



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



jihořmoravský kraj

OPERAČNÍ SYSTÉMY

Performance monitoring linuxových serverů

Metodický list

Autor: Ing. Marek Kocan, Metodik: Ing. Roman Koláčný

Recenzent: Mgr. Jiří Činčura

Rok vydání: 2023

Performance monitoring linuxových serverů podléhá licenci CC BY-SA 4.0 International License (Offline use:
<http://creativecommons.org/licenses/by-nc-sa/4.0/>).



Obsah

Dovednosti	2
Pracovní prostředí	2
1 Performance monitoring – teoretický základ	3
2 Performance monitoring linuxových serverů – praktické ukázky a cvičení	3
2.1 Pracovní prostředí	3
2.2 Základní nástroj.....	3
3 Performance monitoring.....	3
3.1 free – sledování operační paměti	3
3.2 uptime – základní informace o vytížení systému.....	5
3.3 top – pokročilé informace o vytížení systému	5
3.4 vmstat – pokročilé informace o vytížení systému.....	6
4 Performance monitoring – další příkazy	8
Shrnutí a závěr	10
Seznam použitých zdrojů.....	11

Cíle

Studenti se seznámí s teoretickými základy v oblasti sledování výkonu linuxových serverů a osvojí si vstupní dovednosti monitoringu včetně správné interpretace vybraných parametrů. Součástí jsou jak konkrétní příklady předvedené vyučujícím, tak i samostatné prováděné úkoly.

Student bude schopen vlastními slovy vysvětlit co je to výkonnostní monitoring, jaké základní ukazatele lze sledovat a bude se orientovat v základní příkazech. Student dále dokáže vlastními slovy okrajově popsat možné důsledky chybné interpretace získaných informací či přetížení systémů. Přestože v rámci tohoto scénáře jde o základní možnosti, bude student v dlouhodobém horizontu schopen samostatně využít nové znalosti a dovednosti pro monitoring i v reálných prostředích (vyučující dále studenty seznámí s existencí nejméně jednoho pokročilého nástroje využívaným v praxi pro oblast monitoringu – s ohledem na prostředí školy například s platformou Zabbix).

Dovednosti

Student bude schopen provádět základní sledování výkonnostních parametrů v operačních systémech linuxového typu. Dále student zvládne interpretovat vybrané zobrazené parametry použitých příkazů a spustit příkazy s ohledem na případný průběžný monitoring. V dlouhodobém horizontu bude student schopen aplikovat získané dovednosti pro efektivní práci v linuxových operačních systémech.

Pracovní prostředí

Výuku lze realizovat v prostředí:

- Cylab JCEKB, operační systém Debian (linuxový server)

Pro práci postačí standardní nástroje, některé další budou v rámci průběhu scénáře nainstalovány.

1 Performance monitoring – teoretický základ

Performance monitoring je jednou ze základních aktivit specialistů na operační systémy, zejména pak v případě dohledu na zatížení systémů a hledání příčin problémových stavů (nedostatečný výkon, vyčerpání operační paměti, blokující vstupně výstupní operace.).

K performance monitoringu v případě linuxových operačních systémů lze využít jak samostatné jednoúčelové příkazy, tak i pokročilejší nástroje a řešení (příkladem dohledového systému pracujícího i s výkonnostními metrikami může být například systém Zabbix – www.zabbix.com). Cílem tohoto scénáře není vysvětlit jednotlivé mechanismy správy zdrojů v rámci operačních systémů, toto se předpokládá v jiných částech výuky.

2 Performance monitoring linuxových serverů – praktické ukázky a cvičení

V této části vyučující představí jednotlivé příklady související s monitoringem vybraných výkonnostních parametrů v operačních systémech linuxového typu. Na tyto ukázky bude průběžně navazovat samostatná práce studentů (kontrolní body).

2.1 Pracovní prostředí

Výuku lze realizovat v prostředí Cylab JCEKB – Linuxový server (Debian).

2.2 Základní nástroj

Všechny příklady v rámci tohoto scénáře jsou realizovány v prostředí shellu bash, pro připojení je nutné využít některý z terminálů (například putty či terminál na vzdálené ploše systému).

