

Optimalizace výkonu docházkového serveru pro firmy se stovkami či tisíci zaměstnanců

Tato příručka popisuje základní i pokročilé možnosti dostupných úprav konfigurace docházkového systému a optimalizace výkonu docházkového serveru tak, aby byla odezva pro uživatele rychlá i v případě, kdy s programem pracují současně v jeden okamžik stovky zaměstnanců. Například když na přelomu měsíce provádí vedoucí pracovníci kontrolu docházky podřízených, opravy chyb čipování, dohrávají celodenní absence, vytváří a tisknou docházkové sestavy, účetní provádí uzávěrku, tisk sestav, exporty do mezd a i jednotliví pracovníci kontrolují přes webové rozhraní svou vlastní docházku.

U malých firem, kde s docházkou přes webové rozhraní pracuje jen malý počet zaměstnanců, je možné systém provozovat na libovolném PC vyrobeném po roce 2000. Stačí tedy i jednodprocesorový jedno-jádrový počítač s taktem 1 GHz. Hlavní vliv na rychlost systému má právě rychlost CPU, při současném přístupu více uživatelů i počet jader, velikost cache paměti a rychlost přístupu do RAM. Velikost RAM ani velikost či rychlost disku nejsou pro odezvu docházky tak důležité. Samozřejmě nesmí docházet ke swapování (nedostatek RAM).

Základní minimální konfiguraci HW docházkového serveru v závislosti na počtu pracovníků vyjadřuje tato tabulka:

Počet zaměstnanců	Procesor			RAM GB	Disk GB
	bitů	jader	GHz		
10	32	1	0,5	0.25	3
30	32	1	1	0.5	5
100	32	2	2	1	10
300	64	2	2	1	15
1000	64	4	3	2	30

Z ní je patrné, že jako docházkový server pro firmu s max. tisícovkou zaměstnanců lze použít i dnes běžně dostupný obyčejný kancelářský počítač. Pokud je zaměstnanců výrazně více, je možné použít buď výkonný server, nebo lze vytvořit cluster ze dvou či více běžně dostupných počítačů. Opět viz následující tabulka:

Cluster z více serverů							
Počet zaměstnanců	Appl. serverů	DB serverů	Procesor			RAM GB	Disk GB
			bitů	jader	GHz		
2000	2	1	64	2	2	2	20
4000	2	1	64	4	2	2	40
7000	4	1	64	4	3	4	60
10000	4	2	64	8	3	8	100

U konfigurací s tisíci a více pracovníky se již doporučuje použití SSD disků. Operační systém Windows ve verzi server (např. Windows Server 2003 a novější). Použití operačního systému ve verzi pro pracovní stanice nevyhovuje při práci více uživatelů licencí ani výkonem.

A) Základní optimalizace výkonu

1. Kontrola nastavení programu:

Pokud tedy z předchozích uvedených tabulek vidíte, že konfigurace HW vašeho docházkového serveru by měla bez problému zvládnout správu docházky pro firmu s odpovídajícím počtem zaměstnanců a přesto se vám zdá odezva systému pomalá, může být problém v chybném nastavení parametrů v editaci firmy. Oprava je popsána v příručce „Řešení problémů_SW.pdf“ kterou najdete na CD docházky ve složce \Prirucky a zde viz bod 4. Pomalá odezva systému.

2. Používání cache paměti DB serverem:

Dále může být problém v tom, že služba MySQL ve vaší konfiguraci nemá povoleno používání cache paměti. Na hlavním PC docházky (docházkovém serveru) otevřete pomocí *Poznámkového bloku* či jiného jednoduchého textového editoru soubor

C:\apache\mysql\bin\my.ini a do sekce [mysqld] dopište tyto 3 řádky, pokud tam již nejsou:

```
query_cache_size = 33554432
```

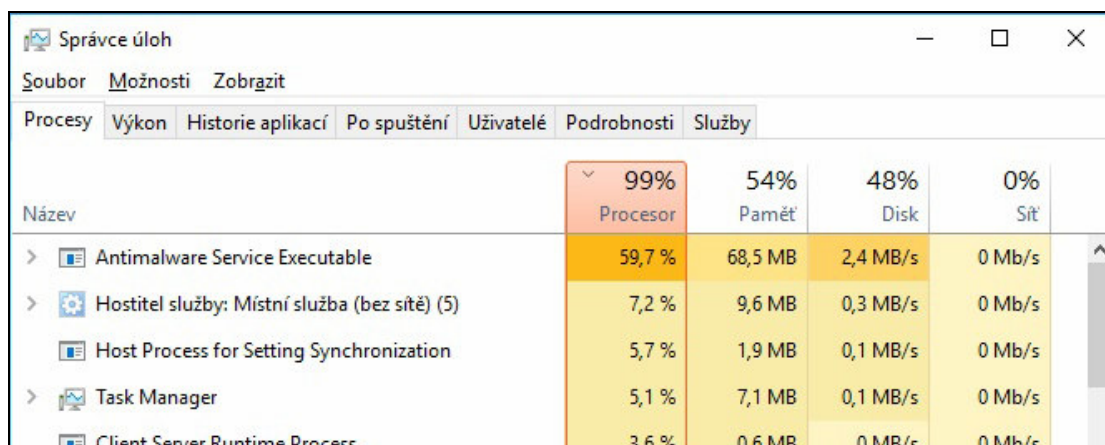
```
query_cache_type=1
```

```
query_cache_limit=1048576
```

A pokud bude do programu přistupovat mnoho uživatelů, přidejte `max_connections = 1900`. Poté soubor uložte a restartujte počítač, nebo alespoň službu MySQL. Databázový server upraví používání vyrovnávací paměti cache, čímž dojde ke zrychlení provádění některých databázových operací, které mají velký vliv právě u výpočetně náročnějších sestav, ale urychlit může třeba i načítání dat z terminálů atd. Používejte serverové verze Windows.

3. Vytížení jinými programy:

Výkon docházkového serveru mohou spotřebovávat také jiné na něm nainstalované aplikace. Takže když na serveru běží další programy a služby náročné na výkon, je procesor vytížený jinou činností a o výpočetní čas se tak docházkový systém dělí s těmito jinými úlohami. Což se projeví právě na pomalejší reakci programu při práci uživatelů. To, který program server právě zatěžuje, zjistíte přes správce úloh, kde na kartě *Procesy* vidíte celkové zatížení CPU i to, který proces nejvíce zatěžuje procesor. Viz ukázka:



Název	Procesor	Paměť	Disk	Síť
> Antimalware Service Executable	59,7 %	68,5 MB	2,4 MB/s	0 Mb/s
> Hostitel služby: Místní služba (bez sítě) (5)	7,2 %	9,6 MB	0,3 MB/s	0 Mb/s
Host Process for Setting Synchronization	5,7 %	1,9 MB	0,1 MB/s	0 Mb/s
> Task Manager	5,1 %	7,1 MB	0,1 MB/s	0 Mb/s
Client Server Runtime Process	3,6 %	0,6 MB	0 MB/s	0 Mb/s

Na ukázce je vidět, že zátěž procesoru je momentálně maximální (99%) a že je v tomto konkrétním případě způsobeno antivirovým programem, který právě provádí kontrolu systému. V klidu by měla být zátěž buď téměř nulová nebo pouze v jednotkách procent.

Možným řešením je buď navýšení výkonu serveru (upgrade HW) nebo přesun docházky na samostatný server, na kterém nebudou běžet jiné výkonově náročné úlohy.

4. Měření výkonu serveru:

V programu lze v menu „Firma / Databáze / Test databáze“ nalézt tři různé testy výkonu serveru.

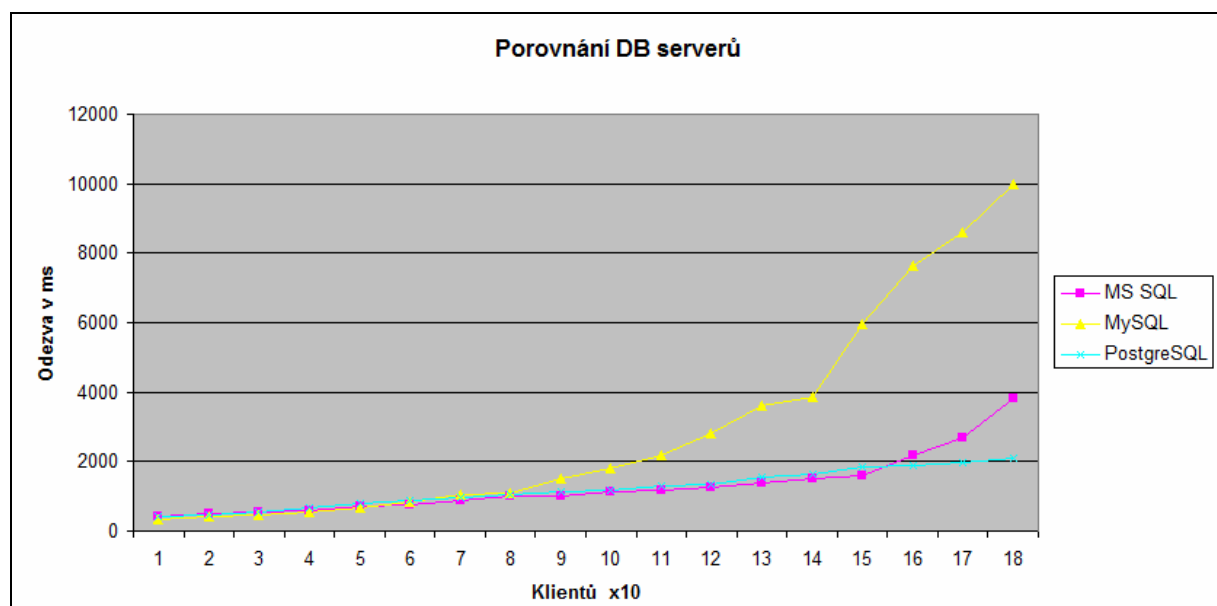
A) První test dostupný přes odkaz „Test rychlosti DB/HDD“ je i popsán v příručce pro řešení problémů SW odkazované na předchozí straně v bodě 1 a pomůže vám zjistit, pro evidenci kolika zaměstnanců je HW konfigurace vašeho docházkového serveru vhodná.

B) Druhý test dostupný přes odkaz „Test rychlosti GUI“ se hodí i pro případ, kdy docházkový server zároveň slouží i pro práci s docházkou. Po spuštění běží zvolený typ testu v nekonečné smyčce a zobrazuje vybrané sestavy. Nechte test běžet třeba jednu minutu a poté opět přejděte do menu „Firma / Databáze / Test databáze / Test rychlosti GUI“. Zobrazí se naměřené hodnoty a jejich porovnání s HW konfiguracemi počítačů vyráběných v různých letech. Hodnoty vašeho PC by měly zhruba odpovídat počítači v tabulce s podobnou konfigurací. V opačném případě projděte body A1 až A3 této příručky.

C) Třetí test dostupný přes odkaz „Test škálování“ je popsán v příručce „Řešení problémů_SW.pdf“ kterou najdete na CD docházky ve složce \Prirucky a zde viz bod 15. Test výkonu hlavního PC docházky. Test je možné spustit z více počítačů (klientů) a první počítač pak slouží jako vyhodnocovací, který zobrazuje i výsledky. Ostatní počítače jsou podřízené a slouží k simulaci současné práce více zaměstnanců s docházkovým programem. Tento test vám pomůže právě pro zjištění výsledného efektu při provádění optimalizací výkonu docházkového serveru nebo clusteru serverů či konfiguraci virtuálu.

5. Změna DB serveru:

Docházka se standardně dodává ve verzi pro MySQL 5.1.37 databázový server. Jsou však k dispozici i varanty pro MS SQL server, PostgreSQL a Firebird. Byla provedena porovnávací měření pro první tři jmenované SQL servery. Konfigurace měření vychází z předchozího bodu 4C, kdy byly použity 3 klientské počítače, na každém postupně spuštěno 6 vláken po 10 měřeních v každém vlákně. Celkem test odpovídá zhruba 180 s programem současně pracujícím uživatelům. Výsledky měření na starším HW jsou na tomto grafu:



Z grafu je patrné, že do 80 současně pracujících uživatelů není mezi jednotlivými DB servery významnější výkonový rozdíl. Odezva pomalu plynule narůstá u všech serverů zhruba stejným tempem. Až od 90 uživatelů začne být MySQL server pomalejší než dva zbývající a od 160 uživatelů začne zpomalovat i MS SQL server. Nejlépe tedy s docházkou škáluje PostgreSQL server, který má stále zhruba lineární průběh odezvy.

I přes uvedené skutečnosti se ale doporučuje firmám do 1000 zaměstnanců používat docházku s výchozí MySQL databází 5.1.37, protože pro ní je k dispozici plná podpora výrobce a i příručky pro zálohování a řešení problémů předpokládají použití této databáze.

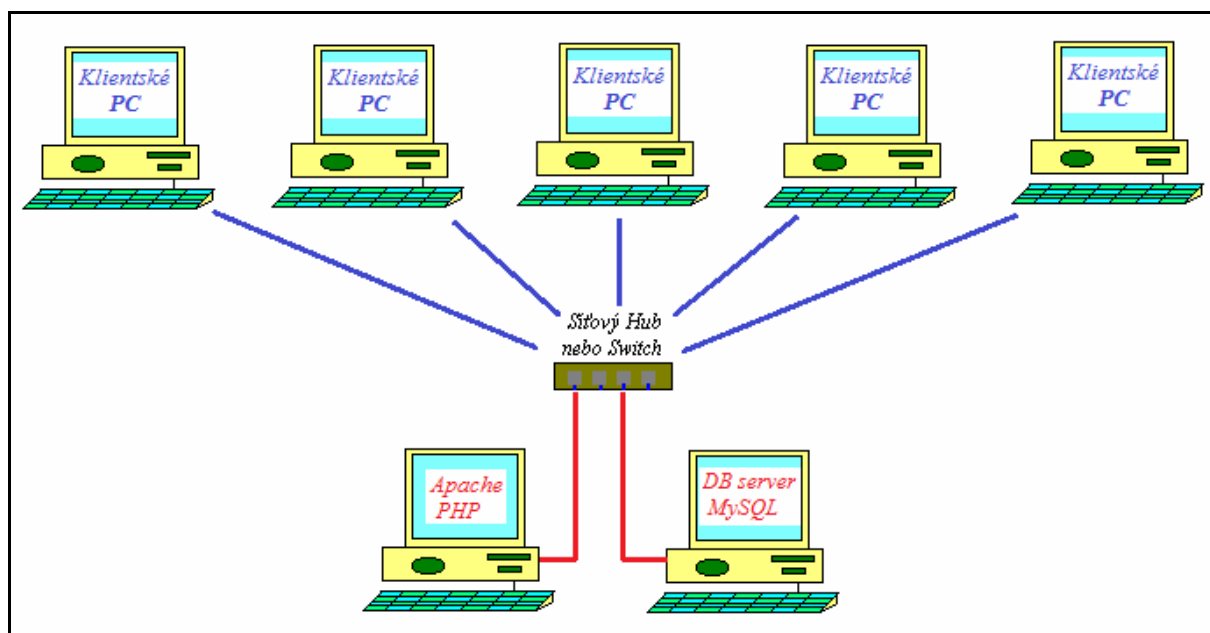
6) Aktualizace programu:

Program je vhodné udržovat aktualizovaný. Například ve verzi 9.98 byly doplněny výrazné optimalizace výpočtu exportů (např. do mezd či do Excelu) a hromadných sestav jako výkaz, výsledovka, přehledka a podobně. Aktualizaci objedná administrátor přes položku *E-shop*.

B) Cluster z více serverů – výběr vhodné konfigurace

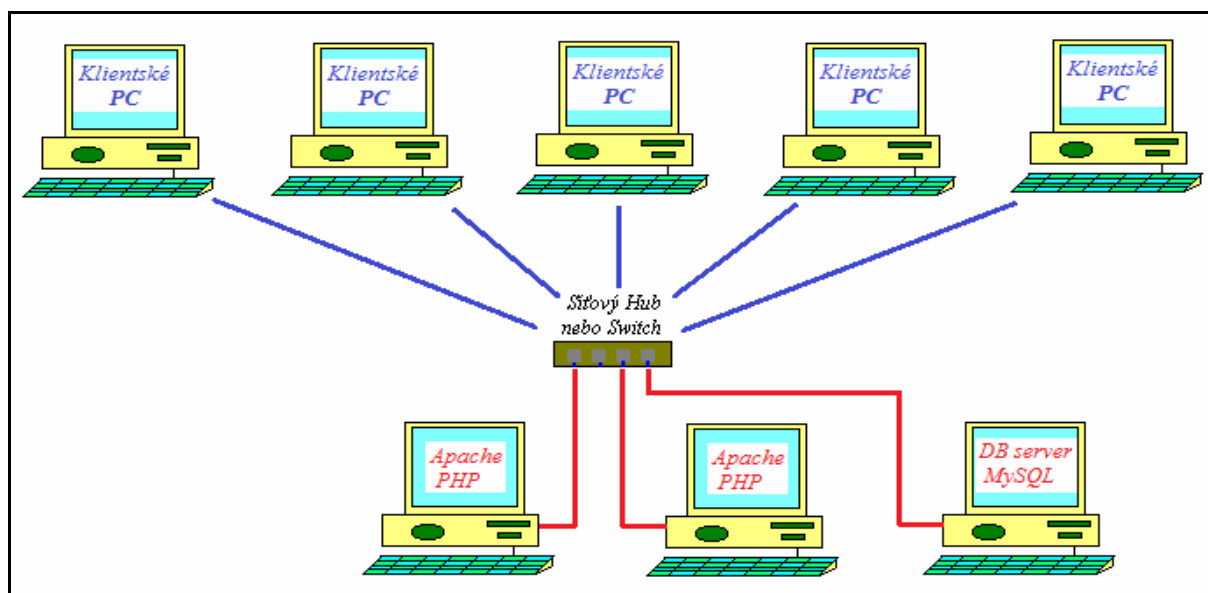
Následující část popisuje metodu, která umožňuje použití více serverů, serverový cluster. Docházka se jako program skládá z několika částí, které na serveru obsahují dvě služby. Jedna služba je webový server Apache s podporou skriptovacího jazyka PHP a druhou službou je databázový server, ve výchozí instalaci s MySQL databáze. Protože při složitějších sestavách leží větší část zátěže na Apache s PHP, je možné systém rozdělit na jeden nebo více aplikačních serverů s Apache + PHP a dále jeden databázový server s MySQL.

