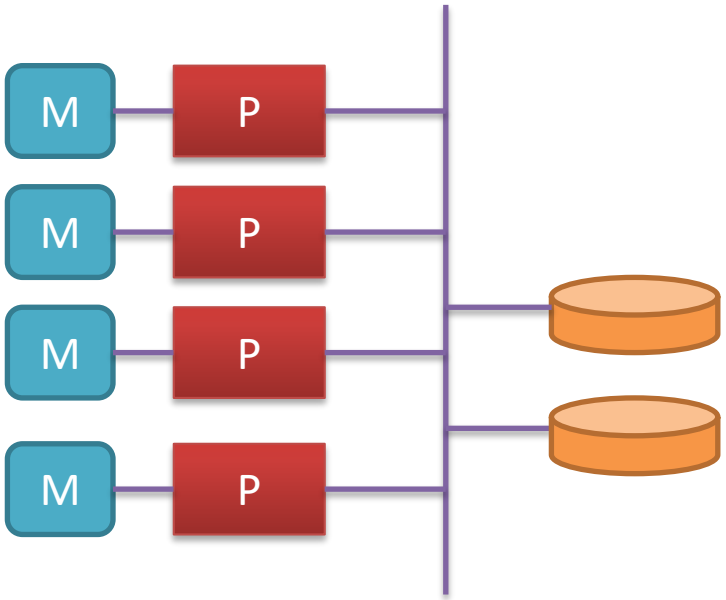
Paralelní architektury

Shared memory



Synchronizace přístupu do datových keší

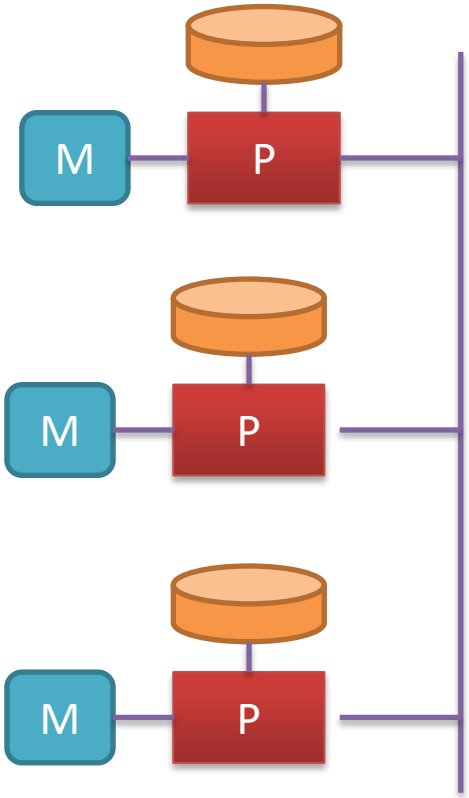
Shared disk



Synchronizace změn v datovém úložišti (datových keší)