3 Performance monitoring

3.1 free – sledování operační paměti

Jedním ze základních ukazatelů výkonnostního pohledu jsou informace o operační paměti. V rámci linuxových operačních systémů je standardní součástí program `free`.

```
$ free
```

Výstup:

	total	used	free	shared	buff/cache	available
Mem:	4025224	2794784	837824	12636	392616	1002968
Swap:	999420	0	999420			

Pokud je tento program spuštěný bez jakýchkoli parametrů, vypíše informace v KB pro celkovou fyzickou operační paměť (v případě virtuálního stroje přiřazenou hostitelem), použitou paměť, volnou paměť, paměť obsazenou buffery a vyrovnávací paměť a celkově dostupnou paměť pro spuštění nových procesů bez nutnosti swapu. Dále příkaz vypíše informace o velikosti swapu, jeho využití a volné kapacitě.

Vyučující diskutuje význam jednotlivých částí a upozorní zejména na fakt, že nedostatek volné (free) paměti ještě nemusí znamenat problém – důležití je dostupná paměť (available). Dále vyučující vysvětlí problematiku swapování a upozorní na to, že obsazení swapu nutně neznamená, že systém aktuálně swapuje.

Užitečná je parametr -h, díky kterému jsou hodnoty zobrazené ve srozumitelnějších jednotkách.

```
$ free -h
```

Výstup:

	total	used	free	shared	buff/cache	available
Mem:	3.8Gi	2.7Gi	809Mi	12Mi	390Mi	978Mi
Swap:	975Mi	0B	975Mi			

Dále je užitečný parametr -t, který vypíše součet některých sloupců. Vyučující upozorní na to, že podobně jako u ostatních linuxových příkazů je možné parametry kombinovat.

```
$ free -h -t
```

Výstup (dle skutečného obsahu):

	total	used	free	shared	buff/cache	available
Mem:	3.8Gi	2.7Gi	809Mi	12Mi	390Mi	978Mi
Swap:	975Mi	0B	975Mi			
Total:	4.8Gi	2.7Gi	1.7Gi			

K dispozici je i parametr -s, který spolu se zadanou hodnotou v sekundách umožňuje pravidelné vypisování aktuálních hodnot. Vyučující upozorní na způsob přerušeni (Ctrl+C) i na fakt, že lze zadat například jednu desetinu sekundy (-s 0.1).

```
$ free -h -t -s 1
```

Výstup (zkrácený výpis):

Mem:	total	used	free	shared	buff/cache	available
Swap:	3.8Gi	2.7Gi	809Mi	12Mi	390Mi	978Mi
Total:	975Mi	0B	975Mi			
	4.8Gi	2.7Gi	1.7Gi			
Mem:	total	used	free	shared	buff/cache	available
Swap:	3.8Gi	2.7Gi	809Mi	12Mi	390Mi	978Mi
Total:	975Mi	0B	975Mi			
	4.8Gi	2.7Gi	1.7Gi			
...						

Kontrolní bod

Studenti zjistí celkovou velikost fyzické (operačnímu systému přiřazené) operační paměti a odpoví na otázku, proč celková paměť se nerovná součtu všech ostatních částí.

3.2 uptime – základní informace o vytížení systému

Hlavním účelem příkazu `uptime` je sice zobrazení informací o tom, jak dlouho je systém v provozu, nicméně část jeho výstupu lze využít pro získání informací o vytížení systému z pohledu poslední minuty, posledních pěti minut a posledních 15 minut. Lze jej tak využít nikoli pro získání aktuálního stavu, ale pro průměrné hodnoty za uvedená časová období.

```
$ uptime
```

Výstup:

```
21:04:19 up 34 min, 1 user, load average: 0.79, 0.18, 0.06
```

Důležité ve výstupu jsou z výkonnostního pohledu tučně označené hodnoty. Zjednodušeně řečeno odpovídají počtu spuštěných procesů a hodnoty by vždy měly být při analýze porovnávány vůči celkovému počtu jader CPU. Při jednom jádru je hraniční hodnota v `uptime` okolo 1, při 4 jádrech 4 apod. Hodnota 1 při 4 jádrech odpovídá cca 25% vytížení.