Základní rozdělení je tedy jeden aplikační server a jeden databázový server. Viz ukázka:



Aplikačních serverů však může být i více. Za optimální konfiguraci lze považovat dva až tři aplikační servery s Apache + PHP a k nim jeden databázový server s MySQL v případě, kdy jsou všechny servery stejné HW konfigurace. Z měření vyplynulo, že u čtyř aplikačních serverů se již stává databázový server přetížený, pokud jsou všechny stejné HW konfigurace. Naopak v případě, kdy máte jeden velmi výkonný server a dále několik serverů „slabších“, je vhodné použít tento výkonný server právě jako databázový a pak lze aplikačních serverů použít více než 3 bez výkonnostního propadu až do hraničního počtu aplikačních serverů.

Na tomto obrázku je jednoduchý náčrt zapojení v případě použití dvou aplikačních serverů a jednoho serveru databázového.



Shrnutí možných konfigurací:

1. Vše na jednom PC:

(klasická instalace nejedná se o cluster)

Z pohledu náročnosti instalace je samozřejmě nejsnazší klasická instalace, kde je jeden server, který zároveň slouží jako aplikační i databázový. Tato instalace je popsána v dodávaném tištěném návodu a zvládne jí provést i běžný uživatel PC. Pro firmy do 1000 zaměstnanců.

2. Jeden aplikační a jeden databázový server:

(cluster dvou serverů)

Rovněž instalace jednoho aplikačního a jednoho databázového serveru z obrázku na předchozí straně jsou poměrně snadné, protože se jedná v podstatě o spuštění klasické instalace na dvou serverech, následně se na aplikačním serveru zakáže služba MySQL, na databázovém serveru se zakáže služba Apache, na databázovém serveru se povolí připojení ze sítě a na aplikačním serveru se upraví konfigurační soubor *access.php* pro spojení s databází – změní se síťová adresa z lokální na adresu serveru s databází.

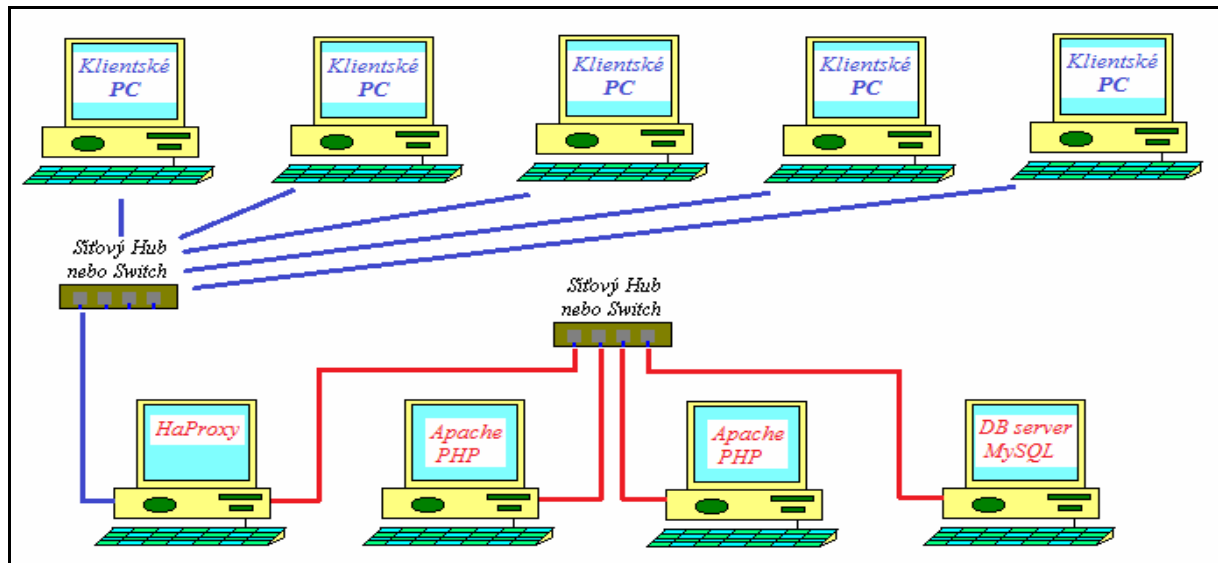
3. Více aplikačních serverů a jeden databázový server:

Komplikovanější je situace při použití více než jednoho aplikačního serveru. Zde je třeba navíc zajistit, aby se stejný uživatel (klientské PC) připojoval stále ke stejnému aplikačnímu serveru, protože jinak by nefungovalo vytváření sestav jako je *Výkaz*, *Výsledovka*, *Přehledka*, exporty do mezd a další části, které si dočasné výpočty ukládají do souborů na disku aplikačního serveru a tvorba sestavy obsahuje více volání http protokolu.

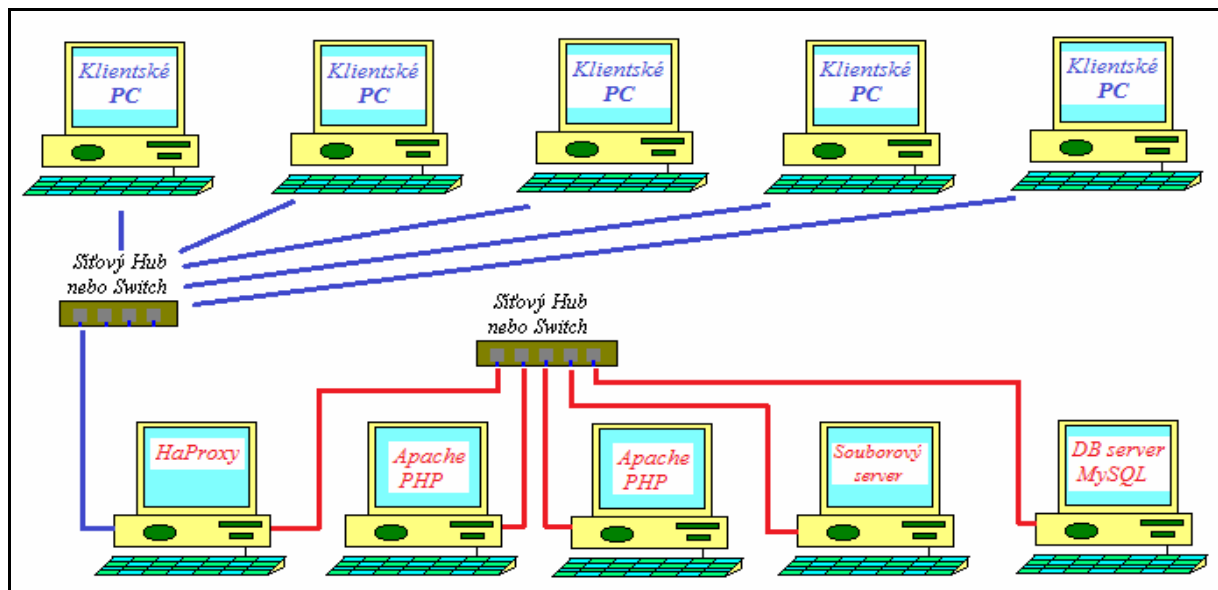
Tento stav lze zajistit minimálně dvěma metodami:

- Buď tak, že jednotliví klienti budou používat stále stejný aplikační server. Například klienti 1 až 100 budou používat vždy aplikační server A a klienti 101 až 200 pak aplikační server B. Jednoduše budou mít v prohlížeči napevno uvedenou IP adresu konkrétního serveru s docházkou. Nevýhodou je nutnost ručně konfigurovat klienty (odhadnout rovnoměrné rozložení klientů mezi aplikační servery)
- Druhou možností je doplnění dalšího počítače (proxy serveru) s nainstalovanou službou HaProxy a tuto službu nakonfigurovanou na režim *source* (místo výchozího *roundrobin*). Zde odpadá ruční konfigurace klientů – všichni budou nastavení na adresu proxy serveru.

Obrázek níže ukazuje velmi zjednodušené schéma zapojení varianty s proxy serverem. Kdy není vyloženě nutné, aby bylo jako na obrázku použito dvou switchů a počítač s proxy serverem musel mít dvě síťové karty. V jednodušším případě lze použít i jeden switch kam budou zapojeny všechny počítače a proxy serveru pak stačí jedna síťová karta.



Dále komunikace s terminály BM-Finger, která rovněž využívá konfigurační soubor na disku aplikačního serveru a při změně konfigurace terminálu (úprava IP adresy, doplnění, editace či smazání terminálu) je nutné ručně soubor *d3kfinger.conf* přenést na všechny aplikační servery, nebo sdílet složku s programovými soubory docházky (*apache\htdocs\dochazka2001*) mezi všemi aplikačními servery, tak aby postupně spouštěná http volání dostala pro služby apache na všech serverech stejné datové soubory na disku. Tuto situaci, tedy sdílení programového adresáře docházky mezi všemi aplikačními servery, lze zajistit úpravou položky `DocumentRoot "C:/apache/htdocs"` v souboru *c:\apache\apache\conf\httpd.conf*, tedy jejím přepsáním cesty na všech aplikačních serverech na sdílený adresář. Sdílení programov. adresáře může zajistit buď jeden ze stávajících serverů, nebo lze použít samostatný souborový server, který bude sdílený pro všechny aplikační servery. Viz ukázka na tomto obrázku:



Použití proxy serveru (load balanceru) a souborového serveru se sdílením programové složky docházky *htdocs* řeší všechny problémy použití více aplikačních serverů. Tedy:

- všechny aplikační servery budou stejně vytížené (při sdílení programového adresáře docházky na souborovém serveru může být HaProxy nakonfigurováno na výchozí režim *roundrobin*)
- správně funguje také komunikace s terminály (sdílí se konfigurační soubor komunikačního modulu v programové složce docházky)
- sestavy jako výsledovka, výkaz, přehledka, export do mezd atd. fungují správně (sdílí se dočasné pracovní soubory s výsledky mezivýpočtů v programové složce docházky)
- klienti mají jednotnou konfiguraci (adresu docházky v prohlížeči mají nastavenou na adresu proxy serveru)
- Operační systém Windows je potřebný pouze na aplikačních serverech. Takže zdatný správce IT může na počítačích klientů provozovat zcela libovolný operační systém (Linux, MacOS, iOS, Android ...) a na proxy serveru, databázovém serveru a souborovém serveru lze použít Linux, čímž se ušetří náklady na licence OS. Stejně tak na samotnou Docházku 3000 stačí jedna licence (pro jednu firmu).

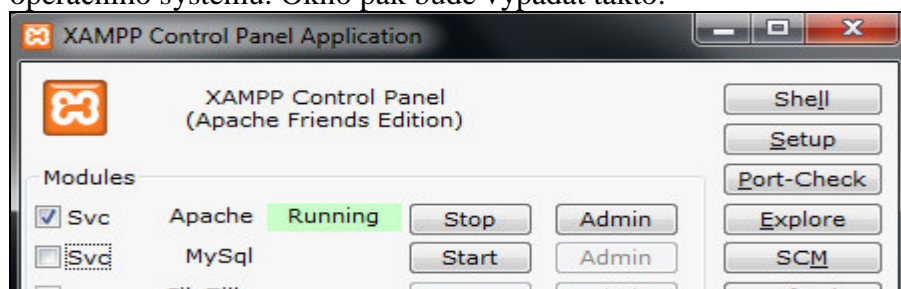
4. Více aplikačních serverů a více databázových serverů:

Díky možnosti nakonfigurovat MySQL službu jako replikační v režimu *master-master* např. pomocí *Galera clusteru* na dva nebo více databázových serverů, případně ještě i ve spojení s *MariaDB MaxScale*, by mělo být možné použití i více databázových serverů. Tuto možnost však výrobce docházkového systém v reálném provozu doposud nenasazoval, takže nejsou k dispozici informace o spolehlivosti multi-master konfigurace databázové služby ve spojení s docházkou. Proto pro nasazení takové konfigurace neposkytuje podrobný návod a tak jí zatím nelze pro ostrý provoz docházkového systému v produkčním prostředí doporučit. Stejně tak pro PostgreSQL nebo MS SQL databázové servery není škálování DB prakticky ověřeno. Doporučujeme tedy spíše HW upgrade DB serveru, pokud jeho výkon přestane stačit. Případně dle příručky *Replikace_db.pdf* rozložit část čtecích operací na server repliky. Postup najdete v PDF návodu od verze programu 9.11 v menu „Firma / Databáze / Replikace DB“.

C) Cluster – jeden aplikační a jeden databázový server – postup

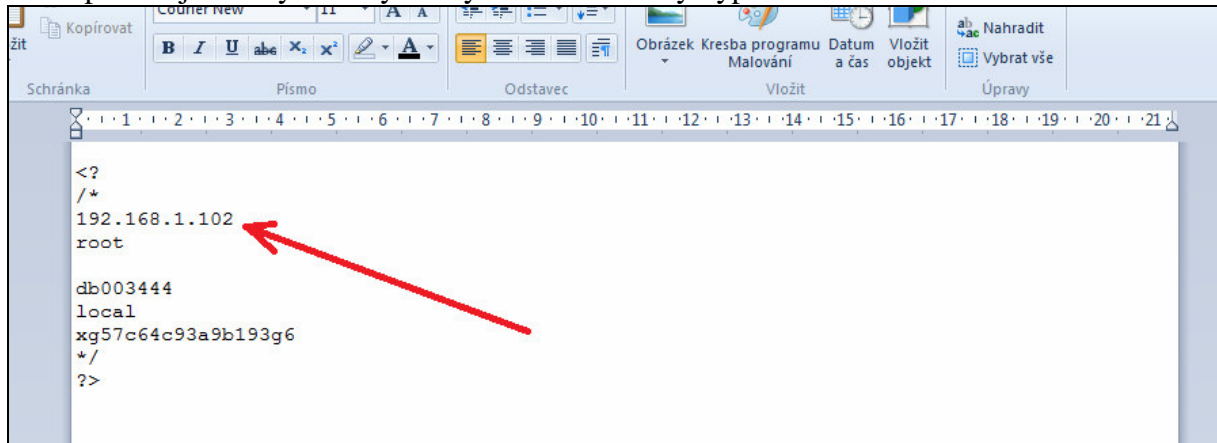
Instalace této varianty odpovídá v předchozí části bodu dva - B2. Stačí vybrat dva servery a na obou provést první bod tištěného návodu – spuštění instalačního programu *instaluj.bat* přímo z CD docházky (ne z kopie, ale přímo z CD vloženého do mechanik serverů). Provedete tedy bod 1 tištěného instalačního návodu na obou serverech a než přistoupíte k bodu 2, provedete před ním ještě těchto 5 kroků:

- 1) Na **aplikačním** serveru zakážete službu MySQL tak, že spustíte program `c:\apache\htdocs\dochazka2001\xampp-control.exe` a v něm kliknete ve druhém řádku pro MySQL na tlačítko *Stop*, potvrdíte zastavení služby, následně ve stejném řádku zrušíte zatržítka *SVC* čímž se MySQL odinstaluje ze služeb operačního systému. Okno pak bude vypadat takto:



Na aplikačním serveru jsme tedy zastavili a zakázali službu MySQL
Program *xampp-control* ukončíte tlačítkem *Exit* vpravo dole.

2) Opět na **aplikačním** serveru upravíte ve složce *c:\apache\htdocs\dochazka2001* pomocí programu *Wordpad* soubor *access.php* tak, že na třetí řádek místo původního slova *localhost* doplníte IP adresu **databázového** serveru. Pokud má tedy aplikační server IP adresu například 192.168.1.101 a databázový server IP adresu 192.168.1.102, doplníte adresu pro databázový server (192.168.1.102) a upravený soubor uložíte. Žádné další změny v souboru neprovádějte ani nepřidávejte řádky atd. Výsledný soubor bude tedy vypadat takto:

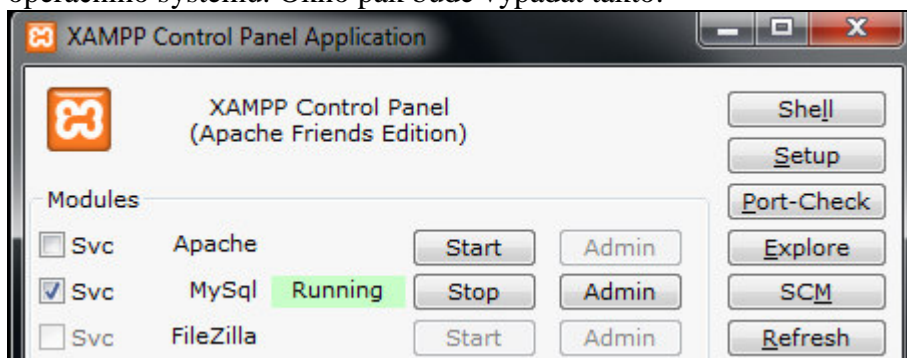


```
<?
/*
192.168.1.102
root

db003444
local
xg57c64c93a9b193g6
*/
?>
```

Poznámka: Pro zkušenější správce doporučujeme na čtvrtý řádek napsat jméno uživatele, pod kterým se bude aplikační server připojovat do databáze na databázovém serveru a na pátý řádek heslo tohoto uživatele. Na databázovém serveru se pak tento uživatel založí s plnými právy k databázi db003444 a povoleným připojením jen z IP adresy aplikačního serveru.

3) Nyní přejdete k **databázovému** serveru, kde jste rovněž provedli podle bodu 1 tištěného návodu standardní instalaci docházky. Poté spustíte program *c:\apache\htdocs\dochazka2001\xampp-control.exe* a v něm kliknete v prvním řádku pro Apache na tlačítko *Stop*, potvrdíte zastavení služby, následně ve stejném řádku zrušíte zatržítka *SVC* čímž se Apache odinstaluje ze služeb operačního systému. Okno pak bude vypadat takto:



Na databázovém serveru jsme tedy zastavili a zakázali službu Apache
Program *xampp-control* ukončíte tlačítkem *Exit* vpravo dole.