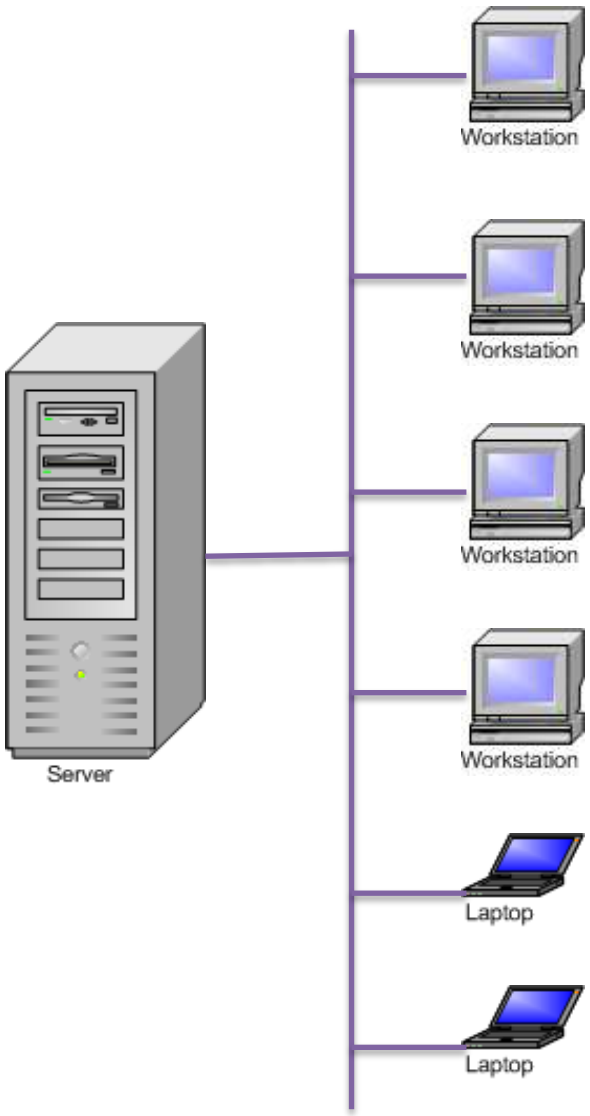
Paralelní architektury

Shared nothing



Distribuce dat potřebných pro výpočty

Klient server



- Jeden server nabízí služby pro mnoho klientů
- Všichni klienti vidí jeden obraz dat
- Klienti si neudržují data
- Klienti předávají pouze příkazy a dostávají výsledky
- Opakem jsou jednouživatelské databáze
  - Microsoft Acces
  - dBase

Klient server

Centralizace

- Modelu
- Dat
- Byznys logiky – funkcionality
- Integrity - ověřování dat
  - Ověření dat na klientovi
  - Duplikace ověření
- Bezpečnostních pravidel
  - Autentizace na úrovni server - aplikace
  - Autorizace na úrovni aplikace
    - Role na úrovni serveru
- Nasazení změn
  - Nutnost podporovat více verzí klientů

Životní cyklus uživatelského požadavku

- Navázání spojení s klientem
- Porozumění požadavku
- Optimalizace a vytvoření výpočtu
- Vlastní výpočet
- Předání výsledků

Navázání spojení s klientem

- Klientský software (aplikační software, ODBC, JDBC, …)
- Adresářové služby pro nalezení serveru
- Informace o uživateli – identifikace, časové pásmo, kódová stránka, formát čísel a data
- Vytvoření klientského procesu
- Navázání spojení
- Vytvoření procesu spravující klientské připojení
- Alokace struktur pro správu klientského připojení
  - Síťová komunikace
  - Prostor pro výsledky
  - Lokální prostor pro výpočty
  - Prostor pro uživatelská data

Porozumění požadavku

- Parser – syntaktická analýza
- Mapování na objekty v databázi
- Shromáždění informací o použitých objektech
  - Ověření práv
  - Informace o struktuře
    - Struktura tabulek
    - Indexy
    - Partitioning
    - Umístění tabulek na discích
    - Statistiky

Vytvoření algoritmu výpočtu

- Sémantická analýza
- Identifikace požadovaných konstruktů
- Vytvoření algoritmu výpočtu
- Optimalizace na základě sémantiky
  - Transakční uzávěr
  - Datové konstanty
  - Boolean logic optimalization
  - Vnořený select – join
- Optimalizace na základě dat
  - Existence indexů
  - Hodnoty statistik
- Optimalizace na základě obdobných výpočtů
  - Existující výpočetní plány

Vlastní výpočet

- Uživatelský proces
  - Execution tree
  - Execution engine
- Databázový proces
  - Asynchronní čtení dat
  - Použití datových keší

