C2115 Praktický úvod do superpočítání

VI. lekce

Petr Kulhánek, Tomáš Bouchal

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta,
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

Obsah

- **Víceuživatelské prostředí**
  proces, vlákno, multitasking, přepnutí kontextu
- **Cvičení**
  efektivita souběžného spouštění aplikací
- **Architektura počítače**
  limitující faktory
- **Cvičení**
  efektivita souběžného síťového přenosu

Víceuživatelské prostředí

prehazet,
CPU, program, process, vlakno, vice programu na jednom CPU, multitasking, prepinani kontextu, vice vlaken, pamet, race condition

Proces a vlákno

Proces (anglicky **process**) je v informatice název pro **spuštěný počítačový program**. Proces je umístěn v operační paměti počítače v podobě sledu strojových instrukcí vykonávaných procesorem. Obsahuje nejen kód vykonávaného programu, ale i dynamicky měnící se data, která proces zpracovává. **Jeden program** může v počítači běžet jako **více procesů s různými daty**. Správu procesů vykonává operační systém, který zajišťuje jejich oddělený běh, přiděluje jim systémové prostředky počítače a umožňuje uživateli procesy spravovat.

Vlákno (též vlákno řízení, anglicky **thread**) označuje v informatice odlehčený proces, pomocí něhož se snižuje režie operačního systému při změně kontextu, které je nutné pro zajištění multitaskingu nebo při masivních paralelních výpočtech. Zatímco běžné procesy jsou navzájem striktně odděleny, sdílí vlákna nejen společný paměťový prostor, ale i další struktury. V rámci jediného procesu je možné vytvořit mnoho vláken. Vlákna usnadňují díky sdílené paměti vzájemnou komunikaci, což však přináší možné komplikace v podobě **souběhu** (anglicky **race condition**).

zdroj: www.wikipedia.cz, upraveno

Multitasking

Multitasking (z angličtiny, multi = mnoho, task = úloha, používán ve víceúlohovém systému) označuje v informatice schopnost operačního systému provádět (přinejmenším zdánlivě) **několik procesů současně**. Jádro operačního systému velmi rychle střídá **na procesoru** běžící procesy (tzv. změna kontextu), takže uživatel počítače má dojem, že běží současně.

Typy:

- nepreemptivní multitasking (dnes se příliš nepoužívá)
- preemptivní multitasking (všechny moderní OS)

V **preemptivním multitaskingu** o přidělování a odebírání procesoru jednotlivým úlohám plně **rozhoduje operační systém**. V **pravidelných intervalech** (typicky zhruba 100× až 1000× za sekundu) přeruší provádění běžícího programu, vyhodnotí aktuální situaci (které úlohy žádají o přidělení procesoru, jejich priority atd.) a nechá běžet buď opět úlohu, kterou přerušil, nebo jinou úlohu, která má zájem o přidělení procesoru. Při **přepnutí úlohy** dochází ke **změně kontextu**. Úloha může v preemptivním multitaskingu dobrovolně požádat o přepnutí kontextu a vzdát se zbytku svého kvanta (úloha takzvaně „usne“ nebo se zablokuje provedením pomalé vstupně-výstupní operace, jako je například čtení dat z pevného disku).

zdroj: www.wikipedia.cz, upraveno

Změna kontextu

Kontext

Pod tímto pojmem si můžeme představit **stav procesoru** (obsah registrů), stav případného koprocesoru, případně i stav dalších zařízení v momentě kdy dojde ke změně kontextu. Tento současný stav procesoru se ukládá buď na zásobník procesu, nebo do připravené oblasti dat v adresním prostoru procesu.

Do kontextu můžeme zahrnout i obsahy různých cache procesoru (např L1 cache nebo TLB): ty se sice při vlastní změně neukládají ani nenačítají, ale **změna kontextu explicitně nebo implicitně zneplatní jejich obsah** a nutnost jejich nového naplnění patří mezi příčiny, proč je přepnutí kontextu na moderních procesorech tak **časově náročné**.

Změna kontextu (anglicky **context switch**) je operace, při níž multitaskingový operační systém **přepíná řízení mezi procesy**. Při tom se ukládá a načítá **aktuální stavu procesoru**. Tento děj se u moderních procesorů opakuje mnohokrát za sekundu. Změny kontextu jsou obvykle výpočetně intenzivní.

zdroj: www.wikipedia.cz, upraveno

Cvičení LVI.1

Zdrojové kódy jsou umístěny v: **/home/kulhanek/Data/C2115/Lesson06**

1. Kolik procesorů obsahuje vaše pracovní stanice? Uveďte jméno stroje, typ CPU a počet CPU. K řešení použijte příkaz **lscpu**.
2. Informace získané v předchozí úloze porovnejte a rozšiřte o informace získané ze souboru **/proc/cpuinfo**.
3. Jaká je velikost vyrovnávací paměti L1, L2 a L3 CPU ve vaší pracovní stanici?
4. Na jaké frekvenci CPU pracuje?
5. Zkompilujte program **load_cpu.f90**

```
$ gfortran -O3 load_cpu.f90 -o load_cpu -lblas
```

6. Určete dobu běhu programu **load_cpu** pomocí příkazu:

```
$ /usr/bin/time --format=%e ./load_cpu
```

7. Proč není přímo použit příkaz **time**? Jaký má význam volba **--format?**

Cvičení LVI.2

1. Ke spuštění programu **load_cpu** použijte následující skript v jazyce bash.

```
#!/bin/bash

N=4     # pocet soubezneho spusteni

for((I=1;I<=N;I++)); do
        ./load_cpu &  # spusti aplikaci na pozadi
done
wait    # ceka na dokonceni vsech aplikaci na pozadi
```