4) Opět na **databázovém** serveru spustíme příkazový řádek. V něm zadáte příkaz:

```
c:\apache\mysql\bin\mysql -u root
```

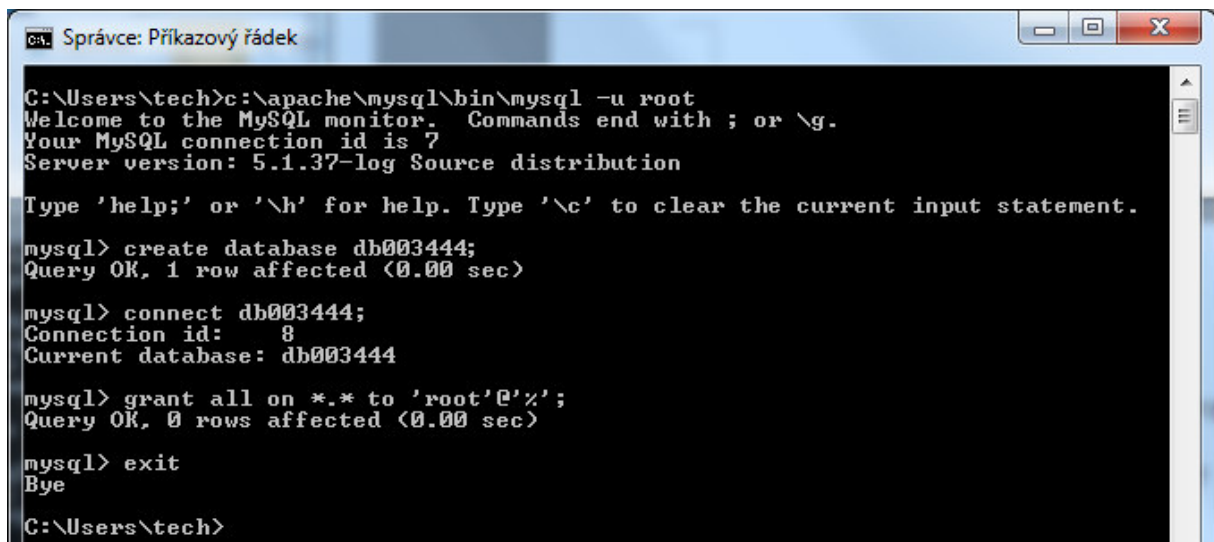
a po spuštění konzoly MySQL pak příkazy

```
create database db003444;
```

```
connect db003444;
```

```
grant all on *.* to 'root'@'%';
```

Vypíše se hlášení „Query OK ...“, viz ukázka na obrázku níže. Nakonec příkazem *exit* a znovu *exit* ukončíte jak konzolu MySQL tak i příkazový řádek.



```
C:\Users\tech>c:\apache\mysql\bin\mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.37-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database db003444;
Query OK, 1 row affected (0.00 sec)

mysql> connect db003444;
Connection id:      8
Current database:  db003444

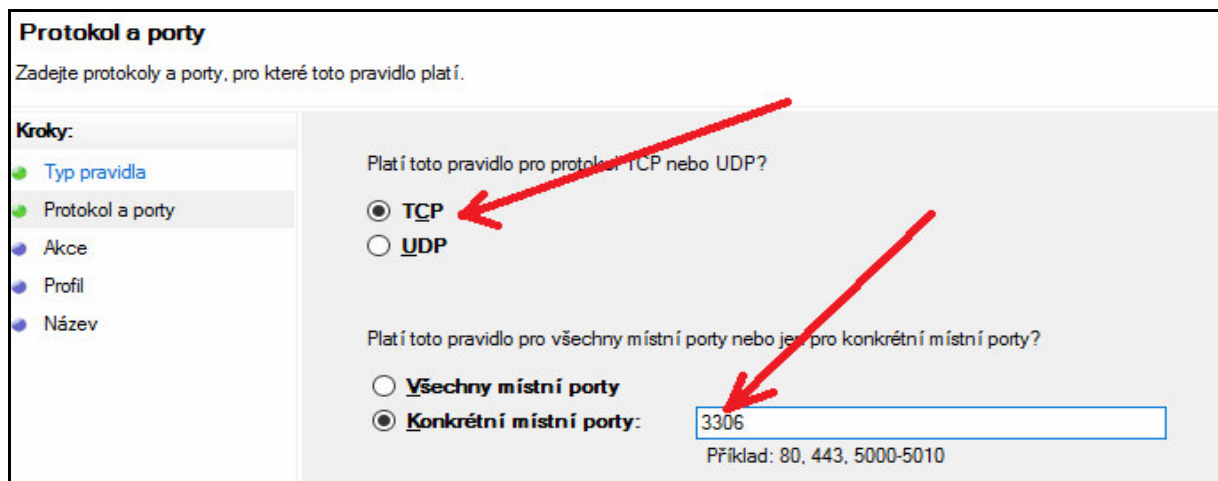
mysql> grant all on *.* to 'root'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

C:\Users\tech>
```

Poznámka: Pro zkušenější správce doporučujeme v MySQL serveru založit speciálního uživatele s přístupem chráněným heslem a do příkazu *grant* uvést jak jméno tohoto uživatele, tak i IP adresu aplikačního serveru. Jméno a heslo se doplní v bodě 2 na 4. a 5. řádek. Přístup do databáze docházky po síti tak bude více chráněn před zneužitím.

5) Ve firewallu databázového serveru povolíte přístup na port 3306, což je port pro síťovou komunikaci s MySQL. Bez tohoto povolení ve firewallu nepustí operační systém aplikační server do databáze. Firewall naleznete v ovládacích panelech pod ikonou *Brána firewall* (zobrazit ovládací panely jako *malé ikony*) / *Upřesnit nastavení* / *Příchozí pravidla* / *Nové pravidlo* / *Port* / *TCP 3306* / *Povolit připojení* / *nechat vše zatrženo* / *název např. MySQL* / *Hotovo*. Poté stejný postup pro UDP protokol.



Nyní na aplikačním serveru dokončíte další body tištěného návodu. Tedy bod 2 a dále. Postup pro přístup do docházky z počítačů klientů je popsán v příručce *pristup_z_jinych_pc.pdf* kterou najdete na CD docházky ve složce *\prirucky*. Odblokování firewallu pro port 80 se provádí na aplikačním serveru a i do prohlížečů klientů nebo odkazu či ikony se vkládá IP adresa aplikačního serveru (v tomto příkladu 192.168.1.101). Klienti sami s databázovým serverem nekomunikují, vždy jen se serverem aplikačním.

Tímto postupem jste tedy docházku nakonfigurovali tak, že aplikační část běží na jiném serveru než databázová a systém zvládne vyšší zatížení, resp. odezva klientům bude v případě mnoha desítek současně pracujících uživatelů rychlejší, než kdyby byl celý systém nainstalovaný na jediném serveru.

D) Cluster – více aplikačních a jeden databázový server – postup

Základní postup je z větší části podobný předchozímu postupu C) pro jeden aplikační a jeden databázový server. Jen aplikačních serverů máte více, takže je také třeba body C1 a C2 provést na všech aplikačních serverech. Body C3, C4 a C5 pak na jednom společném databázovém serveru.

Pokud nechcete instalovat proxy server, musíte rovnoměrně ručně rozdělit adresy aplikačních serverů pro docházku mezi klientské počítače. Nebo minimálně klientům, u kterých se předpokládá vyšší používání webového rozhraní programu. Ruční přidělení IP adres aplikačních serverů klientům řeší problém funkčnosti sestav jako je výsledovka, výkaz, přehledka, exporty do mezd a dalších, pokud nesdílíte programový adresář docházky *htdocs*. U starých terminálů BM-term a BM-net není třeba žádných větších změn v konfiguraci a ovladače mohou běžet na některém aplikačním serveru.

U terminálů BM-Finger je však po nakonfigurování IP adres terminálů třeba zajistit ručně nakopírování souboru *c:\apache\htdocs\dochazka2001\d3kfinger.conf* na všechny aplikační servery. A při každé změně konfigurace v menu „Firma / Terminály BM-Finger“ nezapomenout tento krok provést znovu.

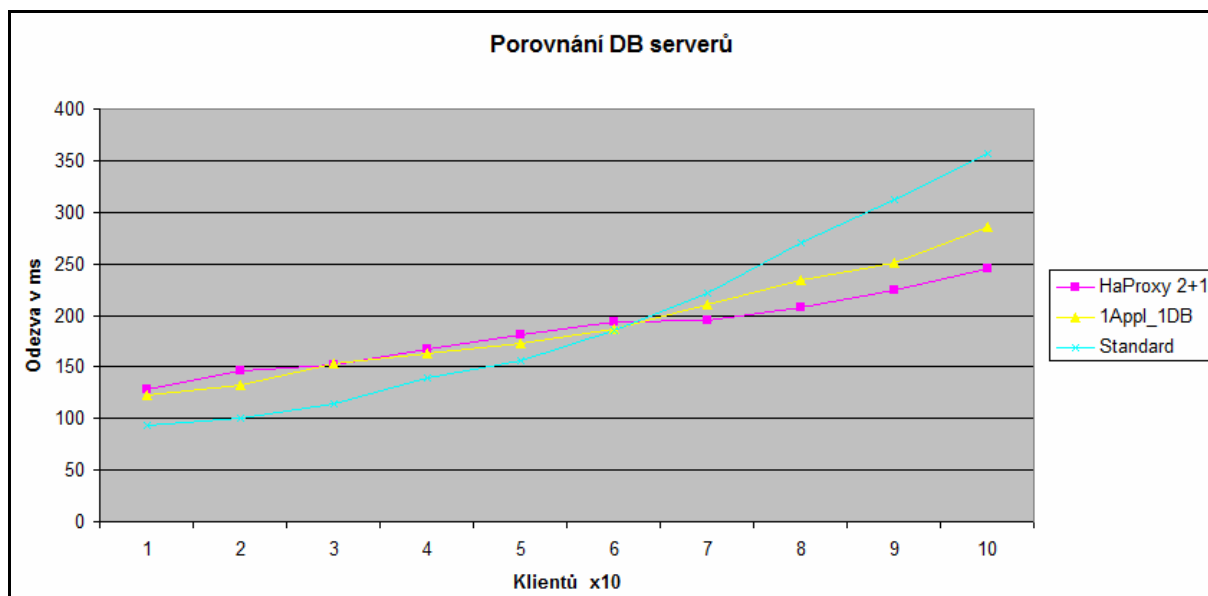
Aby se výše uvedeným problémům předešlo, doporučuje se provést instalaci HaProxy a přesun programových souborů docházky na sdílený disk souborového serveru. Viz informace v bodě B3.

V konfiguraci HaProxy se doporučuje timeouty zvednout na desetinásobek a v konfiguraci mysql zvednout parametr *max_connections* na 1900.

Dále je třeba v konfiguraci HaProxy zapnout parametr *option forwardfor except 127.0.0.1* čímž se zajistí správné logování adres klientů v docházce místo logování adresy proxy serveru (vyžaduje docházku verze 7.72 a vyšší)

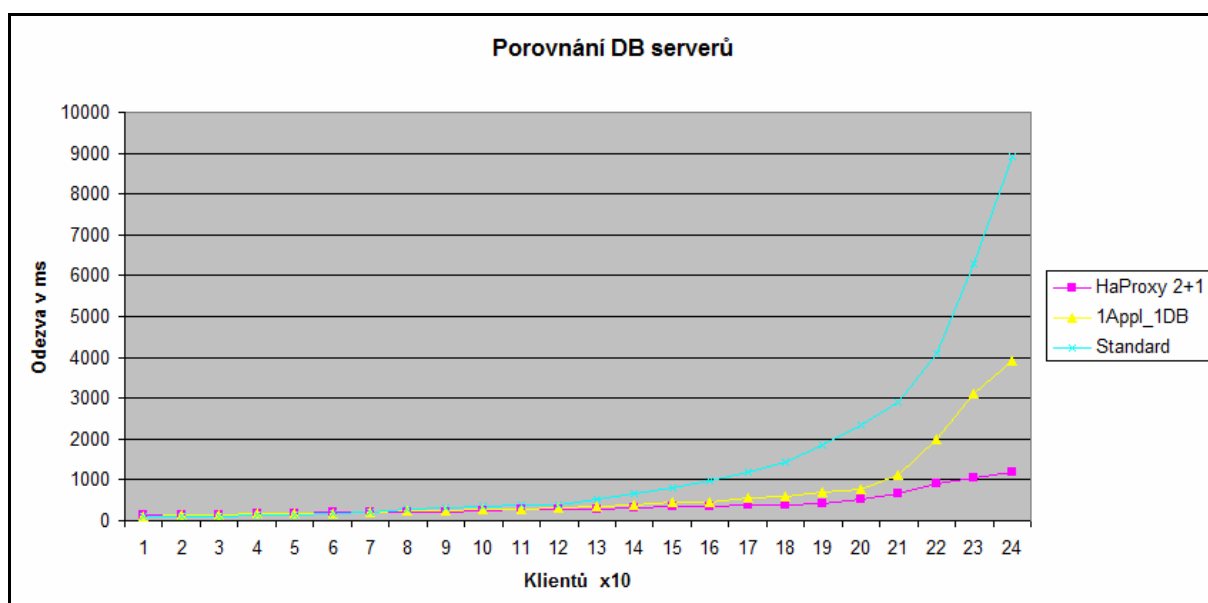
E) Závěr, porovnání, grafy

Pro malý počet klientů se vůbec nedoporučuje rozkládání zátěže na více aplikačních serverů ani oddělovat databázový server. Pozitivní vliv rozložení instalace na samostatné aplikační servery a server databázový se totiž začne projevovat až zhruba od firem s více jak 1000 zaměstnanců, kde se s programem hodně pracuje přes webové rozhraní. U menších firem je díky latencím výsledek rozložení na více serverů spíše horší než u standardní instalace na jediný server. Viz následující graf.



Na grafu je osa X počet klientů $x10$ a osa Y doba odezvy měsíčního výpisu docházky. Pokud s docházkou pracuje najednou maximálně 60 zaměstnanců (všichni najednou spustí zobrazení měsíčního výpisu své docházky), je doba odezvy u výchozí instalace na jediném serveru stejná nebo spíše i lepší než u složitějších instalací s více servery. Proto se zhruba do 1000 zaměstnanců nevyplatí použití více serverů a doporučuje se zcela standardní instalace programu dle dodaného tištěného návodu na jediný server.

Ovšem u velkých firmách je situace zcela opačná a rozložení instalace na dva nebo více serverů se doporučuje. Viz následující graf.



Na grafu je osa X je počet klientů $x10$ a osa Y doba odezvy výpisu docházky. Spojnice (linky):

- standardní instalace docházky (modrá linka)
- oddělení jednoho aplikačního a jednoho databázového serveru (žlutá linka)
- nasazení jednoho databázového serveru pro dva aplikační servery s využitím HaProxy a instalace docházky na sdíleném disku pro programové soubory (fialová linka)

Použití jednoho společného serveru je na tomto testovaném hardware vhodné zhruba do 100 současně pracujících uživatelů. Pak se již začne projevovat výhoda použití dvou a více serverů, kdy od 120 klientů má použití jednoho společného serveru dvakrát horší dobu odezvy než použití dvou a více serverů.

Od 220 klientů je pak vidět výrazná výhoda použití dvou aplikačních serverů namísto jednoho.

U 240 klientů je při použití konfigurace dvou aplikačních serverů, jednoho databázového serveru, proxy serveru a souborového serveru výrazně výhodnější než jednodušší konfigurace se dvěma nebo jen jedním serverem. Nejsložitější konfigurace stále udržela dobu výpisu měsíčního přehledu docházky na úrovni kolem jedné vteřiny, takže reakce je pro uživatele stále velice rychlá i když s docházkou pracuje najednou dalších 240 uživatelů. Kdežto při použití jen jednoho aplikačního serveru s odděleným databázovým serverem již trval výpis kolem 4 vteřin. Zdaleka nejdéle, téměř 10 vteřin již trval výpis při standardní instalaci celého systému na jediném společném serveru.

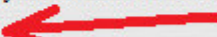
Pokud máte slabší HW konfiguraci dostupných serverů, na kterých potřebujete docházku provozovat a počet zaměstnanců je řádově ve vyšších tisících, lze zvážit přechod na verzi docházky s MS SQL serverem nebo ještě lépe (viz bod 5A) s PostgreSQL serverem, které mají při větší paralelní zátěži lepší dobu odezvy než výchozí instalace s MySQL serverem.