- Reakce na vlastní výpočet
  - Query recompilation

Předání výsledků

- Datasety
  - Definice
  - Data
- Cursory
- Chyby
- Zprávy

Příklady

Oracle history

1972 - Oracle 2, basic SQL, no transaction

1983 – Version 3 - transaction

1984 – Version 4 – read- consistency

1985 – Version 5 – networking, client-server

1988 – Version 6 – PL/SQL, row level locking, hot backup

1992 – Version 7 – referencial integrity, triggers

1999 – Version 8i – java

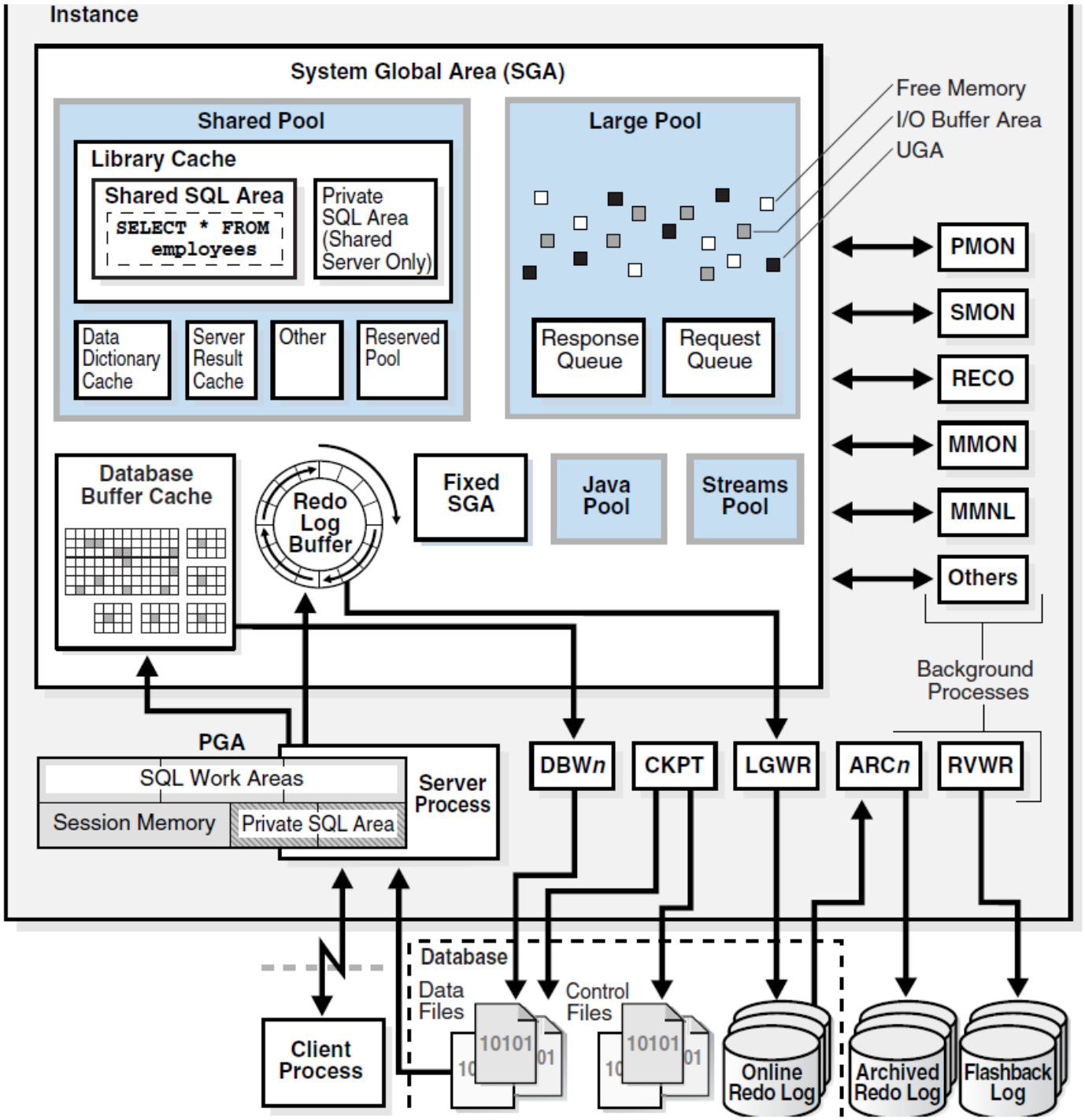
2001 – Version 9i – XML, RAC

2003 – Version 10 – grid computing, flash back

2007 – Version 11 - Exadata

2013 – Version 12 - Pluggable Databases

2017 – Version 12.2.0.1


Instance
System Global Area (SGA)
Shared Pool
Library Cache
Shared SQL Area
SELECT * FROM employees
Private SQL Area (Shared Server Only)
Data Dictionary Cache
Server Result Cache
Other
Reserved Pool
Large Pool
Response Queue
Request Queue
Free Memory
I/O Buffer Area
UGA
Database Buffer Cache
Redo Log Buffer
Fixed SGA
Java Pool
Streams Pool
PMON
SMON
RECO
MMON
MMNL
Others
Background Processes
PGA
SQL Work Areas
Session Memory
Private SQL Area
Server Process
DBWn
CKPT
LGWR
ARCn
RVWR
Client Process
Database
Data Files
Control Files
Online Redo Log
Archived Redo Log
Flashback Log