2. Proveďte analýzu funkce skriptu.
3. Změřte dobu běhu skriptu. Měření proveďte pro N=1, 2, 3, ..., NCPU, 2*NCPU, 3*NCPU, 4*NCPU, 5*NCPU, 6*NCPU, kde NCPU je počet CPU dostupných na vaší pracovní stanici. Běžící procesy dále monitorujte příkazem **top** spuštěným v jiném terminálu.
4. Naměřené časy porovnejte s teoretickou délkou běhu vypočtenou z doby běhu programu **load_cpu** na 1 CPU. Určete režii souběžného běhu procesů vztaženou na teoretickou délku běhu. Výsledky diskutujte.

Cvičení LVI.2

Výsledky

wolf01, **4 CPU**, Intel(R) Xeon(R) CPU X3460 @ 2.80GHz, L1: 32kB, L2: 256kB, L3: 8192kB

	load_cpu		
Počet souběžně spuštěných procesů	**Skutečná doba běhu (t_{wall}) [s]**	**Teoretická doba běhu (t_{ideal}) [s]**	**Režie (overhead) [%]**
1	14.5	14.5	0.0
2	15.7	14.5	8.1
3	18.4	14.5	26.8
4	24.5	14.5	69.1
8	50.3	29.0	73.4
12	78.8	43.5	81.1
16	103.3	58.0	78.1
20	135.7	72.5	87.1
24	892.6	87.0	926.0

$$overhead = \frac{t_{wall} - t_{ideal}}{t_{ideal}} 100$$

udává o kolik % je běh aplikace pomalejší než za ideálního stavu

Výsledky

wolf01, **4 CPU**, Intel(R) Xeon(R) CPU X3460 @ 2.80GHz, L1: 32kB, L2: 256kB, L3: 8192kB

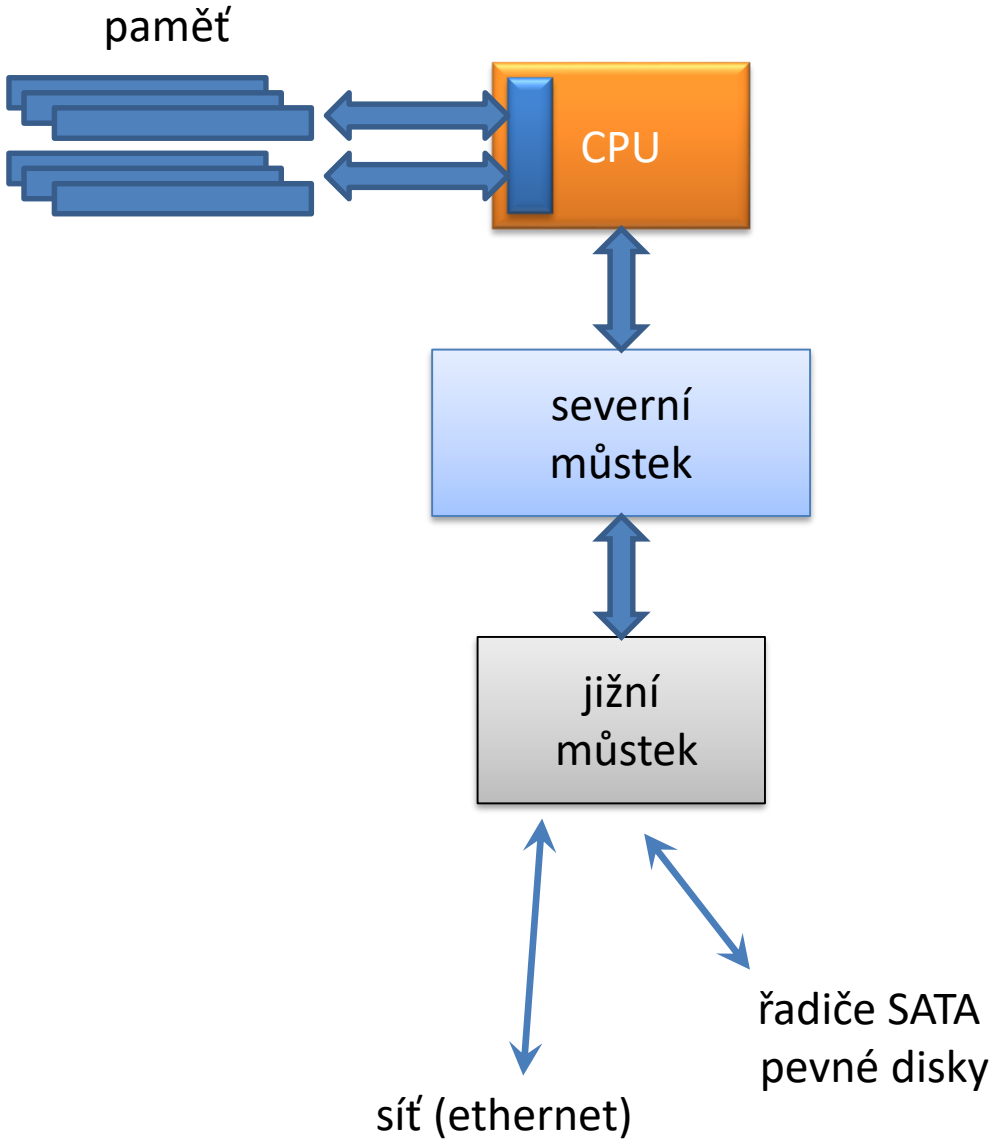
	load_cpu		
Počet souběžně spuštěných procesů	**Skutečná doba běhu (t_{wall}) [s]**	**Teoretická doba běhu (t_{ideal}) [s]**	**Režie (overhead) [%]**
1	14.5	14.5	0.0
2	15.7	14.5	8.1
3	18.4	14.5	26.8
4	24.5	14.5	69.1
8	50.3	29.0	73.4
12	78.8	43.5	81.1
16	103.3	58.0	78.1
20	135.7	72.5	87.1
24	892.6	87.0	926.0