Každopádně při převodu docházky do clusteru dvou a více serverů doporučujeme pečlivě zkontrolovat správnost konfigurace, ověřit korektní vytváření sestav jako je výkaz, výsledovka, přehledka, exporty do mezd atd. a dále zejména správnost přenosu dat z terminálů BM-Finger. Pokud nechcete používat souborový server pro sdílení programového adresáře docházky, musí být proxy server v režimu *source* a pro přenos dat z terminálů BM-Finger lze použít řešení popsané na CD docházky v příručce
`\terminál\doc\bmfinger\autostart_bmfinger.pdf`

Doplnění:

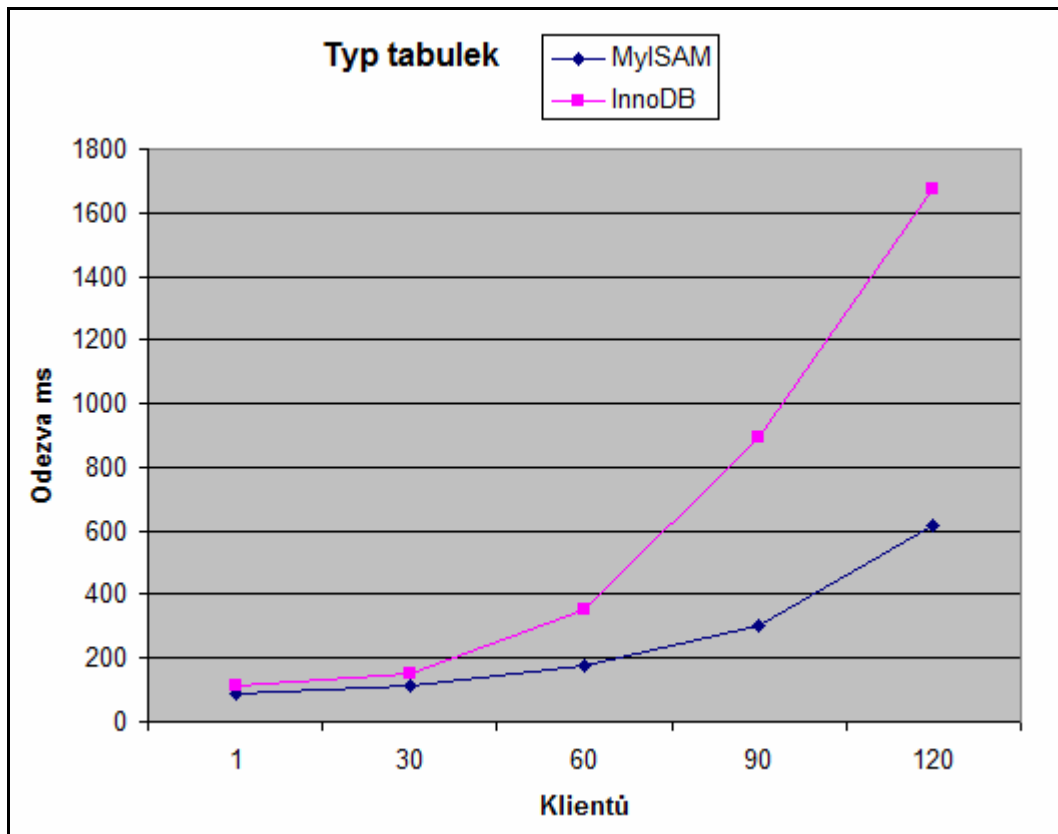
Od verze programu 7.77 umí docházkový systém vyhodnocovat stravné na zahraničních služebních cestách. Stravné je díky legislativě (Zákoníku práce) v tomto případě počítáno komplikovanější metodou, protože při souběhu české a zahraniční služební cesty se počítá speciálním algoritmem. Ten mimo jiné vyžaduje komunikaci se serverem ČNB kvůli dohledání kurzů měn pro zahraniční stravné. Tato komunikace a složitější výpočet způsobí, že doba trvání přípravných sestav jako je výkaz, výsledovka, přehledka, výpis s absencemi, exporty atd., trvá zhruba dvojnásobnou dobu. To, jestli se použije standardní rychlý výpočet, nebo tento složitější předpokládající zahraniční služební cesty, pozná program tak, že se podívá do menu „*Firma / Sazby sl.cest / Přidělení kódů absencí zahraničním služebním cestám*“. Když zde nejsou zadané žádné kódy pro zahraniční cesty (státy), je výpočet rychlý. Ale pokud zadané jsou, musí program použít složitější metodu počítání, která pro každou cestu volá zvláštní algoritmu a může navíc vyžadovat komunikaci s ČNB po internetu, což je časově náročné. Pokud tedy vaši pracovníci nevykonávají zahraniční služební cesty (nezaměstnáváte například řidiče mezinárodní spedice), nezadávejte v tomto menu žádné záznamy.

Přiřazení kódů absencí ke státům:

*Nezadávejte, pokud nemá hlavní PC docházky internetové spojení na server ČNB.
Zatím nejsou zadané žádné kódy absencí ke státům.* 

Verze MySQL databázového serveru:

Docházkový systém se standardně dodává s MySQL ve verzi 5.1.37, která je sice již starší, ale osvědčená pro instalaci s Windows ve verzi pracovní stanice i serveru. Z výkonnostních testů navíc vyplývá výrazně lepší doba odezvy při použití tabulek typu *MyISam* než u tabulek *InnoDB*. Protože docházka nevyžaduje od databázového serveru podporu transakcí, cizích klíčů a další vlastností tabulek *InnoDB*, ale tyto prostředky si zajišťuje sama na úrovni aplikace, postačí jí jednodušší a výkonnější tabulky typu *MyISam*. Jak ukazuje níže uvedený graf, doba odezvy je lepší (kratší / nižší hodnota) pro tabulky *MyISAM* a rozdíl se s počtem klientů ještě dále prohlubuje. Proto výchozí instalace docházky používá tabulky *MyISam*.

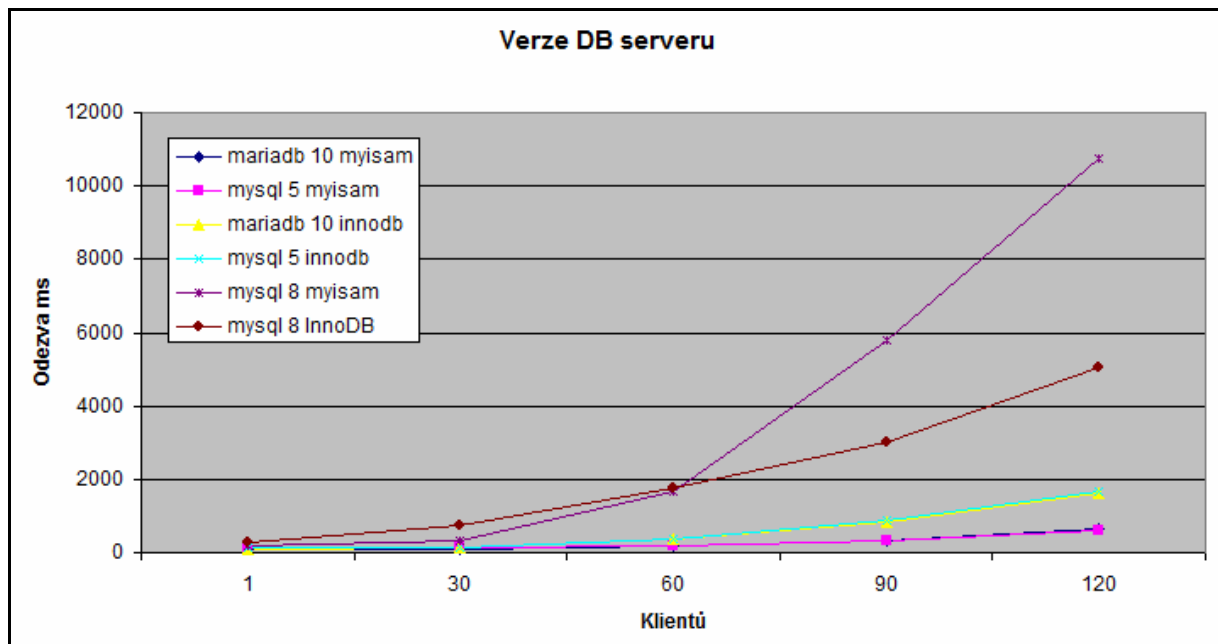


Testováno bylo na databázi s 1000 zaměstnanců a daty docházky za 1 rok.

Dále byl testován výkonnostní průběh tří různých verzí MySQL databázového serveru. A to výchozí 5 (konkrétně 5.1.37), dále v době testů nejnovější MySQL verze 8 (konkrétně 8.0.13) a volně dostupná komunitní MariaDB verze 10 (konkrétně 10.3.12). Testy byly opět prováděny na roční databázi docházky 1000 zaměstnanců.

Jak se při testování ukázalo, zrušení podpory vyrovnávací paměti na straně databázového serveru (query cache) u MySQL 8 je pro docházku velkým handicapem a to dokonce takovým, že jednodušší tabulky typu *MyISam* jsou dokonce pomalejší než *InnoDB*.

Na druhou stranu volně dostupná databáze MariaDB 10 (odnož původní MySQL) je stejně rychlá jako v docházce používaná MySQL verze 5. Pokud tedy chcete používat novější verzi databáze a nevádí vám ztráta podpory ze strany výrobce (ta je dostupná je pro výchozí instalaci s MySQL 5), je možné přejít na MariaDB 10 při zachování výkonu.



Jak je z grafu patrné, odezva MySQL 5 a MariaDB 10 je při použití MyIsam tabulek lepší než při použití InnoDB tabulek, což se dalo předpokládat i z předchozího grafu a obě databáze jsou zhruba stejně výkonné. Naopak použití MySQL verze 8 se pro docházku zatím určitě nedoporučuje.

Verze a licence operačního systému Windows:

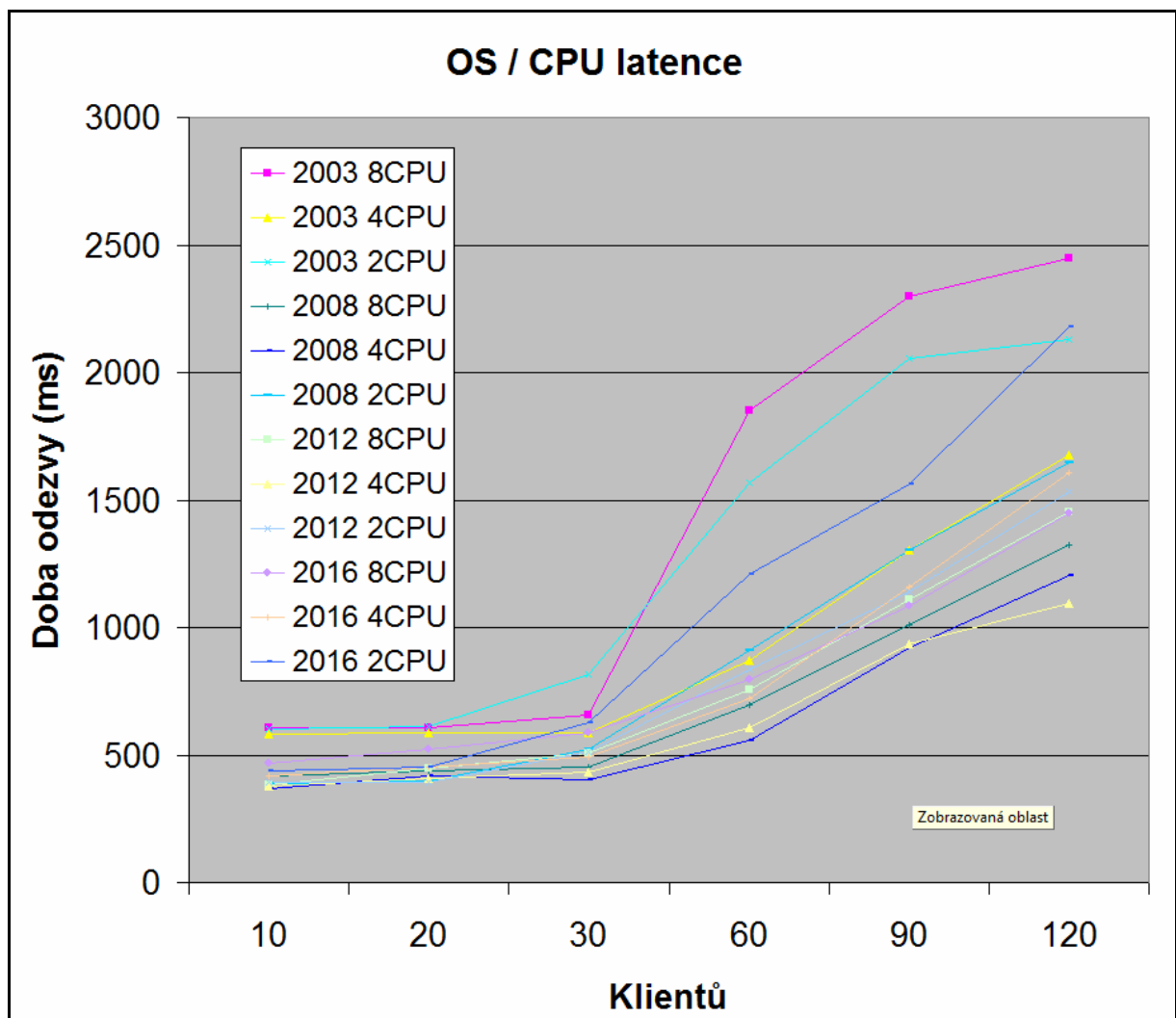
Docházkový systém *Docházka 3000* pracuje se všemi verzemi Windows a to jak ve 32 bitové tak v 64 bitové verzi, počínaje Windows 2000 až po nejnovější Windows 10 nebo 2019 server. Pokud budete docházku provozovat jen na jediném počítači a nebude tedy s programem pracovat současně více uživatelů, stačí vám běžná licence Windows pro pracovní stanice (včetně verze Home). Ovšem v případě, kdy bude po síti s programem pracovat více uživatelů, tak přestože vše funguje správně i při instalaci docházky na obyčejné verzi operačního systému pro pracovní stanice (tedy i třeba na notebook), je třeba kvůli licenci OS použít serverovou verzi operačního systému. Například Windows 2003 server nebo vyšší. Serverové verze Windows jsou navíc i pro současnou práci více klientů optimalizovány a poskytují tak lepší výkon s kratší dobou odezvy, tedy plynulejší práci s docházkou v případě současného připojení mnoha uživatelů.

Byla provedena měření odezvy docházkového systému na různých serverových verzích operačního systému Windows v kombinaci s různým počtem CPU jader procesoru. Test byl prováděn na serveru HP Proliant DL160 G6 s virtualizačním systémem Proxmox, pod kterým byly postupně provedeny instalace virtuálního PC s 24GB RAM, 2 až 8 jádry CPU a verzemi OS Windows Server 2003, 2008 R2, 2012 R2 a 2016. Server měl 2 čtyřjádrové procesory Xeon X5560 (podpora HT byla vypnuta). Bylo provedeno měření doby měsíčního výpisu docházky ve firmě s 1000 zaměstnanci a generovanými daty docházky za 1 rok pro různé počty současně s docházkou pracujících uživatelů – od 10 až do 120 klientů. Pro simulaci zátěže byl využit modul docházky v menu „Firma / Databáze / Test databáze / Test škálování“ spuštěný ze dvou klientských PC v konfiguraci testu 2 PC, 6 klientů na PC, 10 měření v každém kroku.

Naměřené hodnoty v níže uvedené tabulce udávají časy v milisekundách pro případ, kdy si určitý počet uživatelů (postupně 10 až 120) ve stejný okamžik zobrazí měsíční výpis docházky.

Počet jader CPU a verze serveru	Počet současně pracujících uživatelů					
	10	20	30	60	90	120
2003 8CPU	606	606	658	1853	2300	2450
2003 4CPU	584	589	586	869	1304	1675
2003 2CPU	603	613	815	1565	2055	2128
2008 8CPU	416	439	455	698	1012	1324
2008 4CPU	368	416	404	559	921	1202
2008 2CPU	386	397	520	912	1304	1646
2012 8CPU	384	448	505	756	1108	1455
2012 4CPU	378	409	432	607	933	1096
2012 2CPU	391	391	513	836	1138	1534
2016 8CPU	469	523	587	797	1087	1450
2016 4CPU	419	454	492	721	1158	1607
2016 2CPU	439	451	629	1207	1561	2181

Přenesení výsledků do grafu:

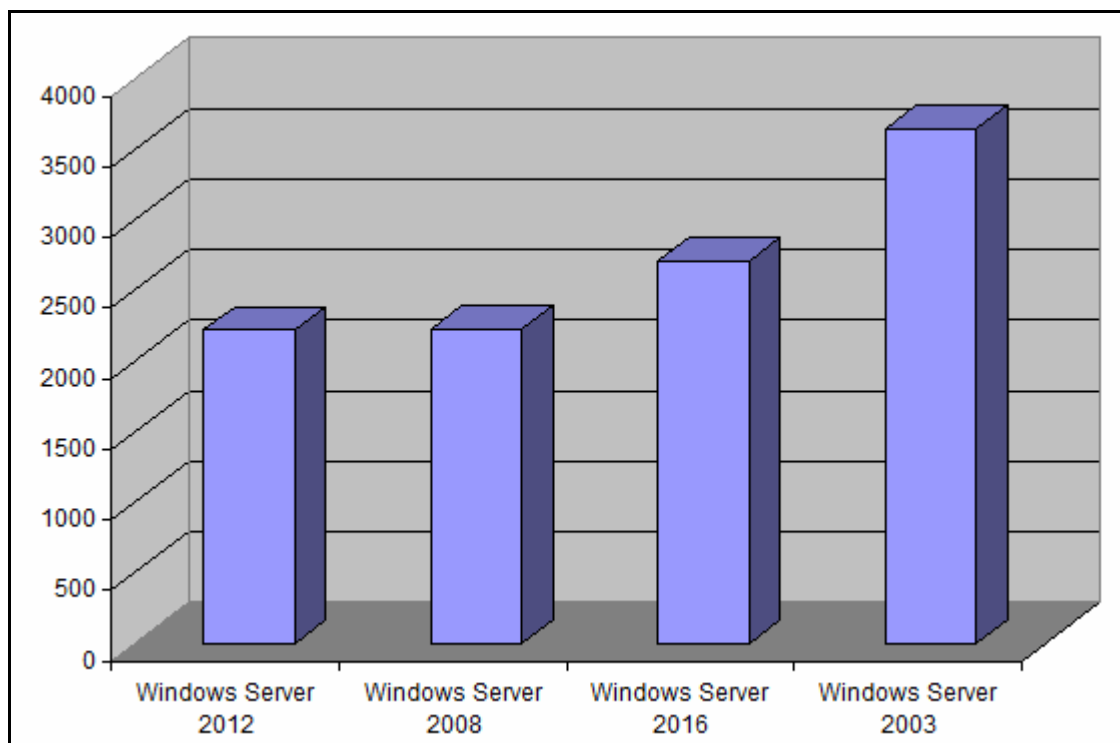


Z výsledků bylo sestaveno toto pořadí vhodné konfigurace *Verze OS / Počet CPU*:

1. Windows Server 2012 se 4 CPU jádry
2. Windows Server 2008 se 4 CPU jádry
3. Windows Server 2008 s 8 CPU jádry
4. Windows Server 2012 s 8 CPU jádry
5. Windows Server 2012 s 2 CPU jádry
6. Windows Server 2016 se 4 CPU jádry
7. Windows Server 2016 se 8 CPU jádry
8. Windows Server 2008 se 2 CPU jádry
9. Windows Server 2003 se 4 CPU jádry
10. Windows Server 2016 se 2 CPU jádry
11. Windows Server 2003 se 2 CPU jádry
12. Windows Server 2003 s 8 CPU jádry

Přitom rozdíl průměru časů mezi prvními dvěma konfiguracemi (server 2012 a 2008, oba se 4 CPU jádry), je velice nepatrný, pouze 0,3% a dá se tedy říct, že se o první příčku dělí. Odstup konfigurace na 3. místě pak je již větší, konkrétně 12,2%

Pokud se zprůměrnují časy odezvy v ms za jednotlivé verze OS Windows Server pro všechny kombinace počtu klientů a CPU jader, vychází pořadí v následujícím grafu:

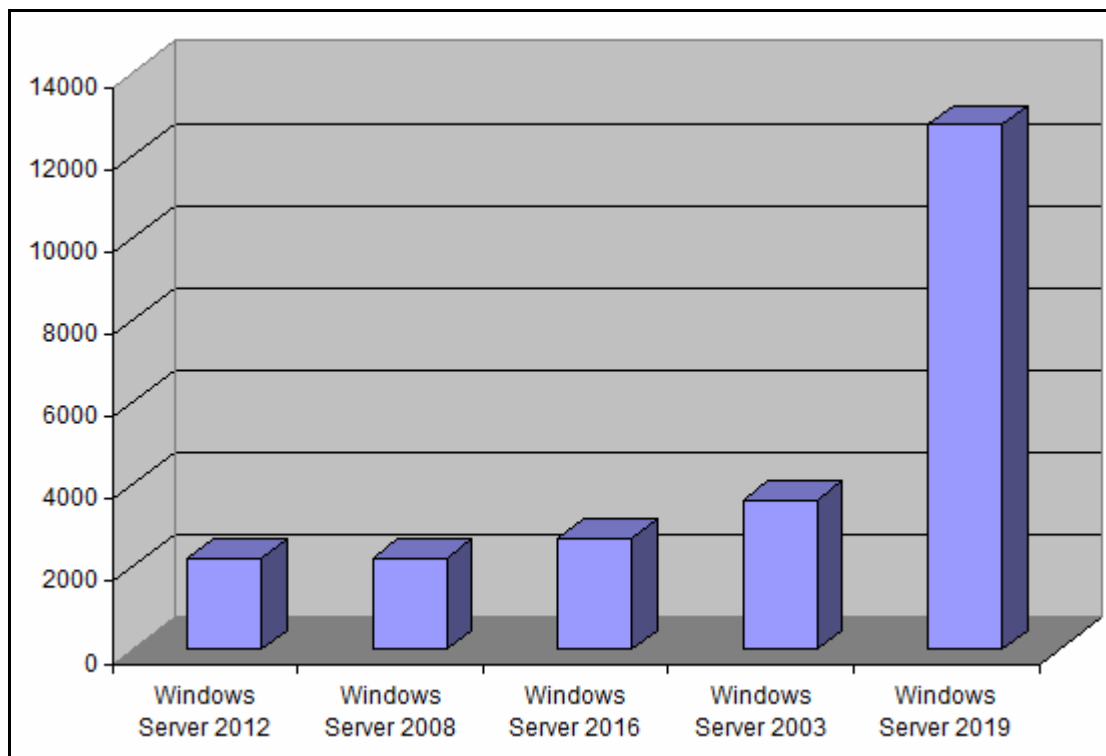


Naměřené výsledky ale samozřejmě platí jen pro uvedenou konfiguraci hostitelského HW a virtualizační technologie. Jiné systémy mohou při různých optimalizacích, kombinaci konfigurací HW/SW a typu zátěže podávat zcela jiné výsledky. Námi testované byly výchozí instalace OS Windows server s grafickým prostředím a jediné změny konfigurace byly povolení portu 80 ve firewallu a nastavení konfigurace sítě. Tedy to, aby klientská PC mohla po síti spustit testy na serveru.

Pokud tedy hledáte nejvhodnější verzi OS pro váš konkrétní server, je možné provést podobná měření s pomocí modulu docházky v menu „Firma / Databáze / Test databáze / Test škálování“.

Doplnění pro Windows Server 2019:

Na stejné HW konfiguraci, stejné verzi SW Docházka 3000 a se stejnými daty zaměstnanců jako výše uvedené testy byl proveden i test pro Windows Server 2019. Stejně jako u předchozích testů OS Windows Server byla použita zcela výchozí konfigurace OS bez úprav nastavení. Výsledky jsou však výrazně horší než u testů předešlých.



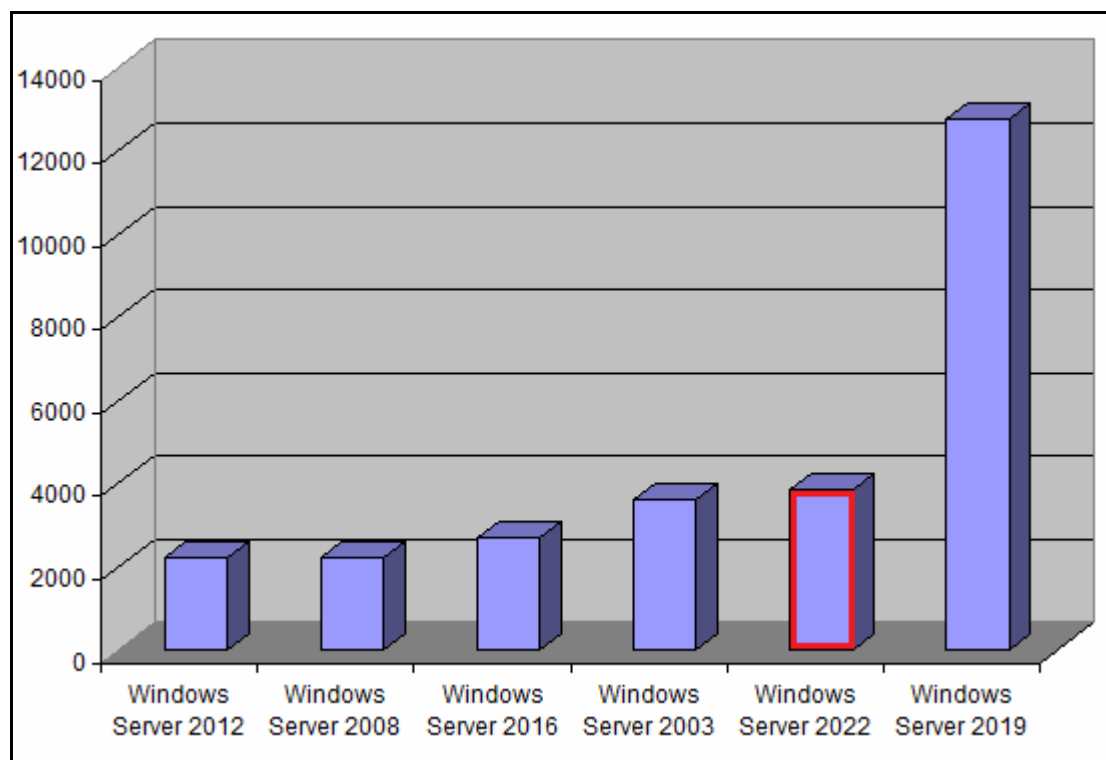
Částečné zlepšení by přineslo omezení služby *Antimalware service executable*, která v průběhu testu zabírala zhruba polovinu výkonu CPU neustálou kontrolou služby MySQL a PHP interpretu, které se při testu používají. Ovšem při reálném nasazení serveru ve firemní síti není samozřejmě vhodné zabezpečení vypínat.

Počet jader CPU a verze serveru	Počet současně pracujících uživatelů					
	10	20	30	60	90	120
2019 16CPU	644	767	926	2188	4996	6242
2019 4CPU	2373	3273	4019	5715	6387	7737
2019 2CPU	2409	3233	4033	5432	7270	8984

Při požadavku na vysoký výkon docházky tedy zatím OS Windows Server 2019 nevykazuje dostatečně krátkou dobu odezvy v porovnání s ostatními staršími verzemi Windows Serveru. Navíc z testu byla vypuštěna měření pro 8 jader CPU, protože výsledky byly ještě výrazně horší než u 4 a 2 jader CPU. Proto bylo použito náhradních výsledků pro 16 jader CPU.

Doplnění pro Windows Server 2022:

V roce 2022 byl na stejné HW konfiguraci, verzi 8.92 programu Docházka 3000 a se stejnými daty zaměstnanců jako výše uvedené, proveden i test pro Windows Server 2022. Stejně jako u předchozích testů Windows Server byla použita zcela výchozí konfigurace OS bez úprav nastavení.



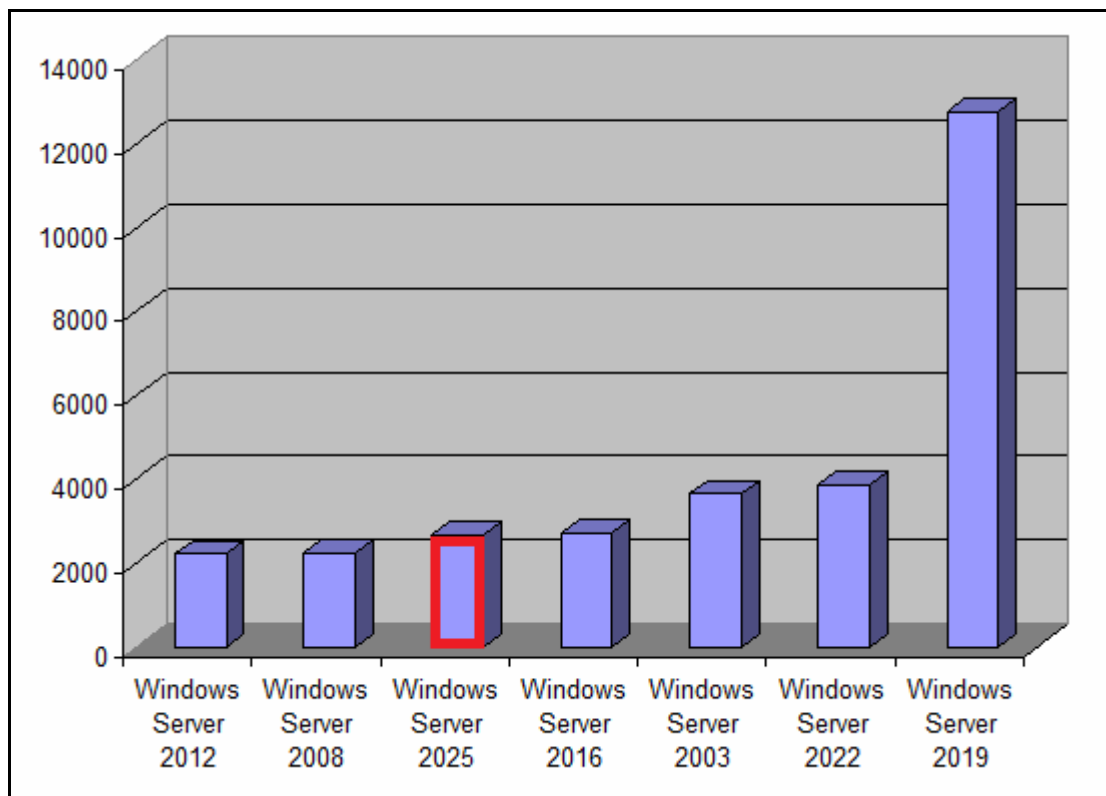
Výsledky jsou již výrazně lepší než na Windows Server 2019 a blíží se Windows Serveru 2003. Na WS 2008, 2012 a 2016 je výkon sice stále lepší, na rozdíl od nich docházka na WS2022 při velkém počtu současně pracujících klientů pořád zhruba o 60% pomalejší, ale již je výrazně lepší než byla na WS2019.

Počet jader CPU a verze serveru	Počet současně pracujících uživatelů					
	10	20	30	60	90	120
2022 4CPU	803	682	692	961	1241	1498
2022 2CPU	837	748	971	1567	2192	2738
2022 8CPU	765	667	766	1508	4601	4732

Při požadavku na vysoký výkon již tedy lze pro provoz Docházky 3000 operační systém Windows Server 2022 doporučit oproti verzi 2016 a starší, a to ne přímo z hlediska lepšího výkonu, ale protože nová verze OS bude mít od výrobce tohoto serverového systému zajištěnou podporu na delší dobu než u šest a více let staré verze OS serveru.

Doplnění pro Windows Server 2025:

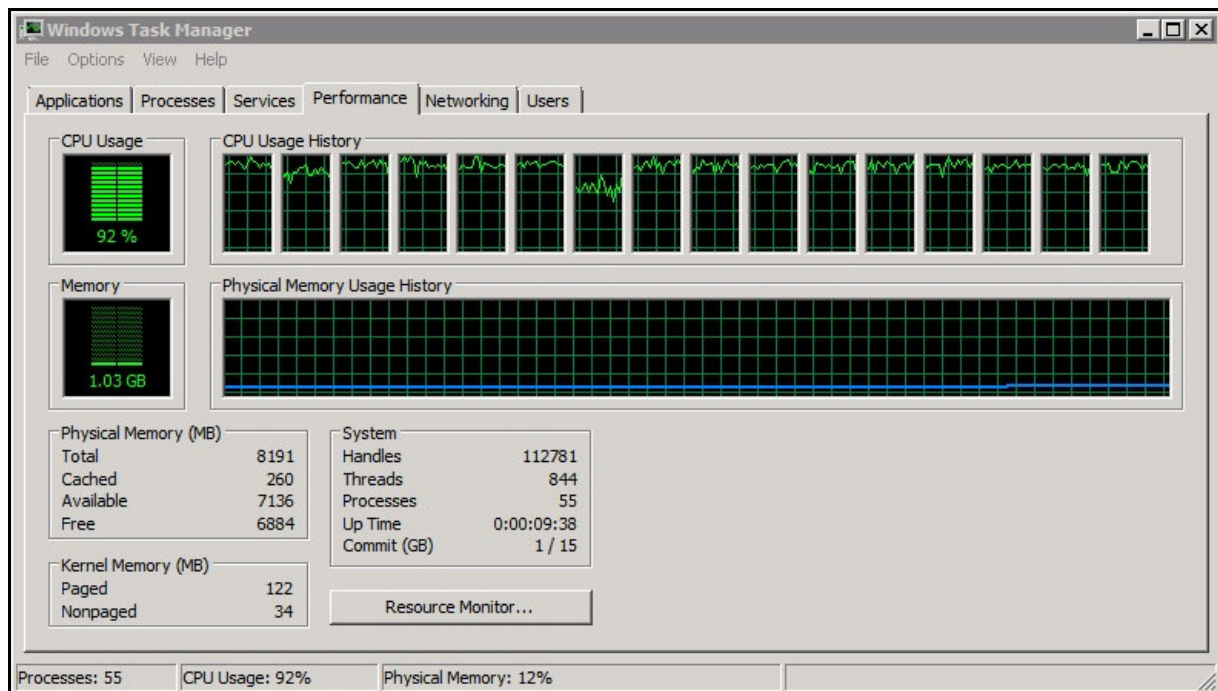
V roce 2025 byl na stejné HW konfiguraci, tedy HP Proliant DL160 G6 se 2 čtyřjádrovými procesory Xeon X5560, ověřen výkon pro virtualizovaný Windows Server 2025. Aktuální virtualizační systémem Proxmox 8.2 a v něm VM s WS2025 s konfigurací 2x4 CPU (8 jader), 16GB RAM, podpora virtIO a disková cache nastavena na WriteBack. Samotný WS2025 měl v konfiguraci napájení nastavenou optimalizaci na maximální výkon. Program Docházka 3000 ve verzi 9.82 s daty 1000 zaměstnanců za 12 měsíců.



Výsledky jsou opět o něco lepší než předchozí Windows Server 2022 a blíží se Windows Serveru 2008 a 2012. Může to být kombinací více faktorů, jako je lepší optimalizace v novějším systému Proxmox 8.2, instalací podpory virtIO, ale i lepší podporou WS2025 ve verzi docházky 9.82.

Počet jader CPU a verze serveru	Počet současně pracujících uživatelů					
	10	20	30	60	90	120
2025 2CPU	534	576	708	1183	1658	2185
2025 4CPU	581	527	535	637	1139	1382
2025 8CPU	548	594	648	677	838	1016

Při požadavku na vysoký výkon lze pro provoz Docházky 3000 operační systém Windows Server 2025 určitě doporučit. Ne nutně oproti verzi 2022, ale zejména proti verzi 2019. Navíc nová verze OS bude mít od výrobce tohoto serverového systému zajištěnou podporu na delší dobu než u šest a více let staré verze OS serveru.



Server 2008R2 s 16 CPU – správce úloh, vytížení CPU při testech

Použití PHP cache:

U jiných intranetových aplikací založených na PHP se můžete na internetu dočíst o možnosti urychlení při opakovaném vykonávání stejných skriptů tím, že do interpretu PHP doplníte vyrovnávací paměť, která si uchová již jednou načtený a zkompilovaný PHP skript a tak je jeho další spuštění rychlejší. Může se přeskočit fáze parsování a kompilace skriptu.

Docházka 3000 však používá již předem přeložené skripty díky použití knihovny `php_bcompiler`, takže parsování a kompilace byla provedena u výrobce a tak váš server již při používání programových php skriptů docházky nemusí tyto operace provádět. Proto použití cache na straně PHP nepřináší u docházky žádné další zrychlení.