Vyučující předvede konstrukci `watch -d -n 1 uptime` a vysvětlí jednotlivé parametry (zobrazování změn a obnovení výstupu po jedné sekundě). Vyučující dále informuje studenty o tom, že informace o procesorech je možné získat například pomocí příkazu `lscpu`.

3.3 top – pokročilé informace o vytížení systému

Příkaz `top` zobrazuje přehledové informace o vytížení systému včetně podrobných informací u jednotlivých procesů.

```
$ top
```

Výstup (zkrácený výpis):

```
top - 21:21:13 up 51 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 78 total, 1 running, 77 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3930.9 total, 116.6 free, 2732.3 used, 1082.0 buff/cache
MiB Swap: 976.0 total, 976.0 free, 0.0 used. 967.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0  98244 10092  7792 S   0.0   0.3   0:00.64 systemd
    2 root        20   0     0     0     0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/
    9 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 mm_percpu_wq
   10 root        20   0     0     0     0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
   11 root        20   0     0     0     0 S   0.0   0.0   0:00.00 rcu_tasks_trace
   12 root        20   0     0     0     0 S   0.0   0.0   0:00.01 ksoftirqd/0
   13 root        20   0     0     0     0 I   0.0   0.0   0:00.08 rcu_sched
   14 root         rt    0     0     0     0 S   0.0   0.0   0:00.01 migration/0
   15 root        20   0     0     0     0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root        20   0     0     0     0 S   0.0   0.0   0:00.00 kdevtmpfs
   18 root         0 -20     0     0     0 I   0.0   0.0   0:00.00 netns
...

```

Jde o interaktivní průběžně aktualizovaný obsah, mezi základní ovládací klávesy patří:

- šipky a Home, End, PgUp, PgDown – posun v rámci výstupu
- E – přepínání jednotky paměti v přehledové části

- e – přepínání jednotky paměti v procesové části
- t – přepínání vytížení CPU mezi číselnými hodnotami a grafem (2 varianty)
- l – zobrazení vytížení CPU pro jednotlivá jádra (pozor, jde o hodnotu jedna, nikoli el)
- m – přepínání využití RAM mezi číselnými hodnotami a grafem (2 varianty)
- z – podbarvení
- y – zvýraznění aktuálně skutečně běžících procesů
- P – třídění podle vytížení CPU
- M – třídění podle využití RAM
- N – třídění podle čísla procesu
- T – třídění podle doby běhu

Vyučující diskutuje jednotlivé hodnoty (viz například `man top`) a předvede ovládání. Dále v případě dostatku času nainstaluje program `htop` – `apt install htop` – a předvede tuto alternativu k příkazu `top`. V případě `htop` doporučeno předvést strom procesů a ukončení (kill) některého nedůležitého procesu.

Podobně jako ostatní programy a příkazy má i `top` řadu parametrů, vyzdvihnout lze `-d`, který spolu se zadanou hodnotou v sekundách umožňuje pravidelné vypisování aktuálních hodnot.

Kontrolní bod

Studenti si v případě `top` vyzkouší různé hodnoty parametru `-d` a zjistí, jaký proces nejvíce vytěžuje výpočetní zdroje (CPU).

3.4 vmstat – pokročilé informace o vytížení systému

Přínosným příkazem je `vmstat`, který umožňuje pravidelně vypisovat aktuální informace o řadě parametrů včetně paměti, CPU, diskových operací či swapování. Současně má k dispozici přepínače pro zobrazení statistických informací.

```
$ vmstat
```

Výstup:

```
procs -----memory-----  ---swap--  -----io----  -system--  -----cpu-----
 r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa  st
 0  0     0 118848 16404 1091944    0    0   89  386    0  77  0  0 98  1  0
```

Vyučující upozorní na sloupce pro paměť, swap (in/out), io (in/out) a vytížení CPU (zejména na `us` – uživatelské procesy, `sy` – systémové, `id` – idle).