Výsledky

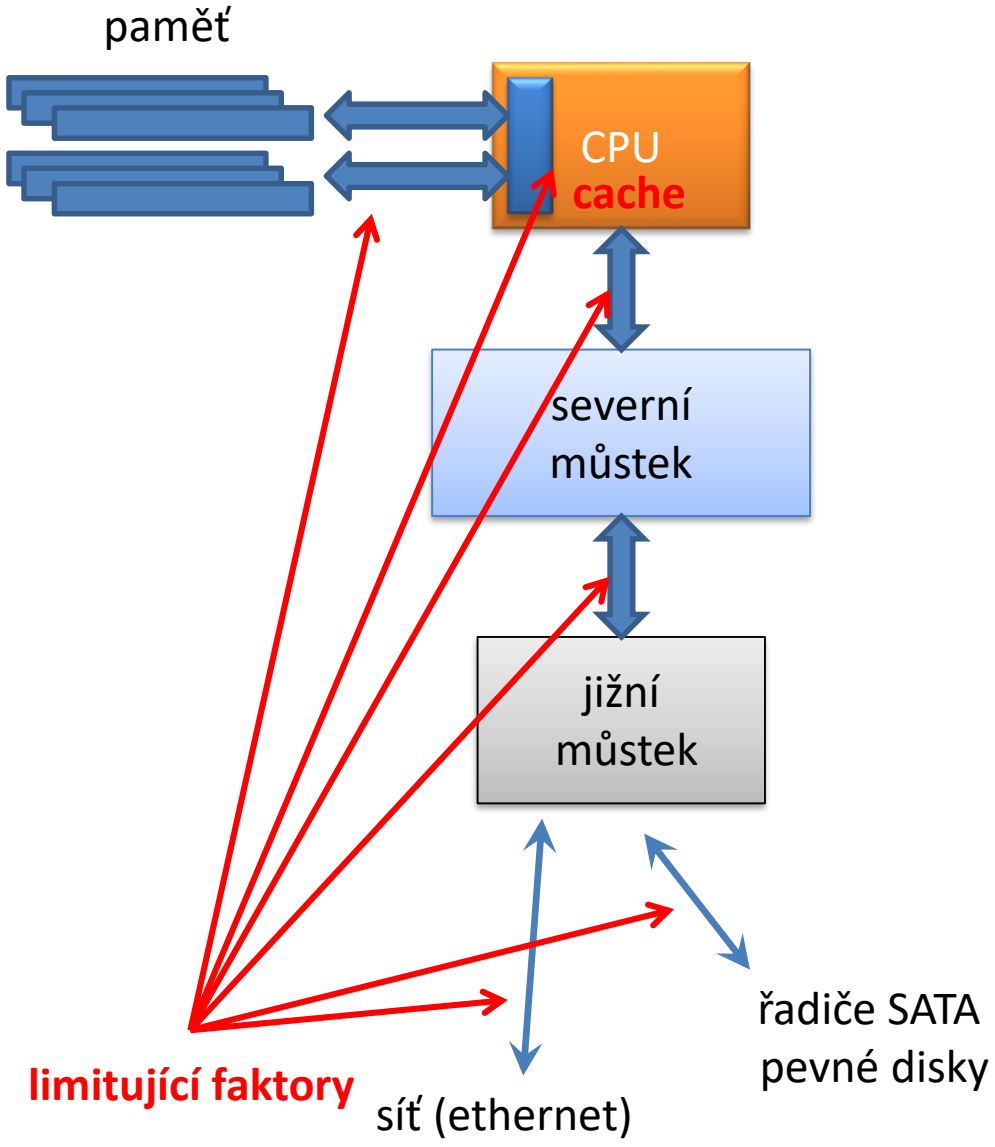
wolf01, **4 CPU**, Intel(R) Xeon(R) CPU X3460 @ 2.80GHz, L1: 32kB, L2: 256kB, L3: 8192kB

	load_cpu		
Počet souběžně spuštěných procesů	**Skutečná doba běhu (t_{wall}) [s]**	**Teoretická doba běhu (t_{ideal}) [s]**	**Režie (overhead) [%]**
1	14.5	14.5	0.0
2	15.7	14.5	8.1
3	18.4	14.5	26.8
4	24.5	14.5	69.1
8	50.3	29.0	73.4
12	78.8	43.5	81.1
16	103.3	58.0	78.1
20	135.7	72.5	87.1
24	892.6	87.0	926.0