Pokud byste však na svém serveru chtěli používat další webové intranetové aplikace, které nejsou předem přeloženy do bytekódu, chtěli přitom využít prostředí již dodávané s docházkou (apache, php, mysql) a pro tyto další aplikace použití kešování uvítali, je od verze programu 7.86 v instalačním souboru připravena knihovna APC (Advanced PHP chache). Stačí jí aktivovat tak, že v souboru `c:\apache\php\php.ini` v sekci „*Dynamic Extensions*“ smažete úvodní středník u řádku `;extension=php_apc.dll` a soubor uložíte.

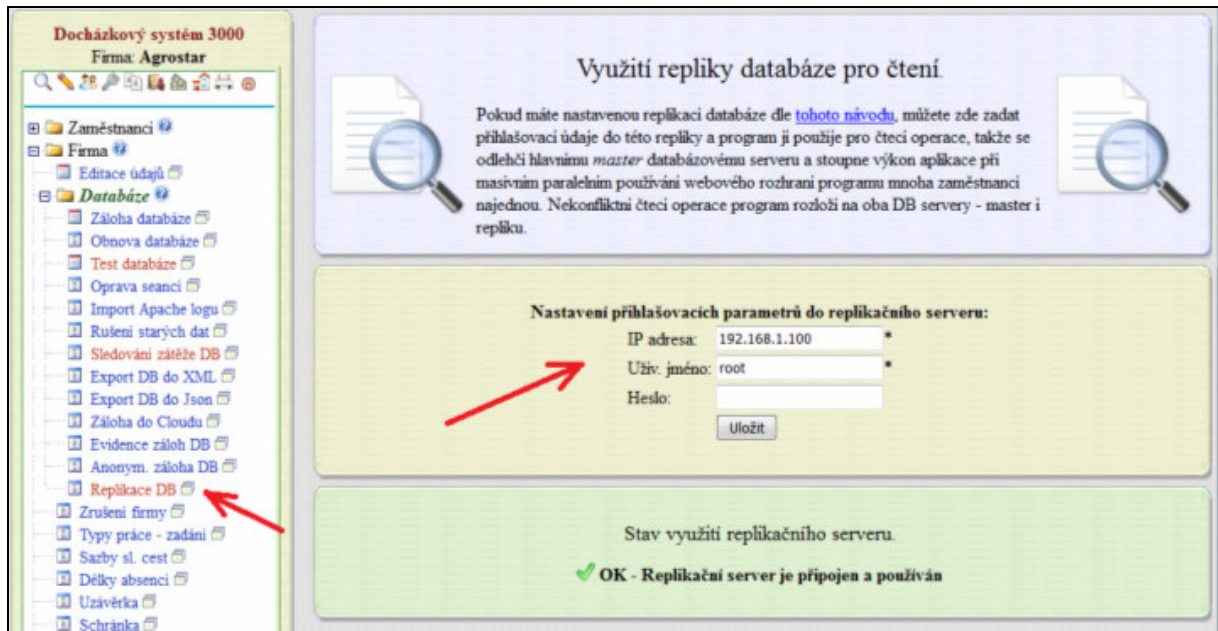
Pro samotnou Docházku 3000 však nemá použití cache na straně php žádný význam, spíše naopak, takže aktivaci kešování php nedoporučujeme. Na rozdíl od *query cache* na straně MySQL, která přínosná je a její aktivace byla popsána v úvodu návodu v bodě A2.

Navýšení rychlosti PHP na dvojnásobek lze dosáhnout změnou konfigurace, kdy PHP nebude z apache spouštěno jako externí aplikace přes `cgi` rozhraní, ale bude nahráno jako modul web serveru. Pro přechod na tuto konfiguraci bude připravena samostatná příručka.

Využití repliky databáze pro rozložení části výkonu na dva DB servery:

Od verze programu 9.11 je v menu „Firma / Databáze / Replikace DB“ odkaz na návod popisující zprovoznění on-line replikace databáze na druhý DB server a v tomto návodu je popsána i možnost využití replikačního serveru pro odlehčení zátěže hlavnímu (master) DB serveru tím, že program část nekonfliktních čtecích operací nasměruje na databázi repliky. Což program Docházka 3000 podporuje od verze 9.11 a připojení na replikační DB server se nastavuje právě v menu „Firma / Databáze / Replikace DB“.

Tato replika pak obsluhuje asi 35% čtecích operací a tím se hlavnímu DB serveru výrazně uleví. Stačí v programu zadat IP adresu a přihlašovací údaje do replikačního serveru:



The screenshot shows the software interface for 'Docházkový systém 3000' by 'Firma: Agrostar'. On the left is a navigation tree with 'Replikace DB' highlighted by a red arrow. The main window is titled 'Využití repliky databáze pro čtení' and contains the following text: 'Pokud máte nastavenou replikaci databáze dle tohoto návodu, můžete zde zadat přihlašovací údaje do této repliky a program ji použije pro čtecí operace, takže se odlehčí hlavnímu master databázovému serveru a stoupne výkon aplikace při masivním paralelním používání webového rozhraní programu mnoha zaměstnanci najednou. Nekonfliktní čtecí operace program rozloží na oba DB servery - master i repliku.'

Below the text is a form titled 'Nastavení přihlašovacích parametrů do replikačního serveru:' with the following fields: 'IP adresa: 192.168.1.100', 'Uživ. jméno: root', and 'Heslo:'. A red arrow points to the 'Uživ. jméno' field. There is an 'Uložit' button. At the bottom, a green box indicates 'Stav využití replikačního serveru. OK - Replikační server je připojen a používán'.

Poté lze i sledovat zatížení obou DB serverů v menu „Firma / Databáze / Sledování zátěže DB“ kde při zapnuté výše uvedené funkci přibude i druhý graf pro sledování zátěže repliky.

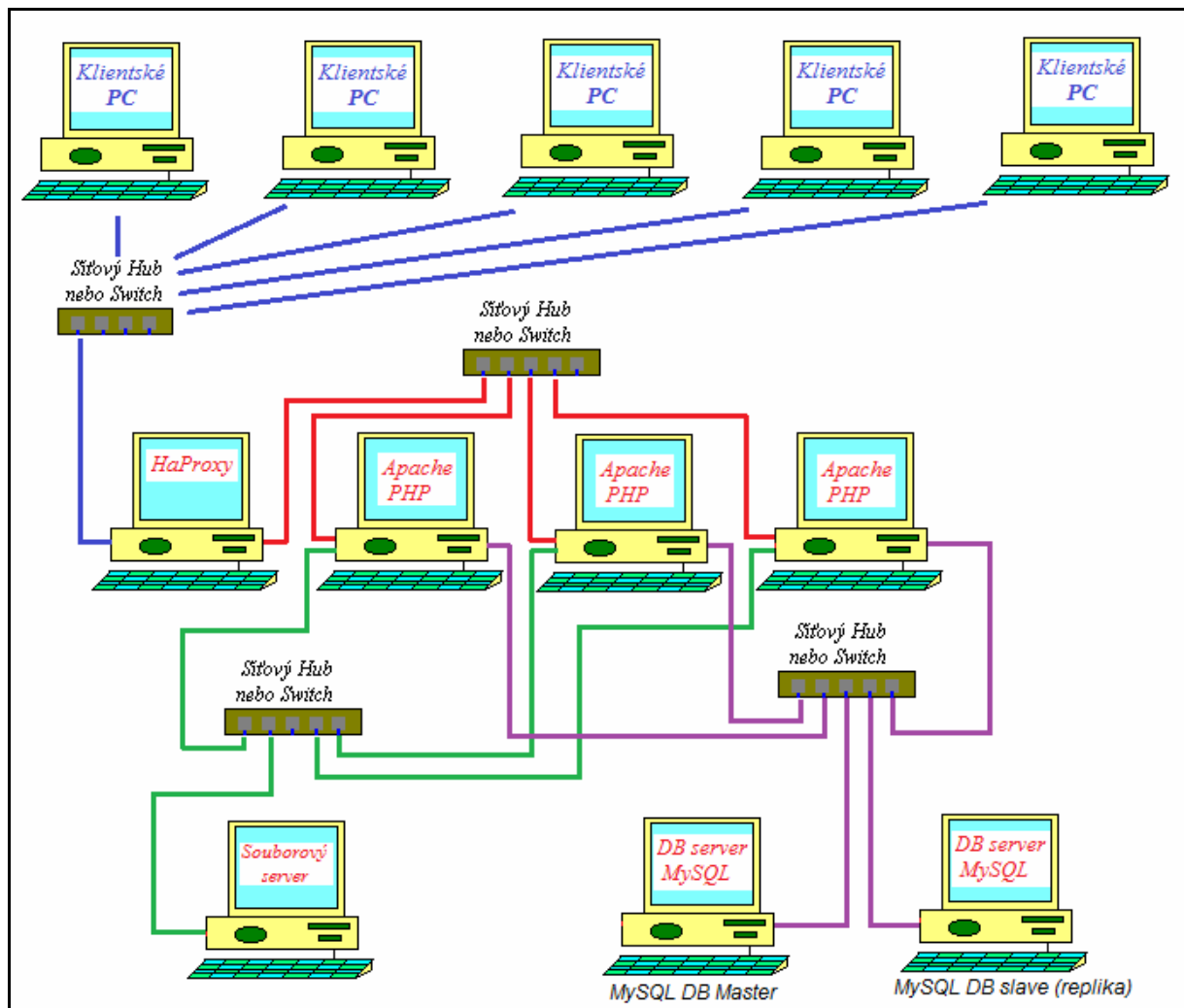


Další tipy:

V konfiguraci *mysql* (soubor *my.ini*) nastavit parametr *max_connections* na vyšší hodnotu. Výchozí 151 není pro mnoho současně připojených klientů dostatečné a doporučuje se dvou až tří násobek předpokládaného počtu současně pracujících klientů (maximální hodnota je 1900).

Oddělená síť mezi aplikačním a databázovým serverem - aplikační server s apache by měl mít dvě síťová rozhraní (dvě LAN karty). Do hlavního switchu se připojí jednou kartou a ta bude sloužit pro komunikaci s počítači uživatelů. Druhá karta bude sloužit čistě pro připojení databázového serveru.

Stejně tak pokud je použitý souborový server pro sdílení programového adresáře docházky mezi více aplikačními servery, měla by být použita samostatná síť mezi aplikačním a souborovým serverem. Což je vidět na následujícím obrázku, kde mají aplikační servery s Apache dokonce tři síťové karty.



Tato situace se ale v praxi příliš nepoužívá a vzhledem k tomu, že datové toky vyžadované docházkou pro souborový a databázový server nejsou nijak veliké, lze v praxi pro aplikační servery použít jedinou kartu a tuto zapojit do společného switchu spolu se souborovým, databázovým a proxy serverem (jak je znázorněno na posledním nákresu v bodě B3).

Virtualizované PC docházky a problémy s výkonem

Pokud používáte virtualizovaný operační systém pro provoz docházky a na hostiteli běží více virtuálů, může být nižší výkon docházky způsobený zátěží jiného virtuálního počítače běžícího pod stejným hostitelem.

Dále také v případě problému s výkonem virtuálního PC docházky věnujte pozornost takzvanému over-provisioningu. Tato technika umožní přidělit virtuálním počítačům v celkovém součtu více zdrojů, než kterými hostitelské PC ve skutečnosti disponuje.

CPU Over-Provisioning

Pokud máte například dvou procesorový server a každý procesor má čtyři jádra, tak (bez hyperthreadingu) hostitelské PC celkem disponuje 8 procesorovými jádry (8 pCPU - fyzických jader). Vy ale můžete na takovém počítači klidně spustit třeba tři virtuální počítače a každému přidělit 4 virtuální procesory (vCPU). Celkem tedy bude 8 fyzických jader simulovat práci 12 virtuálních CPU, což v případě zatížení všech virtuálů a všech jejich CPU způsobí velký propad výkonu oproti situaci, kdyby byly fyzické a virtuální procesory přiděleny v celkovém součtu v poměru 1:1.

RAM Over-Provisioning

Stejně tak paměť RAM přidělená jednotlivým virtuálním PC může přesáhnout celkovou paměť hostitele. Docházka sice není paměťově nijak náročná, ale pokud ostatní virtuály či aplikace spotřebují veškerou paměť hostitele, bude v krajním případě docházet k takzvanému *swapování* (odkládání paměti na disk), což má ještě výraznější vliv na propad výkonu. Menší propad výkonu může u serverů s více fyzickými procesory (sokety) znamenat i špatné přidělení paměti u jednotlivých numa nodů. Každý fyzický procesor (celý procesor, nikoli jádro) má připojenou vlastní RAM. Pokud virtuální počítač poběží na jednom procesoru, ale část paměti bude muset využívat od jiného procesoru, bude přístup do paměti připojené ke druhému CPU ztlačně pomalejší. Pokud má server například dva sokety a celkem 48GB RAM, je zpravidla ke každému CPU připojeno 24GB RAM. Bude-li proces či virtuální PC používat jeden CPU ale více jak 24 GB RAM, použije se část paměti připojené k jinému CPU a tím dojde k propadu výkonu. Problematika numa architektury je ale mimo rozsah tohoto dokumentu a předpokládá se, že správce IT má správné rozdělení paměti ošetřeno.

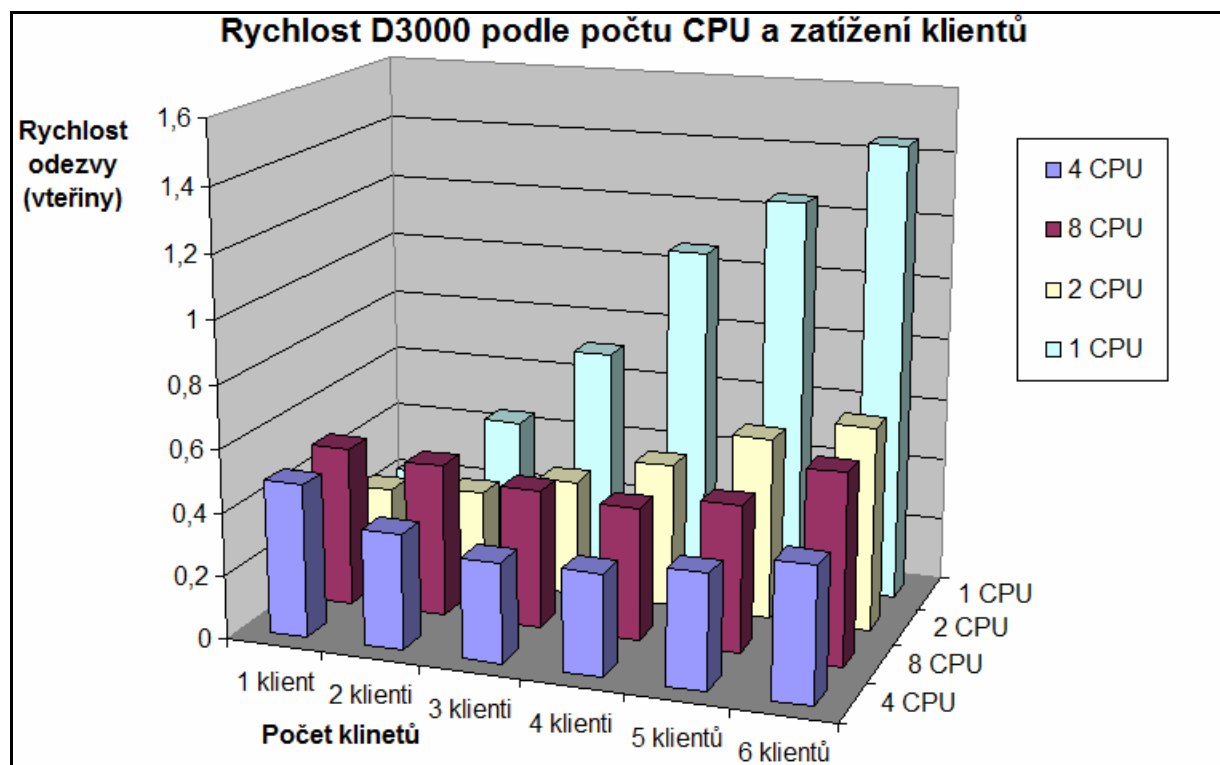
Storage Over-Provisioning

I velikost diskového prostoru může být v případě virtualizace problémem, pokud jednotlivým virtuálním počítačům přidělíte v součtu více celkového diskového prostoru, než fyzicky připojené úložiště nabízí. Například pokud máte v serveru 1TB disk a budete provozovat tři virtuální počítače, kde každému přidělíte 500GB diskové kapacity. Což lze, pokud při vytváření virtuálu použijete dynamickou alokaci. Každý virtuál pak ve skutečnosti použije z disku jen skutečně daty obsazenou část a nikoli hned celý přidělený prostor. Ovšem pokud časem začnou virtuály své diskové prostory více zaplňovat, snadno může dojít k překročení celkové velikosti disku. Při překročení celkové fyzické kapacity úložiště hrozí nevratné ztráty dat, protože OS virtuálních PC nepředpokládají, že když mají přidělený virtuální disk 500GB, může ve skutečnosti dojít místo daleko dříve, pokud ostatní virtuály již své disky zaplňují.

Over-provisioning lze u procesorů a diskového prostoru použít nejen pro více virtuálů, ale i pro jediný virtuální počítač. Pokud ovšem správce přidělení zdrojů nemonitoruje, hrozí problémy s výkonem a dokonce i ztráty dat. Proto v případě použití virtuálního počítače pro docházku věnujte této oblasti náležitou pozornost.