Oracle – hlavní body architektury

ORACLE DATABASE

- Oddělený listener
- Procesy operačního systému
  - Process Monitor Process (PMON)
  - System Monitor Process (SMON)
  - Database Writer Process (DBWn)
  - Log Writer Process (LGWR)
  - Checkpoint Process (CKPT)
  - Manageability Monitor Processes (MMON and MMNL)
  - Recoverer Process (RECO)
- Diskové prostory
  - Dataspaces
    - User
    - Systém
    - Temp
  - On-line redolog
  - Archive redolog
  - Flashback log

Oracle

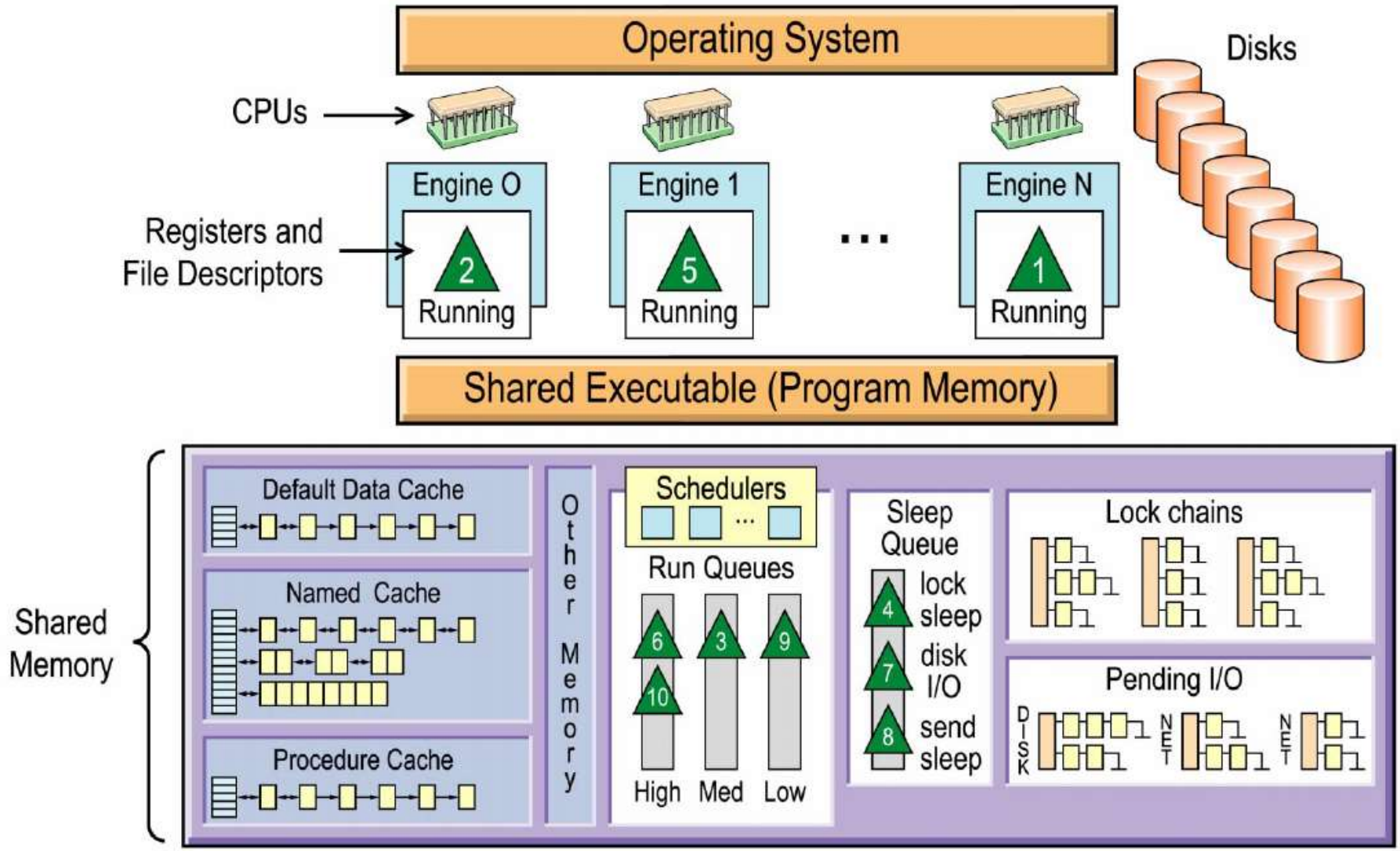
ORACLE DATABASE

- SGA – Systém Global Area
  - Database Buffer Cache
  - Redo Log Buffer
  - Shared Pool
  - Large Pool
  - Java Pool
  - Streams Pool
  - Fixed SGA