Výhodné je spouštět `vmstat` s přepínačem `-t` a zadáním číselné hodnoty pro interval automatického opakování výpisu (toto je využitelné při přeměrování do souboru například pro získání přehledného protokolu během sledovaného období apod.). Přepínač `-t` zajišťuje přidání časové známky, číselná hodnota pak samotné opakování (v intervalu dle zadaných sekund).

```
$ vmstat -t 1
```

Výstup (zkrácený výpis):

```
procs -----memory----- ---swap-- -----io----- -system-- -----cpu----- -----timestamp-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st 2022-11-28 21:40:58 CET
 0 0 0 118848 16436 1091984 0 0 85 367 0 76 0 0 98 1 0 2022-11-28 21:40:59
 0 0 0 118840 16436 1091984 0 0 0 0 0 57 0 0 100 0 0 2022-11-28 21:41:00
 0 0 0 118840 16436 1091984 0 0 0 0 0 60 0 0 100 0 0 2022-11-28 21:41:01
 0 0 0 118840 16436 1091984 0 0 0 0 0 54 0 0 100 0 0 2022-11-28 21:41:02
 0 0 0 118840 16436 1091984 0 0 0 0 0 59 0 0 100 0 0 2022-11-28 21:41:03
 ...
```

Přepínač `-d` zajistí vypsání informací z pohledu disků.

```
$ vmstat -d
```

Výstup:

```
disk- -----reads----- -----writes----- -----IO-----
 total merged sectors ms total merged sectors ms cur sec
sr0 15 0 0 1 0 0 0 0 0 0 0 0
sda 5381 2899 724274 111512 5998 2408 3141481 1331491 0 62
```

Vyučující orientačně diskutuje jednotlivé sloupce.

Popisovaný příkaz nabízí i statistické výstupy, v případě paměti jde o parametr `-s`, v případě disků o parametr `-D`.

```
$ vmstat -s
```

Výstup:

```
4025224 K total memory
2797940 K used memory
 76340 K active memory
1085756 K inactive memory
118848 K free memory
 16460 K buffer memory
1091976 K swap cache
999420 K total swap
 0 K used swap
999420 K free swap
 725 non-nice user cpu ticks
 8 nice user cpu ticks
 891 system cpu ticks
442981 idle cpu ticks
 5281 IO-wait cpu ticks
 0 IRQ cpu ticks
 64 softirq cpu ticks
 0 stolen cpu ticks
362137 pages paged in
1571000 pages paged out
 0 pages swapped in
 0 pages swapped out
 16 interrupts
 335355 CPU context switches
1669663774 boot time
 1222 forks
```

Vyučující diskutuje vybrané výstupy, dle své volby, ale zaměří se na počty stránek zápisů a čtení do/ze swapu.

```
$ vmstat -D
```

Výstup:

```
2 disks
3 partitions
```

```
5396 total reads
2899 merged reads
724274 read sectors
111513 milli reading
6042 writes
2424 merged writes
3142001 written sectors
1331885 milli writing
0 inprogress IO
63 milli spent IO
```

Vyučující diskutuje vybrané výstupy.

Kontrolní bod

Studenti pomocí příkazu `vmstat` vytvoří jednoduchý protokol (soubor) obsahující časové značky a vypisující informace o discích po dvou sekundách. Vyučující případně připomene, jak lze výstup přesměrovat do souboru. Dále studenti pomocí téhož příkazu zjistí celkový počet disků v systému, celkový počet čtení z disků od spuštění počítače a celkový počet stránek zapsaných do swapu.