Zajímavým poznatkem u virtualizovaných PC také je, že když se spustí pouze test výpisů docházky a na PC neběží žádná jiná úloha, může být výhodné přidělit takovému virtuálnímu počítači pouze jedno jádro CPU. Zejména pokud hostitelský server má více jader na více fyzických procesorech. Například na serveru HP DL160-G6 s 2x CPU Intel Xeon X5560, kde jsme při testování pro virtuální operační systém Windows 7 Ultimate provedli měření jak pro 4 přidělená jádra CPU tak pro 1 jádro. Paradoxně je při přidělení jediného jádra odezva docházky více než 2x rychlejší (0,2 vteřiny) oproti 4 jádrům CPU kde odezva trvala 0,5 vteřiny. Vysvětlením je samozřejmě rozdělení přidělených jader mezi fyzické CPU jednotky víceprocesorového serveru, kdy pokud vše běží pouze na jednom jádru, je výhodou že nedochází k přepínání kontextu procesů docházky a databáze mezi více CPU jednotek či jader a mnohem lépe se využije cache paměť procesoru jednoho pevného jádra. Tato výhoda platí zejména pokud s docházkou pracuje jediný uživatel v jediném okně webového prohlížeče a nespouští sestavy s asynchronním načítáním dat. Takže se o CPU dělí pouze jediný výpočetní PHP skript programu docházky s jedinou úlohou databáze mysql, kdy se oba tyto procesy vzájemně sériově střídají ve výkonu instrukcí takže se dobře využije paměť cache. Ovšem v momentě kdy tato přesná serializace volání přestává platit, například když s docházkou pracují více pracovníků najednou nebo se volají sestavy umožňující těžit z výhody paralelního načítání dat z více procesů, začne být výhodnější přidělit virtuálnímu serveru docházky více jader jednoho CPU a pokud je uživatelů či paralelních úloh ještě více, začne se projevovat výhoda víceprocesorového serveru. Jak ukazuje tabulka a graf níže s naměřenými časy odezvy docházky na měsíční výpis podle počtu současně paralelně pracujících uživatelů:

Počet jader	Počet klientů					
	1 klient	2 klienti	3 klienti	4 klienti	5 klientů	6 klientů
4	0,486	0,365	0,316	0,321	0,361	0,424
8	0,507	0,485	0,439	0,419	0,465	0,602
2	0,277	0,301	0,372	0,46	0,579	0,646
1	0,239	0,446	0,709	1,065	1,251	1,451



Pro jediného uživatele je nejvhodnější 1 jádro, pro 3 a více uživatelů pak 4 CPU jádra.

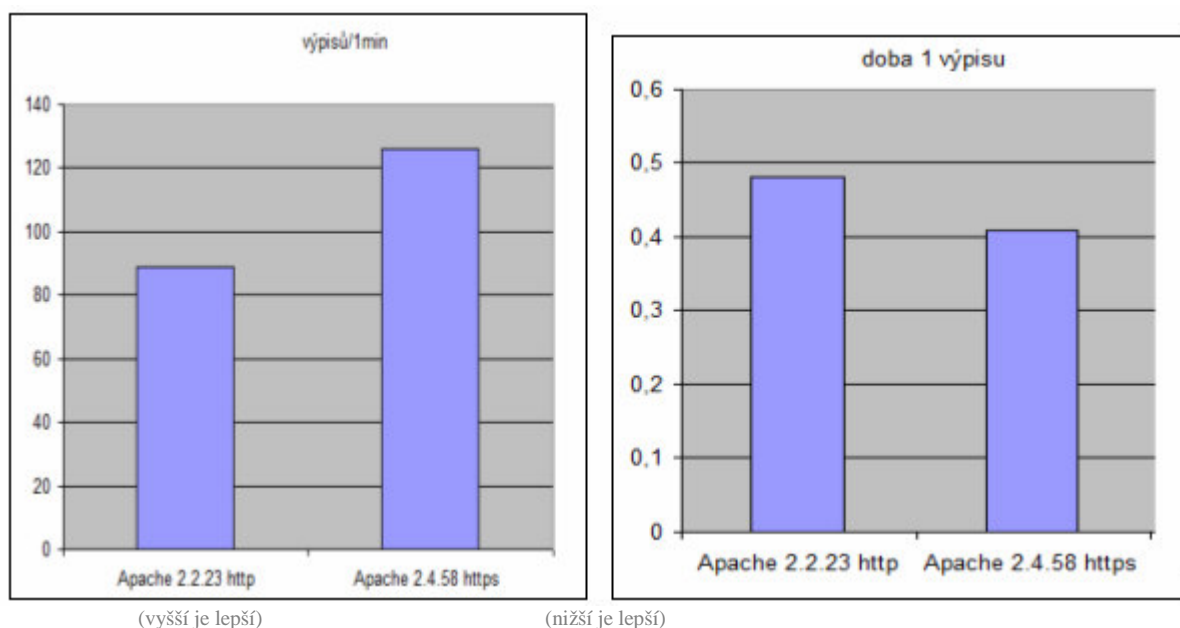
Použití nové verze Apache a případně i DB serveru MariaDB

Od verze programu 9.49 je podporovaná verze Apache 2.4.58 se kterou je docházka například při výpisech v prohlížení docházky o 41% rychlejší než s původní verzí Apache dodávanou na instalačním CD.

Například s poměrně pomalým minipočítačem Firebat AK7 byly naměřeny tyto časy při minutovém běhu zátěžového testu v docházce v menu „Firma / Databáze / Test databáze / Test rychlosti GUI“ s typem testu *Jen výpis* a zatřenou volbou *Zobrazovat jen časy*

Apache 2.2.23 http	89 výpisů za minutu	0.482 vteřiny trval jeden výpis
Apache 2.4.58 https	126 výpisů za minutu	0.408 vteřiny trval jeden výpis
Apache 2.4.58 https MariaDB	137 výpisů za minutu	0.364 vteřiny trval jeden výpis

Z výsledků je vidět, že přestože samotná doba výpočtů v PHP a načítání dat z MySQL v prvních dvou měřeních je téměř shodná (rozdíl je jen 74 ms, tedy zhruba 15%), tak díky menší režii na straně Apache se během minuty načetlo s novým Apache o 37 výpisů víc, tedy doba odezvy se zkrátila o 41%. Což se projeví například v hromadných sestavách docházky či při větší paralelní zátěži s přístupem mnoha zaměstnanců do programu současně. A to přesto že při testu byl u nového Apache použit náročnější šifrovaný protokol https.

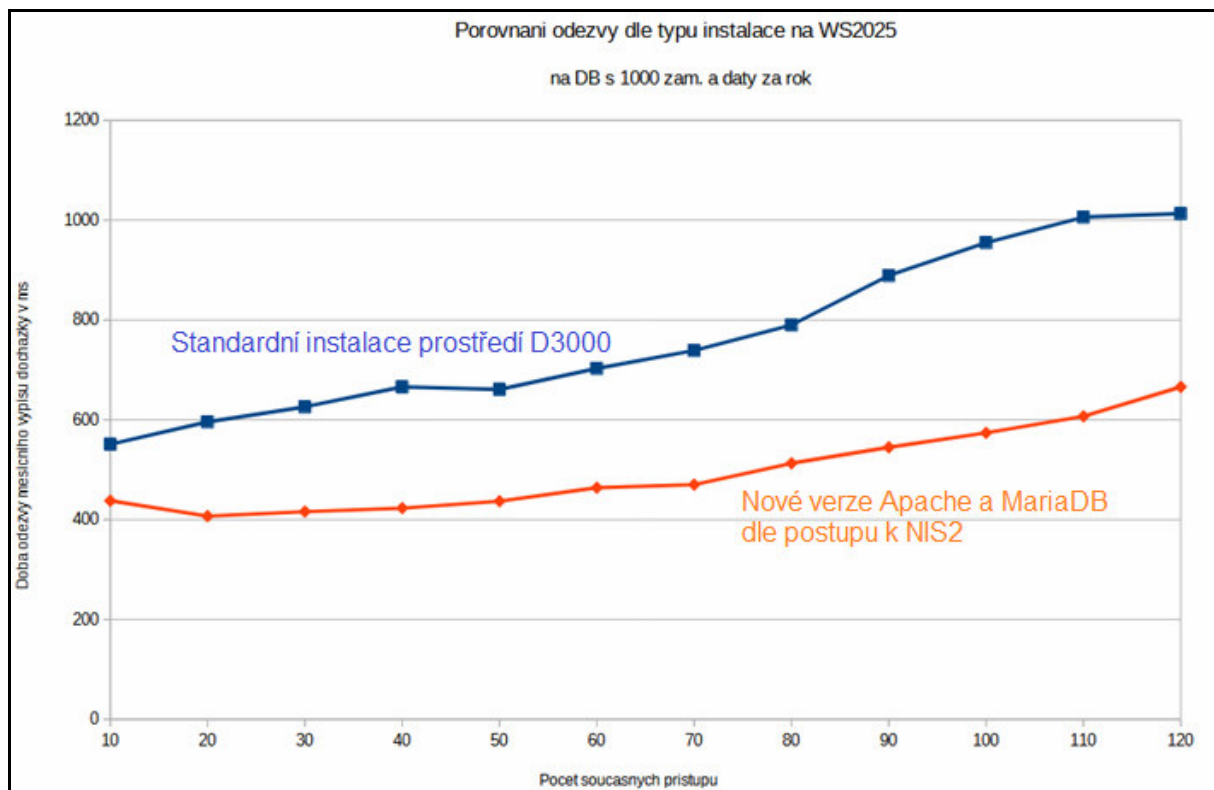


Postup přechodu na novou verzi Apache 2.4.58 naleznete na instalačním či aktualizacím disku docházky od verze 9.49 ve složce */Prirucky* v souboru *Zmena_webserveru.PDF* od strany 32 v bodě nazvaném *Provoz docházky s novým Apache verze 2.4.58*

Pokud máte starší verzi, zakoupíte aktualizaci na webu www.dochazka.eu v sekci *E-shop / Katalog komponent / Příslušenství* jako položku číslo 35 - *Aktualizace SW Docházka 3000*.

Třetí měření s nejlepšími výsledky je pak při instalaci prostředí pro shodu se směrnicí NIS2 kdy se používá nejen nový Apache 2.4.58 ale i MariaDB 11.3.2, což je dostupné od verze programu 9.50 a postup instalace tohoto nového prostředí je pak programem v menu „Zaměstnanci / Nařízení GDPR / Směrnice NIS2“ dole v PDF příručce. Z výsledků měření (uvedených zelenou barvou) je patrné, že s tímto novým prostředím je reakce programu ještě svižnější a třeba u výše uvedeného testu výpisů docházky je odezva oproti původnímu prostředí lepší o 54%.

Graf níže ukazuje výhodu přechodu na nové verze Apache web serveru a MariaDB databázového serveru dle příručky NIS2.pdf pro větší firmy (zde 1 000 zaměstnanců). Zlepšení doby odezvy je zhruba 30%. Oranžová linka grafu s hodnotami doby odezvy (v milisekundách) při měsíčním výpisu docházky dle počtu současných zpracovávaných požadavků ukazuje o kolik je Docházka 3000 s aktualizovanými komponentami rychlejší (kratší doba odezvy) oproti výchozí instalaci prostředí (tedy nižší hodnota je lepší).



Testováno bylo na starším serveru HP Proliant DL160 G6 se dvěma čtyřjádrovými CPU Xeon X5560 (podpora HT byla zapnuta). Operační systém Windows Server 2025 Datacenter edition běžel virtualizovaně pod Proxmox VE 8.2. Virtuální PC mělo přiděleno 8 jader CPU (2 sokety po 4 jádrech) a 16GB RAM. Test proběhl na databázi s 1000 zaměstnanci a generovanou docházkou za 12 měsíců. Jednalo se o test v menu „Firma / Databáze / Test databáze / Test škálování“ spuštění ze 2 PC po 10 měřeních v každém kroku s 6 vlákny na PC (10 až 120 současných požadavků na měsíční výpis docházky, pokaždé na náhodného zaměstnance z 1000 a náhodný měsíc výpisu docházky z období 12 měsíců). U klasické instalace (modrá čára) byl použit nešifrovaný http protokol, u instalace s aktualizovaným prostředím dle příručky k NIS2 pak šifrovaný https protokol.

Využití novější verze MariaDB běžící v Linuxu

(doplněno v 04/2025)

Pokud se rozhodnete například u docházky běžící na virtualizovaných Windows použít databázi MariaDB běžící v hostiteli kterým je OS Linux, může být potřeba v konfiguračním souboru `/etc/my.cnf` doplnit nebo upravit některé parametry tak, aby se použil rychlý úložný engine `myisam`, aby databáze poslouchala na všech síťových rozhraních, aby nedocházelo k prodlevám při překladu DNS jmen na IP adresy a aby se používala rychlá vyrovnávací paměť `QueryCache`. Soubor `/etc/my.cnf` tedy upravte tak, aby obsahoval následující:

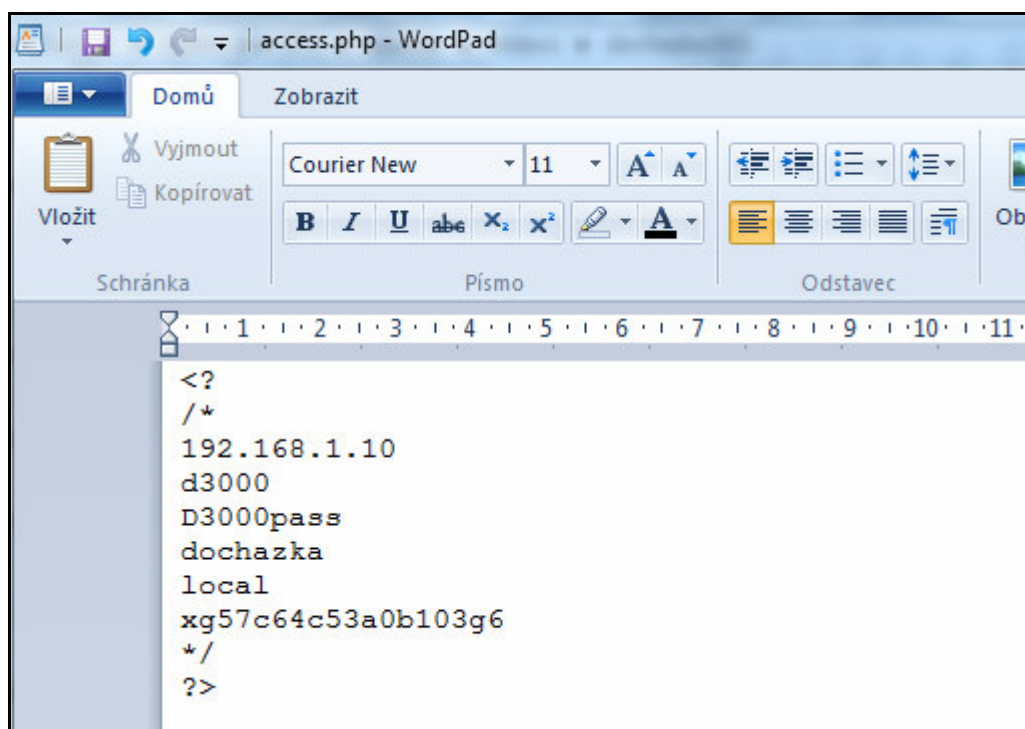
```
max_connections=1900
query_cache_size = 33554432
query_cache_type=1
query_cache_limit=1048576
default-storage-engine = myisam
skip-name-resolve=ON
bind-address=0.0.0.0
```

Nastavení uložte, službu mysql restartujte, zkontrolujte zda firewall neblokuje port 3306 a až poté vytvořte databázi pro docházku. Je vhodné vytvořit uživatele, pod kterým se bude docházka do mysql hlásit, tento bude mít přidělené heslo a veškerá práva. Ukázka vytvoření databáze *dochazka* pro uživatele *d3000* s heslem *D3000pass* a plnými právy přístupu všude: V příkazovém řádku spusťte příkaz *mysql*. Pokud jste při instalaci MySQL vytvořili účet (např. správce) chráněný heslem, pak spusťte *mysql* příkazem *mysql -u správce* a na vyžádání zadejte heslo. Poté vytvoříte databázi a nastavíte práva například takto:

```
create database dochazka;
use dochazka;
create user 'd3000' identified by 'D3000pass';
grant all on *.* to 'd3000'@'%';
```

Pokud by poslední příkaz vypsál chybu, pak zkuste zkrácenou verzi:
`grant all on *.* to 'd3000';`

Nyní je ještě potřeba na PC s Windows, které slouží jako docházkový server, upravit obsah souboru *c:\apache\htdocs\dochazka2001\access.php*. Otevřete jej program *Wordpad*, který správně zobrazí konce řádků. Pro výše uvedený příklad a za předpokladu, že PC s databází má například IP adresu 192.168.1.10, změníte obsah souboru takto:

The image shows a screenshot of the WordPad application window. The title bar reads "access.php - WordPad". The window has a menu bar with "Domů" and "Zobrazit". Below the menu bar is a ribbon with various editing tools. The main text area contains the following code:

```
<?
/*
192.168.1.10
d3000
D3000pass
dochazka
local
xg57c64c53a0b103g6
*/
?>
```

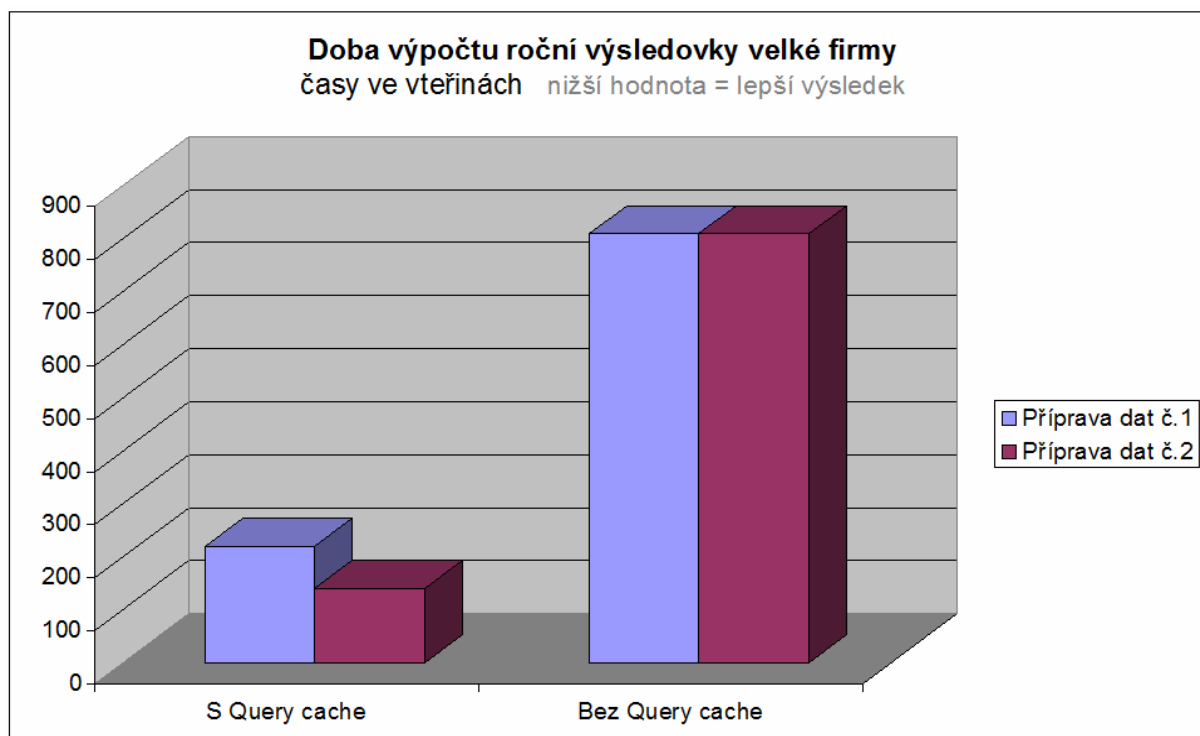
Upravíte tedy řádky 3 až 6 kde na 3. řádku je IP adresa serveru s databází MySQL nebo MariaDB, na 4. řádku je jméno uživatele pod kterým se docházka bude do databáze hlásit a který má plná práva, na 5. řádku je heslo tohoto uživatele a na 6. řádku je jméno databáze pro docházku. Ostatní řádky neměňte, nepřidávejte další řádky ani mezery ani neměňte pořadí řádků a podobně. Soubor uložte a vyzkoušejte fungování docházky.