- PGA – Program Global Area
  - SQL Work area – Sort, Hash, Bitmap merge
  - Private SQL Area – Session memory, Persitent area, Runtime Area

Sybase – hlavní body architektury

- Engine
  - Proces na procesoru
- User processes
  - Queue
- Shared memory
  - Stránky
    - Datové, logu
  - Keše
    - Datové
    - Kódu
- Serverové struktury
- Diskové prostory
  - Data
  - Transakční logy


Operating System
Disks
CPUs
Engine 0
Engine 1
Engine N
Registers and
File Descriptors
2
Running
5
Running
1
Running
Shared Executable (Program Memory)
Shared
Memory
Default Data Cache
Named Cache
Procedure Cache
Other Memory
Schedulers
Run Queues
6
10
3
9
High
Med
Low
Sleep
Queue
4
lock
sleep
7
disk
I/O
8
send
sleep
Lock chains
Pending I/O
DISK
NET
NET

Limity databází

Microsoft SQL Server 2008

SQL Server Database Engine object	Maximum sizes/numbers SQL Server (32-bit)
Bytes per short string column	8,000
Bytes per GROUP BY, ORDER BY	8,060
Bytes per index, foreign, primary key	900
Bytes per row[8]	8,060
Bytes per **varchar(max)**, **varbinary(max)**, **xml**, **text**, or **image** column	2^31-1
Columns in GROUP BY, ORDER BY	Limited only by number of bytes
Columns per index, foreign, primary key	16
Columns per nonwide/wide table	1,024/30000
Columns per SELECT/INSERT statement	4096
Database size	524,272 terabytes
Databases per instance of SQL Server	32,767
Foreign key table references per table[4]	253
Identifier length (in characters)	128
Locks per instance of SQL Server[5]	Up to 2,147,483,647