Architektura, celkový pohled



Architektura, limitující faktory



Nejrychlejší komponentou je CPU
ostatní komponenty jsou pomalejší

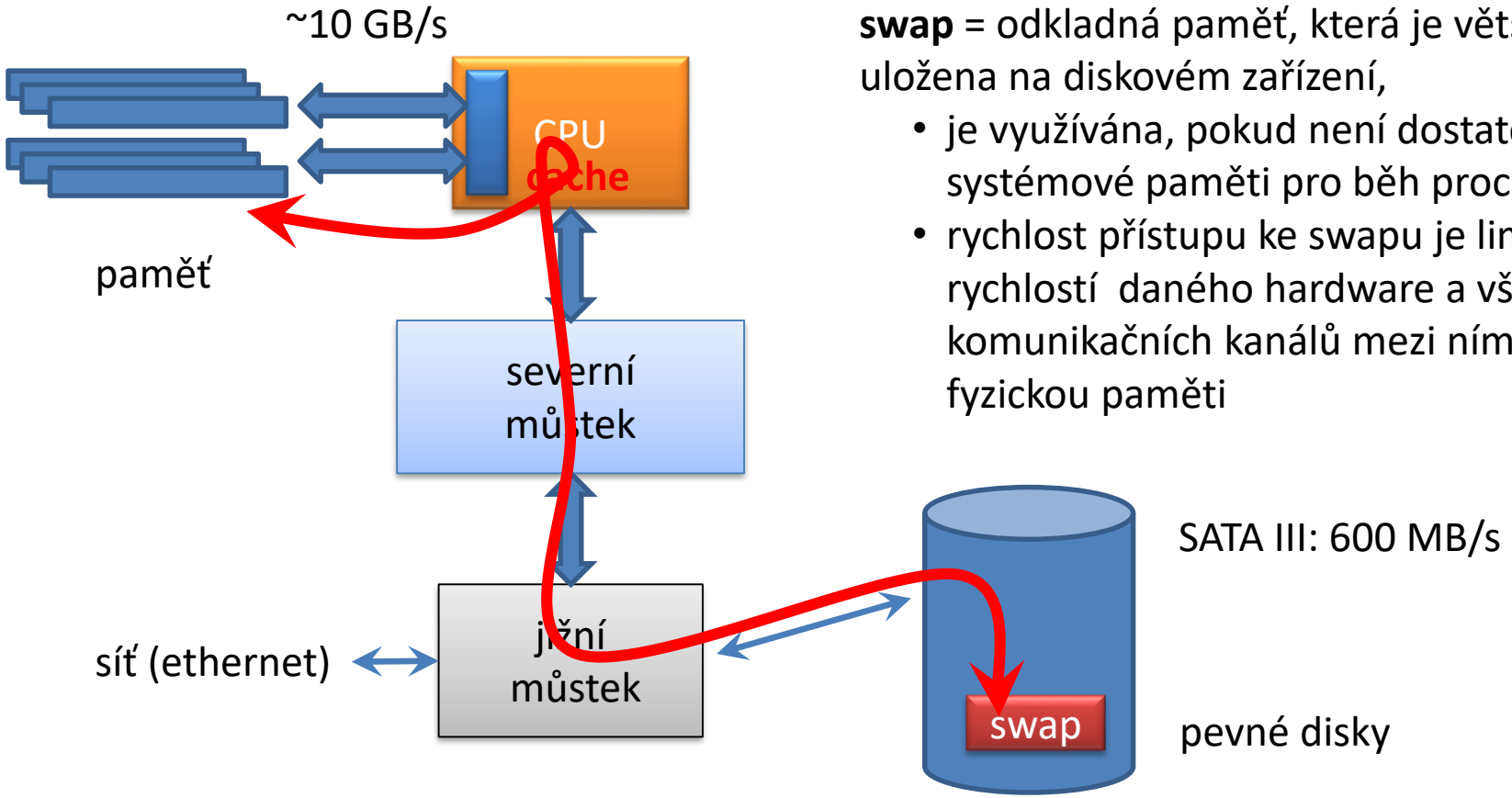
RAM
~10 GB/s

SATA disk
SATA III: 600 MB/s

Síť
10/100/1000 Mb/s

Paměť, swap

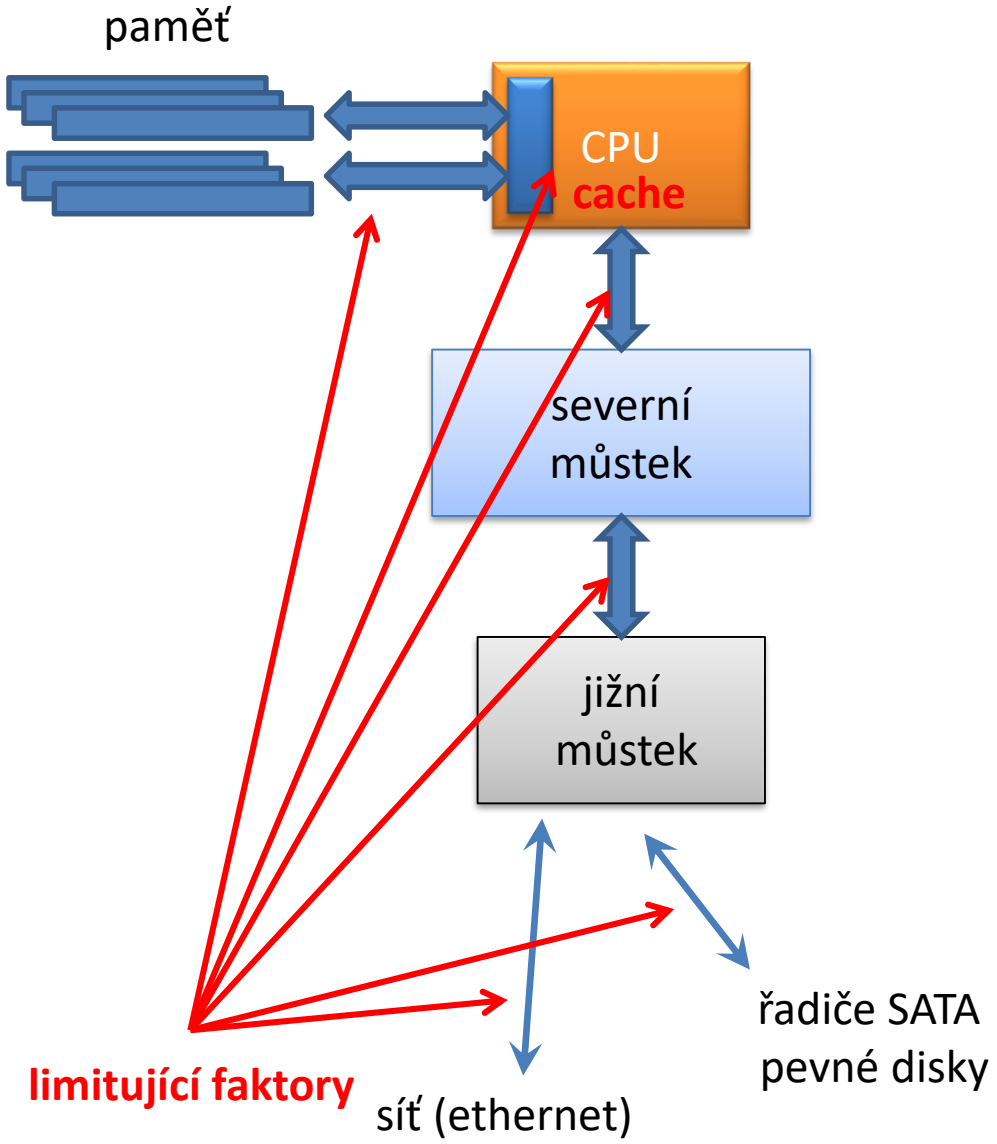
```
top - 12:34:29 up 25 days, 18:13,  8 users,  load average: 0.07, 0.04, 0.05
Tasks: 197 total,   1 running, 196 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.8%us, 18.0%sy,  0.0%ni, 79.7%id,  1.5%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:   8112068k total,  3579460k used,  4532608k free,    39336k buffers
Swap:  4194300k total,    40788k used,  4153512k free,  1242500k cached
```



swap = odkladná paměť, která je většinou uložena na diskovém zařízení,

- je využívána, pokud není dostatek systémové paměti pro běh procesů
- rychlost přístupu ke swapu je limitován rychlostí daného hardware a všech komunikačních kanálů mezi ním a fyzickou pamětí