Pro co nejvyšší výkon je potřeba používat buď databázi MariaDB, nebo poslední pětkové verze MySQL. U novějších verzí MySQL (od 8.x.x výše) totiž firma Oracle odstranila podporu pro *Query Cache* a typu tabulek *MyISAM*, takže databáze pak používá úložný typ tabulek *InnoDB* a s novými verzemi MySQL je docházka mnohem pomalejší. Chcete-li tedy použít jinou verzi databáze než je dodaná přímo s docházkou, nějakou novější a zároveň zajistit co nejrychlejší odezvu programu, používejte MariaDB s výše uvedenou úpravou konfigurace tak, aby se používala *Query Cache* a úložný prostor *MyISAM*.

Důvody použití Query Cache v MySQL/MariaDB

Docházkový systém Docházka 3000 dokáže využít *Query Cache* zabudovanou v databázovém serveru pro výrazné urychlení operací. Jelikož firma Oracle v MySQL verze 8 podporu této vyrovnávací paměti zrušila, je pro docházku z pohledu co nejvyšší rychlosti doporučeno používat buď starší verze MySQL (řady 5 která je součástí instalace docházky), nebo pokud potřebujete použít novější verzi databáze, pak místo MySQL použít MariaDB. Ta stále *Query Cache* i typ tabulek *MyISAM* podporuje a pro program je tedy vhodnější.


Vliv *Query Cache* na rychlost odezvy je poznat zejména při tvorbě velkých měsíčních či ročních sestav u firem s mnoha zaměstnanci využívajících práci na směny, časté čerpání absencí, nočních směn, práce o víkendy a ve svátky, přesčasy a další funkce docházky, které zvyšují potřebu výpočtů. Například při testu na databázi firmy s 1800 zaměstnanci bylo u roční výsledovky změřeno, že s *Query cache* je doba přípravy dat pro sestavu zhruba 4x až 6x kratší (rychlejší) než když se vyrovnávací paměť dotazů nepoužije:

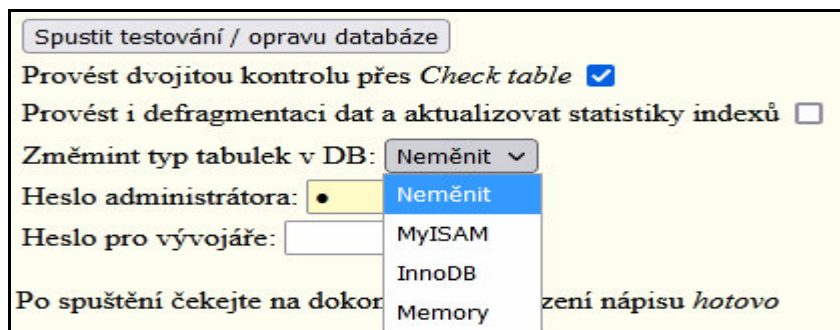


S využitím Query cache tedy výpočty pro zobrazení sestavy trvaly zhruba 3 minuty, bez ní zhruba o 10 minut déle. Obojí bylo měřeno při využití MyISAM typu databázových tabulek. Proto je instalace docházky připravena tak, aby Query cache i rychlý typ tabulek MyISAM využívala. Pokud tedy docházku překonfigurujete tak, aby používala vaši vlastní databázi MySQL a odezva systému se vám bude zdát pomalá, zkontrolujte zda není problém právě v typu tabulek nebo nedostupné vyrovnávací paměti.

IT profesionálové se zaměřením na databáze samozřejmě dokáží prostředí nakonfigurovat tak, aby se dalo použít MySQL verze 8 bez Query cache a s tabulkami InnoDB a nemělo to dopad na výkon docházky. Například s využitím ProxySQL a rozložení databáze na více serverů s multi-master konfigurací pro zajištění odolnosti vůči výpadkům (vysoké dostupnosti), škálovatelnosti atd. Tyto postupy jsou ale již nad rámec této příručky.

Přepnutí typu tabulek u MySQL databáze

Od verze programu 10.12 je doplněna možnost přepínat typ tabulek databáze při používání MySQL nebo MariaDB. Což je určeno jen pro odborníky na databáze, kteří migrují program mezi různými platformami a můžete tak například napravit situaci, kdy jste při přechodu na jinou verzi této databázové platformy (například kvůli požadavkům NIS2) špatně nastavili úvodní konfiguraci nového databázového serveru a místo rychlého MyISAM se teď používá pomalejší InnoDB. Přepínání se provádí v menu "*Firma / Databáze / Test databáze*" lze přepínat stávající databázové tabulky mezi typy MyISAM a InnoDB. Dialog pro přepínání se zobrazí po kliknutí na malou šipku vpravo  a zobrazí se dialog umožňující přepnutí typu tabulek (změnu DB engine).



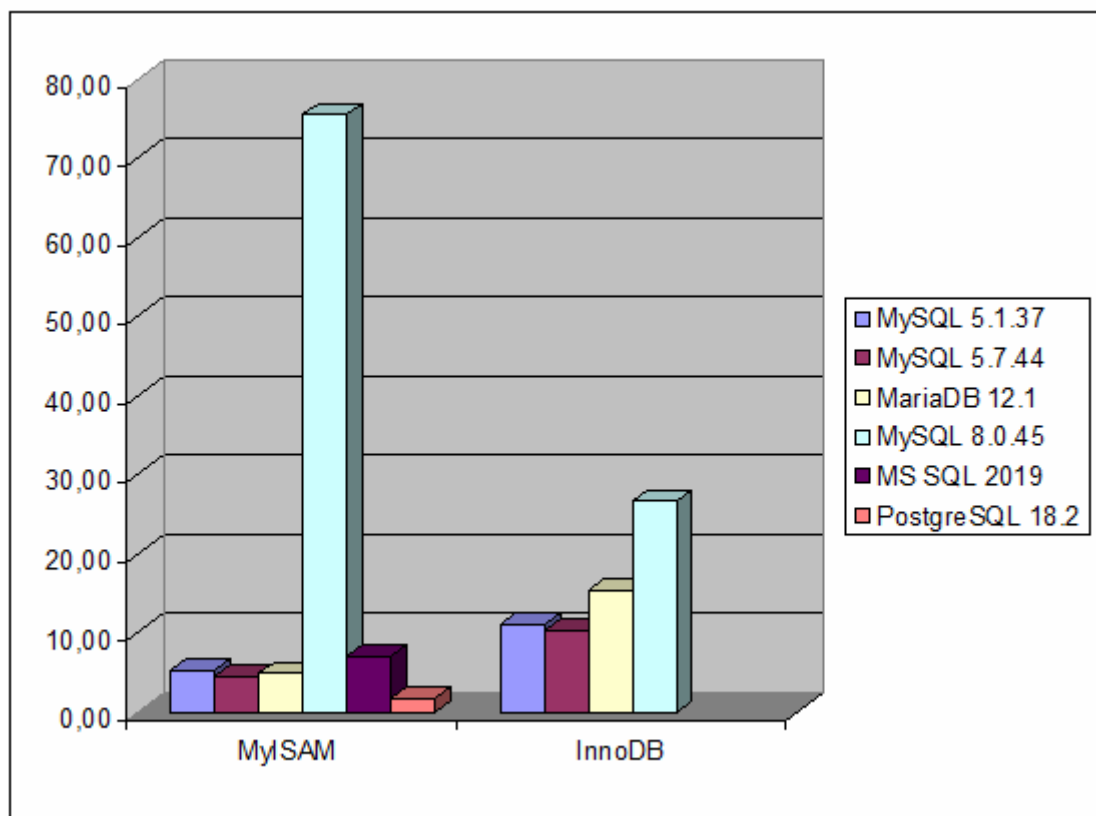
Viz podrobnější nápověda přímo v programu. Přepnutí na typ *Memory* je dostupný jen pro vývojáře a je určený pro testovací účely, kdy nevádí nevratná ztráta dat databáze při restartu.

Přepnutí na InnoDB má smysl jen ve dvou krajních případech. Jednak když provozujete program na velmi nestabilním PC, které se často nečekaně vypíná či resetuje (což je zcela nevhodný počítač pro databázové aplikace a docházku z něj co nejdříve přesuňte jinam) a chcete zlepšit spolehlivost po těchto nečekaných výpadcích a nevádí vám pomalejší odezva programu, lze tabulky přepnout na typ InnoDB, který by měl teoreticky tyto nekorektní pády databáze zvládat lépe. Po přesunu programu na spolehlivé PC pak můžete tabulky přepnout zpět na typ MyISAM a program se tím urychlí. Druhý případ je to, že nemůžete použít MariaDB či pětkovou verzi MySQL a z nějakého důvodu jste museli migrovat na MySQL 8.x.y, kde došlo ke kompletnímu přeprogramování funkcí správy dat a MyISAM je tam jen z důvodu zpětné kompatibility, tedy velice neoptimalizovaný a drtivě zpomalený, takže je možné přepnout alespoň na InnoDB, které je pomalejší jen významně a nikoli drtivě.

Při porovnávání rozdílů mezi výchozí verzí MySQL 5.1.37, poslední pětkovou verzí MySQL 5.7.44, aktuální verzí MySQL 8.0.45, aktuální verzí MariaDB 12.1 a dále MS SQL 2019 a PostgreSQL 18.2 byly při současném běhu 64 klientů se spuštěným zátěžovým testem rychlosti na zobrazení měsíčního výpisu docházky na starém serveru HP-DL160 Gen6 naměřeny tyto výsledky udávající dobu ve vteřinách potřebnou k načtení všech 64 paralelně běžících měsíčních výpisů docházky:

	MyISAM	InnoDB
MySQL 5.1.37	5,25	11,00
MySQL 5.7.44	4,50	10,40
MariaDB 12.1	4,90	15,35
MySQL 8.0.45	75,50	26,65
MS SQL 2019	7,00	
PostgreSQL 18.2	1,70	

Což po přenesení do grafu lépe ukáže výkon docházky na jednotlivých typech DD serverů a u MySQL/MariaDB i typů tabulek. Kratší sloupec znamená lepší výsledek, rychlejší odezvu.



Vidíte, že pro MyISAM je docházka lépe optimalizovaná než pro InnoDB (s výjimkou nedoporučované osmičkové verze MySQL) a že nejrychlejší odezva je na PostgreSQL serveru (který je s MS SQL serverem na grafu informačně zařazen v první sadě sloupců).

Optimalizace využití paměti MySQL serveru

Od verze programu 10.12 byla do programu doplněna i kontrola využití paměti pro indexové soubory. Takže můžete zjistit, zda se indexy využívají efektivně a je na ně v paměti RAM dostatek místa. Kontrola je součástí kontroly konzistence databáze, takže stačí v menu *Firma /*

Databáze / Test databáze kliknout na tlačítko *Spustit testování a opravu databáze*. Ve spodní části pak najdete sekci nazvanou *Spuštěna kontrola konfigurace MySQL* :

Spuštěna kontrola konfigurace MySQL.

Poměr `Key_read_requests / Key_reads = 120:1 (15049300 / 124993) ... Varování`

Poměr by se měl blížit 1000:1 ale u vás je jen 120:1 což je málo a indexy se nevyužívají optimálně. Pokud jste s docházkou delší dobu nepracovali a nebo se do ní teprve přihlásili, není potřeba dělat žádné změny. Ale pokud je docházka na serveru, který běží již dlouho bez restartu OS nebo služby MySQL a s docházkou celou dobu intenzivně pracuje spousta zaměstnanců, je potřeba upravit konfiguraci MySQL dle příručky *Optimalizace výkonu*, kterou najdete v menu *Firma / Návody PDF* a v ní jsou příslušné pokyny v bodě *Optimalizace využití paměti MySQL*.

Na výše uvedeném obrázku je vidět stav, kdy kontrola zjistila, že se paměť pro indexy nevyužívá optimálně a bude potřeba navýšit položku `key_buffer_size` v konfiguraci MySQL v souboru `c:\apache\mysql\bin\my.ini` na hlavním PC docházky. Velikost nové hodnoty záleží na tom kolik má server paměti RAM a jestli je server vyhrazený jen pro docházku a nebo na něm běží i jiné paměťově náročné aplikace. Pokud je vyhrazený jen pro docházkový systém, můžete položku `key_buffer_size` nastavit až třeba na 1/3 dostupného množství RAM. Poté je potřeba restartovat službu MySQL (nebo celý server), aby se nová nastavení uplatnila.

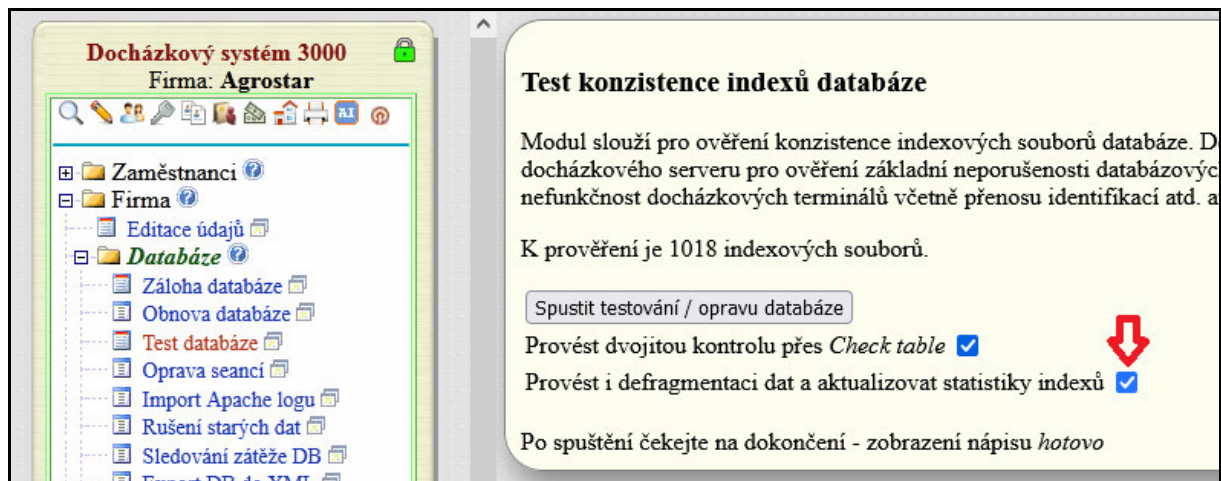
Prizpůsobit doporučujeme i další parametry, ale podrobný rozbor konfigurace MySQL je nad rámec tohoto návodu. Pro příklad níže vidíte možné nastavení vhodných položek v sekci `[mysqld]` souboru `c:\apache\mysql\bin\my.ini` pro virtuální server se 2GB RAM vyhrazeném jen pro docházku:

```
key_buffer_size=512M
read_buffer_size=2M
read_rnd_buffer_size=8M
default-storage-engine = MyISAM
query_cache_size=64MB
query_cache_type=1
query_cache_limit=1MB
max_allowed_packet=100M
open_files_limit=8192
table_open_cache=1024
tmp_table_size=16MB
thread_cache_size=16
back_log=128
```

Po této úpravě se výrazně zlepšila odezva docházky na vytíženém docházkovém serveru, který měl původně nastavené omezené výchozí hodnoty pro univerzální instalaci předpokládající nedostatek paměti RAM. Poměr `Key_read_requests / Key_reads` byl původně dokonce jen zhruba 50 a po optimalizaci konfigurace se zvedl k 800, což se pozitivně projevilo výrazně kratší dobou odezvy. Původně se třeba měsíční sestava *Výsledovka* pro 130 zaměstnanců počítala přes 30 vteřin, po optimalizaci a naplnění keší klesla doba na 8 vteřin.

Defragmentace databáze a optimalizace indexových souborů:

Pokud docházku používáte již mnoho let, doporučujeme v menu *Firma / Databáze / Test databáze* zatrhnout volbu *Provést i defragmentaci dat a aktualizovat statistiky indexů* a kliknout na tlačítko *Spustit testování a opravu databáze*.



System provede nad všemi databázovými tabulkami příkaz *Optimize table*, čímž například znovu uspořádá uložená data a vyčistí smazané záznamy, takže se tabulky zmenší a rychleji se s nimi pracuje. Tuto akci doporučujeme spouštět třeba 1x ročně nebo i v kratších intervalech pokud často přidáváte a mažete zaměstnance a máte jich stovky.