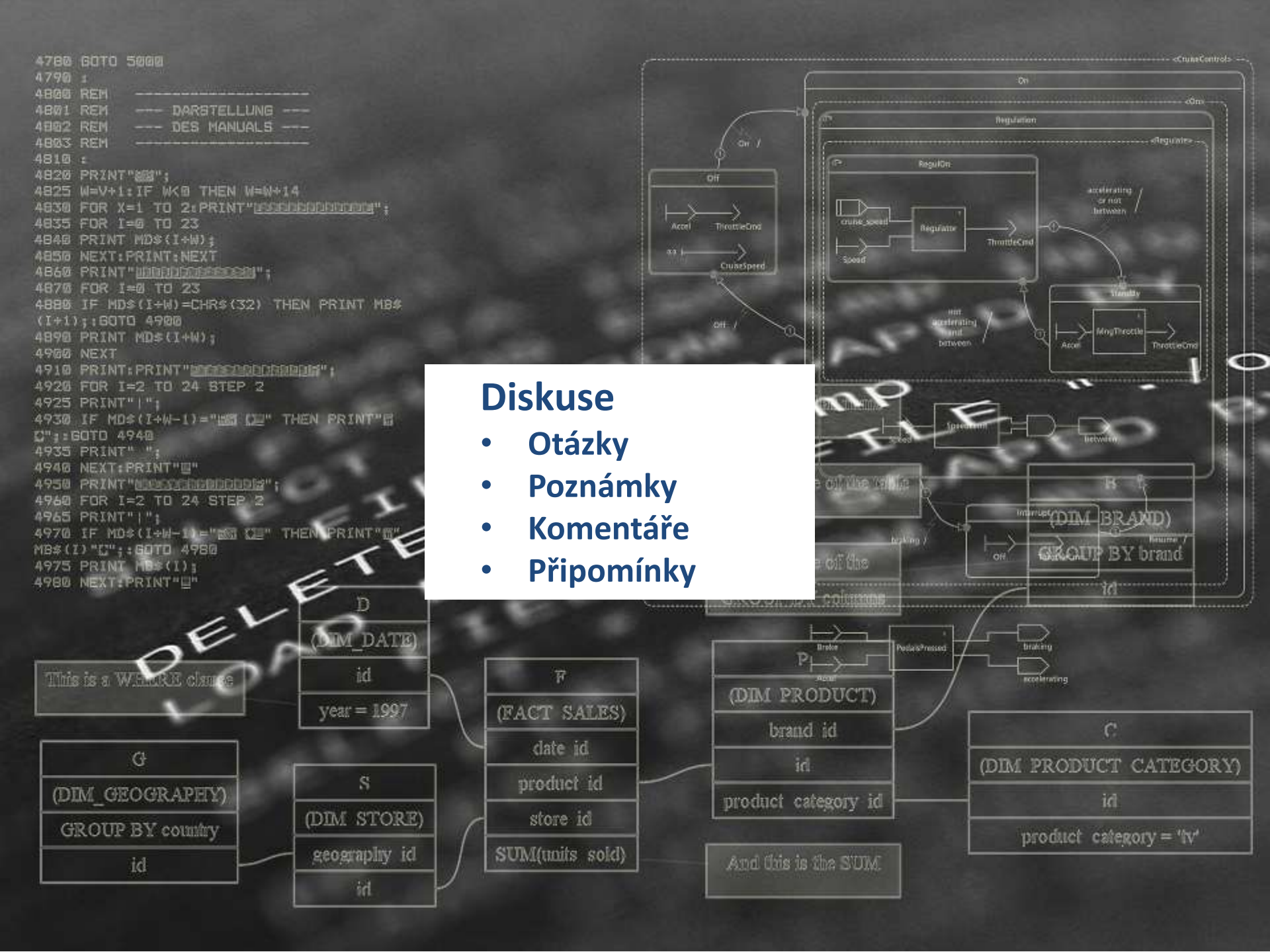
Limity databází

Microsoft SQL Server 2008

SQL Server Database Engine object	Maximum sizes/numbers SQL Server (32-bit)
Nested stored procedure levels[6]	32
Nested subqueries	32
Nested trigger levels	32
Nonclustered indexes per table	999
Parameters per stored procedure	2,100
Parameters per user-defined function	2,100
REFERENCES per table	253
Rows per table	Limited by available storage
Tables per database[3]	Limited by number of objects in a database
Partitions per partitioned table or index	1,000
Tables per SELECT statement	Limited only by available resources
Triggers per table[3]	Limited by number of objects in a database
Columns per UPDATE statement (Wide Tables)	4096
User connections	32,767

Co si zapamatovat

- Které zdroje HW a OS využívají databázové systémy
- Jak pracuje datová keš typu LRU
- Co to je write-ahead log model a k čemu slouží
- Rozdíly mezi shared memory a shared nothing architekturou
- Které všechny služby centralizuje architektura Klient-server
- Jaké jsou základní kroky životního cyklu dotazu
- Jak probíhá navázání spojení mezi serverem a klientem


4780 GOTO 5000
4790 :
4800 REM ----------
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM ----------
4810 :
4820 PRINT"
4825 W=V+1:IF W<0 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT MB$
(I+1);:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)="
4935 PRINT" ";
4940 NEXT:PRINT"
4950 PRINT"
4960 FOR I=2 TO 24 STEP 2
4965 PRINT"|";
4970 IF MD$(I+W-1)="
MB$(I)"
4975 PRINT MB$(I);
4980 NEXT:PRINT"
Diskuse
Otázky
Poznámky
Komentáře
Připomínky
D
(DIM_DATE)
id
year = 1997
This is a WHERE clause
G
(DIM_GEOGRAPHY)
GROUP BY country
id
S
(DIM_STORE)
geography_id
id
F
(FACT_SALES)
date_id
product_id
store_id
SUM(units_sold)
P
(DIM_PRODUCT)
brand_id
id
product_category_id
And this is the SUM
C
(DIM_PRODUCT_CATEGORY)
id
product_category = 'tv'
(DIM_BRAND)
GROUP BY brand
id