Architektura, limitující faktory



Nejrychlejší komponentou je CPU
ostatní komponenty jsou pomalejší

RAM
~10 GB/s

SATA disk
SATA III: 600 MB/s

Síť
10/100/1000 Mb/s

vysoké latence

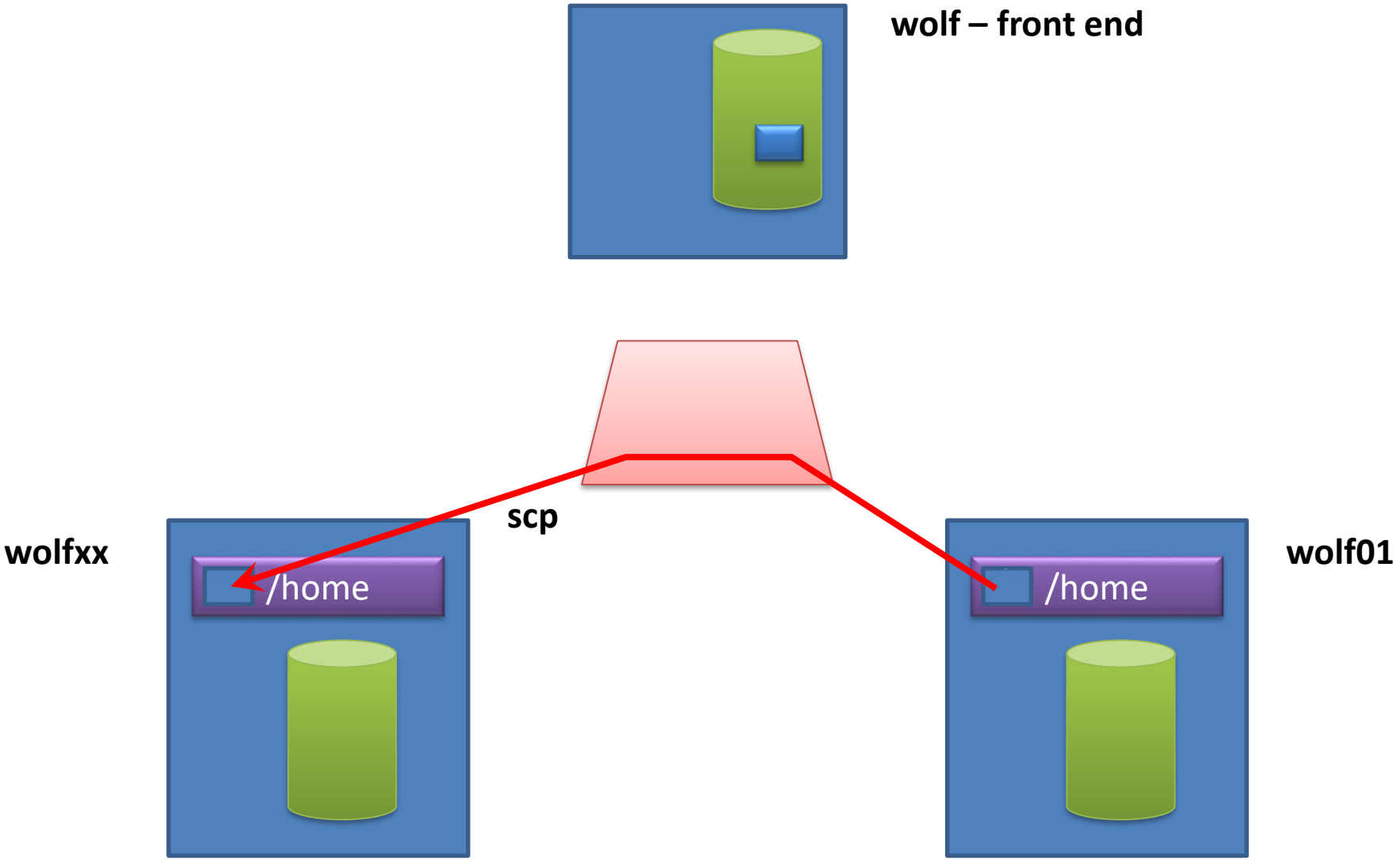
Cvičení LVI.3

1. Příkazem **scp** zkopírujte ze stroje wolf01 soubor **/home/kulhanek/Data/C2115/Lesson05/ubuntu-14.04.1-server-amd64.iso** do vašeho domovského adresáře. Dobu kopírování změřte příkazem **time**. Zkopírovaný soubor smažte.
2. Příkazem **scp** zkopírujte ze stroje wolf01 soubor **/home/kulhanek/Data/C2115/Lesson05/ubuntu-14.04.1-server-amd64.iso** do vašeho scratch adresáře. Dobu kopírování změřte příkazem **time**. Zkopírovaný soubor smažte.
3. Příkazem **scp** zkopírujte ze stroje wolf01 soubor **/scratch/kulhanek/C2115/Lesson05/ubuntu-14.04.1-server-amd64.iso** do vašeho scratch adresáře. Dobu kopírování změřte příkazem **time**.
4. Příkazem **cp** zkopírujte z adresáře scratch soubor **ubuntu-14.04.1-server-amd64.iso** do podadresáře **iso**. Dobu kopírování změřte příkazem **time**. Soubor v adresaři iso smažte.
5. Určete přenosové rychlosti pro různý počet stahování. Určete místo/a, které limituje/í datový přenos.

Týmová organizace!!!
Navrhněte zátěžový test a realizujte jej.