4 Performance monitoring – další příkazy

V případě dostatku času či další časové dotaze předvede vyučující následující příkazy:

- `iostat` (vyžaduje instalaci `apt install iostat`) – vyučující diskutuje tento monitor vstupně výstupních operací, zaměří se na klávesu o (shodně přepínač `-o`), která přepíná zobrazení mezi všemi procesy a pouze procesy aktuálně provádějící některou ze vstupně/výstupních operací. Dále zmíní třídění (šipky) a reverzní třídění (klávesa `r`) – třídění je identifikováno znakem `>`, resp. `<` u jednotlivého sloupce. Vhodné je pro demonstraci spustit nějaký diskově náročný proces, například v jiném terminálu generování náhodného souboru pomocí `cat /dev/urandom >soubor.out`).
- `iostat` – vyučující předvede program principiálně obdobný `vmstat`, ale zaměřený na I/O operace.
- `iftop` (vyžaduje instalaci `apt install iftop`) – vyučující předvede program zobrazující statistické informace o síťové komunikaci.
- `lsof` – vyučující předvede program informující o otevřených souborech, vhodným příkladem je `lsof -i TCP:22` zobrazující informace o otevřených souborech pro TCP komunikaci na portu 22 (ssh). Vyučující zmíní možnou vazbu na program `iftop` při abnormálním vytížení a využití `lsof` jako jednoho z prostředků pro prvotní identifikaci, zda problém není například v přenosu velkých souborů. V této souvislosti může vyučující zmínit i `netstat`, konkrétně `netstat -s` vypisující statistické informace o síťové komunikaci od startu operačního systému.

```
$ netstat -s
```

Výstup:

```
Ip:
Forwarding: 2
5365 total packets received
3 with invalid addresses
```

```
0 forwarded
0 incoming packets discarded
4810 incoming packets delivered
6826 requests sent out
132 dropped because of missing route
Icmp:
  1 ICMP messages received
  0 input ICMP message failed
  ICMP input histogram:
    echo requests: 1
  1 ICMP messages sent
  0 ICMP messages failed
  ICMP output histogram:
    echo replies: 1
IcmpMsg:
  InType8: 1
  OutType0: 1
Tcp:
  4 active connection openings
  1 passive connection openings
  0 failed connection attempts
  0 connection resets received
  1 connections established
  4564 segments received
  8121 segments sent out
  0 segments retransmitted
  0 bad segments received
  0 resets sent
Udp:
  55 packets received
  0 packets to unknown port received
  0 packet receive errors
  55 packets sent
  0 receive buffer errors
  0 send buffer errors
  IgnoredMulti: 190
UdpLite:
TcpExt:
  4 TCP sockets finished time wait in fast timer
  6 delayed acks sent
  460 packet headers predicted
  2419 acknowledgments not containing data payload received
  1326 predicted acknowledgments
  Detected reordering 1 times using SACK
  TCPBacklogCoalesce: 3
  TCPSackShiftFallback: 1
  TCPRcvCoalesce: 27
  TCPAutoCorking: 2929
  TCPOrigDataSent: 8021
  TCPhystartTrainDetect: 1
  TCPhystartTrainCwnd: 20
  TCPDelivered: 8020
IpExt:
  InBcastPkts: 190
  InOctets: 476308
  OutOctets: 5153426
  InBcastOctets: 25060
  InNoECTPkts: 5382
```

Kontrolní bod

Studenti zajistí generování velkého souboru s náhodným obsahem (využijí /dev/urandom) a postupně sledují vytížení pomocí vhodného výběru z předvedených nástrojů.

Shrnutí a závěr

Studenti se seznámili jak se základními příkazy pro sledování výkonnostních parametrů v operačním systému linuxového typu (většina příkazů je plně přenositelných a dostupných – v nejhorším případě po nainstalování – na drtivě většině distribucí, a to i v rámci dalších unixových systémech). Dle svého uvážení může vyučující připravit pro studenty například test, a to jak z pohledu teorie, tak i praktického opakování nad zdokumentovanými postupy, případně zadat za domácí úkol například na následující témata:

- vytvořte soubor s protokolem informující za období jednoho dne v intervalu 5 minut o velikosti dostupné operační paměti (náročnost 4/10), *například free*
- vytvořte soubor s protokolem informujícím za období jednoho dne v intervalu 5 minut o vytížení CPU (náročnost 4/10), *například vmstat*
- diskutujte statistické informace o diskových operacích v období od startu operačního systému (náročnost 7/10), *například vmstat -D*

Seznam použitých zdrojů

SPI. *Manuálové stránky*. Dostupné z: <https://manpages.debian.org>