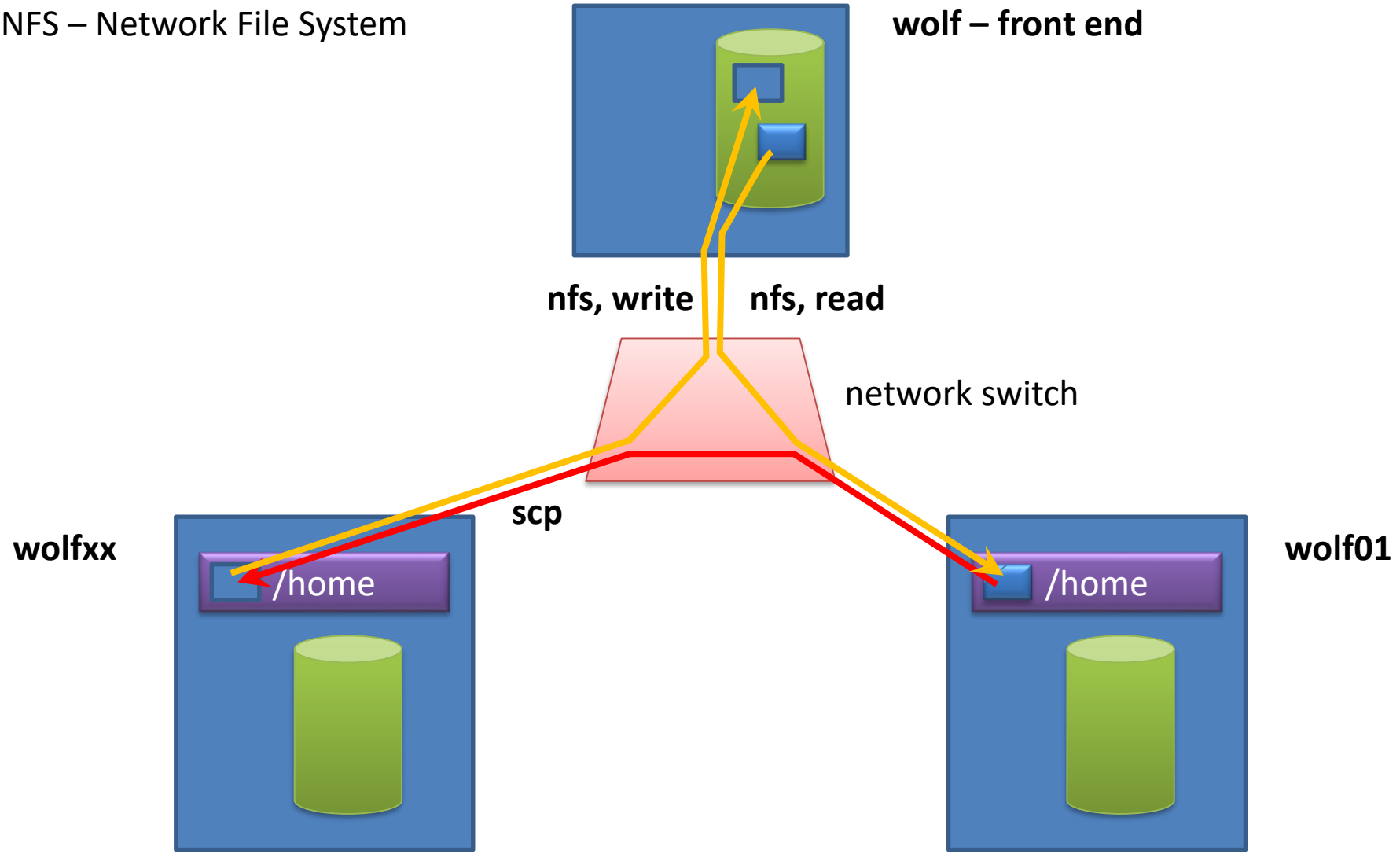
Cvičení LVI.3

Datové toky, LVI.3.1



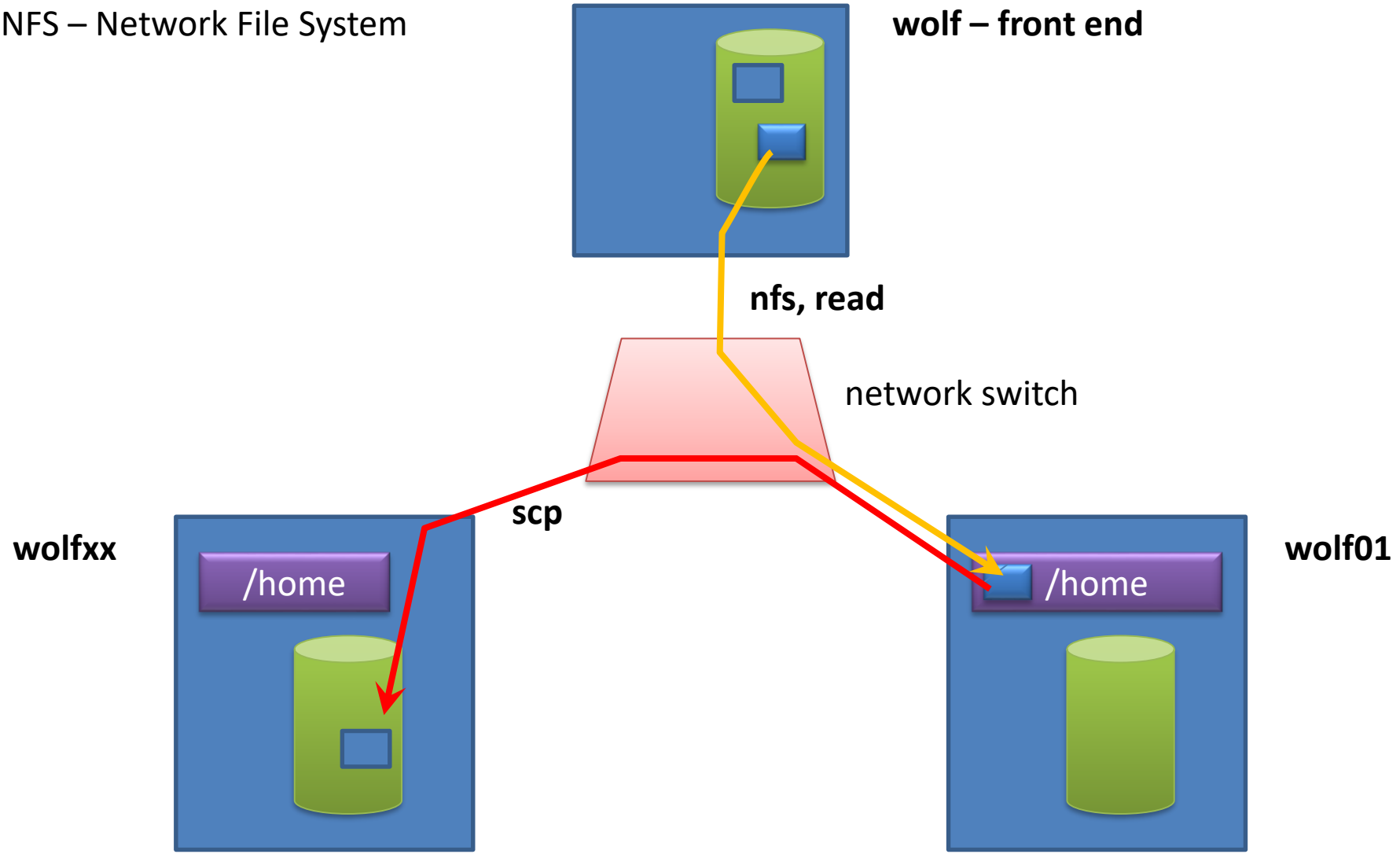
Datové toky, LV.3.1

NFS – Network File System



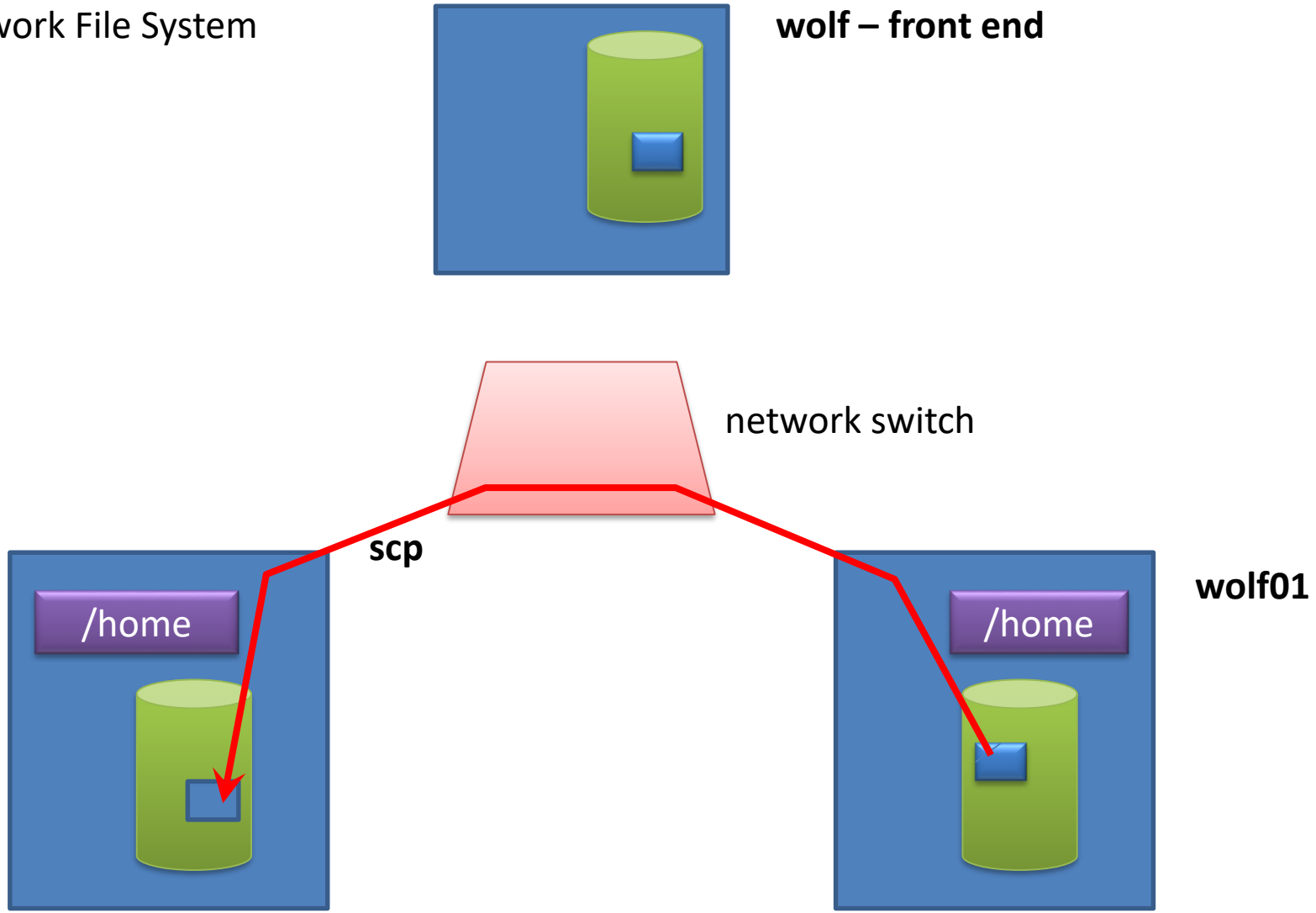
Datové toky, LV.3.2

NFS – Network File System

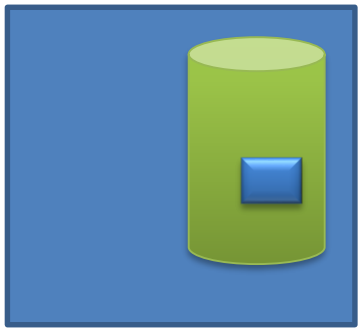


Datové toky, LVI.3.3

NFS – Network File System



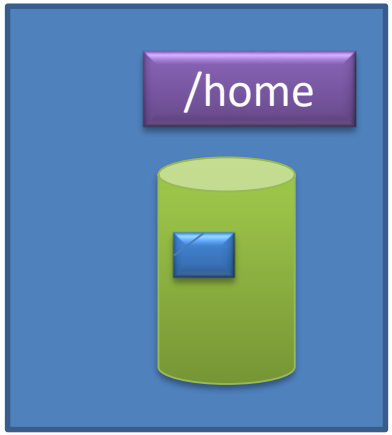
Datové toky, LVI.3.4



wolf – front end

network switch

wolfxx



wolf01

Parametry síťového rozhraní

```
[kulhanek@wolf01 ~]$ ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 2
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: off
```

Infiniband

InfiniBand (abbreviated IB) is a computer network communications link used in high-performance computing featuring very **high throughput** and very **low latency**. It is used for data interconnect both among and within computers.

srovnani s ethernetem, porovnani, ifconfig
informace o rozhrani?