

Storage Over IP

Protokol HyperSCSI

Filip Staněk, Jan Haluza
Technická zpráva CESNET-u č. 23/2003
23. listopadu 2003

Úvod

Tato zpráva popisuje praktické zkušenosti použitelnosti protokolu HyperSCSI a stručný popis samotné technologie. Je částečně i srovnáním s protokolem iSCSI, který byl popsán v předchozích zprávách [1], [2] a [3].

Protokol HyperSCSI, popsán v příslušném dokumentu [3], je síťový protokol, který má sloužit k přenosu SCSI příkazů a dat po síti. Původně vznikl v roce 2000 v Data Storage Institute (DSI) jako alternativa k iSCSI protokolu [4], který coby alternativa protokolu Fibre Channel pro sítě typu Ethernet nevyhovoval. Vyvinul se z výzkumu možnosti zapouzdření SCSI do Ethernetových rámců. Stejně jako iSCSI se i HyperSCSI (dále jen HSCSI) snaží především poskytnout levnější variantu pro budování SAN než s použitím technologie Fibre Channel (dále jen FC). Oproti iSCSI se navíc snaží poskytnout vyšší výkon, který je v obou variantách horší než při použití technologie FC.

Možnosti HSCSI

Protokol je navržen se dvěma variantami síťového přenosu. První, momentálně implementována a zde popsána, je označena jako HS/Eth a je založena na přenosu dat přímo nad Ethernet vrstvou. Dle autorů funguje i nad bezdrátovými sítěmi typu 802.11b. Výhledově se počítá s implementací, která by pro přenos používala IP vrstvu (HS/IP). Bohužel není vůbec jasné kdy varianta HS/IP bude k dispozici.

Současná implementace je k dispozici pouze v podobě software a to pro OS Linux a Windows 2000. Je použit přístup klient – server. Projekt je vyvíjen jako tzv. „Open Source“, ovšem pouze pro platformu Linux. Zde jsou přístupné zdrojové kódy i binární instalační balíčky jak serveru tak i klienta. Pro platformu Windows je k dispozici pouze zkušební binární verze klientské části, bez zdrojových kódů. Současná verze je 20030930 a umožňuje zpřístupnění následujících zařízení:

- SCSI zařízení (pevné disky, optické mechaniky, páskové zařízení, scanner-y)
- IDE zařízení (pevné disky, optické mechaniky)
- softwarově řešené RAID pole a LVM svazky
- USB zařízení (pevné disky, výměnné disky, a další USB-Storage kompatibilní zařízení)

Protokol HSCSI řeší také zabezpečení přenosu dat a to sice těmito mechanismy:

- Ochrana dat před neoprávněným přístupem
Klient se musí serveru autentikovat pomocí kombinace jméno/heslo. Toto lze nastavit pro každý exportovaný svazek.
- Zajištění integrity přenášených dat
Současně je implementován pouze algoritmus SHA1 HMAC, ale počítá se s dalšími. Architektura je navržena modulárně, lze si dopsat i vlastní mechanismus.
- Kryptování přenášených dat
Implementován algoritmus AES (Rijndael), konkrétně 128 bitová varianta. Opět se do budoucna počítá s dalšími algoritmy.

Srovnání HSCSI s iSCSI

Protože je protokol HSCSI dostupný pouze pro síť Ethernet, jeho největší nevýhodou je nemožnost směrování a tedy nemožnost rozlehlejšího nasazení. Dále je třeba si uvědomit, že HSCSI není žádný oficiální standard (jako např. iSCSI) a tedy není podporován žádným výrobcem zabývajícím se HW řešeními *storage* zařízeními. Zatím ani žádný výrobce podporu neplánuje.

Výhody proti iSCSI jsou především nižší zátěž jak sítě, tak i koncových systémů (klient, server). Je to způsobeno jak návrhem a implementací protokolu, tak faktem že iSCSI používá pro přenos dat TCP/IP které má větší režii. Dále díky plně funkční programové implementaci HSCSI serveru i klienta (na rozdíl od iSCSI, kde jsou problémy především s částí serveru), lze postavit řešení s HSCSI na běžném vybavení a není třeba dokupovat specializované a drahé HW řešení. Může se tak hodit na budování malých levných SANů.

Použité vybavení

HW:

- 3 x Pentium III 500 Mhz (512 kB cache), 384 MB RAM, Intel EtherExpress100 NIC, SCSI controller Adaptec AHA-2940U2/W, HDD Seagate ST136403LW (Type: Direct-Access, ANSI SCSI revision 2.0, 80 MB/s)
- 4 x AMD Athlon MP 1900+ (1600 Mhz, 256 kB cache, 2 SMP systémy se 2 CPU), 2 GB RAM, 2 x 3Com Corporation 3c980-TX 10/100baseTX NIC, Netgear GA620 NIC, AMD-765 IDE Controller, IC35L040AVVA07-0 ATA DISK drive (2MB cache)
- Notebook Compaq Evo N1020v, Intel Celeron M 1,6 Mhz, 256 MB RAM, IDE disk HITACHI_DK23DA-30 (Ultra DMA 33), kombinovaná DVD-ROM/CD-RW IDE mechanika HL-DT-STCD-RW/DVD DRIVE GCC-4240N
- Wireless NIC Micronet 906b (Prism 2.5 chipset), Wireless NIC Avaya Silver PCMCIA
- přepínač Cisco Catalyst 1900
- CD-RW mechanika na USB Sony CRX10U
- PDA Compaq Ipaq 3970, CPU XScale 400 Mhz, 64 MB RAM, Dual Slot PC Card Expansion Pack

SW:

- OS Linux, distribuce RedHat 7.3 (Valhalla)
- OS Linux, distribuce Debian Woody
- Jádra verze 2.4.7-20.7 a 2.4.22
- hyperscsi verze 20030930

Instalace

Klient i server se distribuují současně v jednom balíku. Na stránkách projektu [5] jsou připraveny jak binární balíčky pro některé distribuce, tak balíček obsahující celý zdrojový kód, který není závislý na konkrétní distribuci ani jádru.

V případě kompilace ze zdrojového kódu je třeba mít i zdrojové kódy jádra již se závislostmi a povoleným SCSI subsystémem (může být i ve formě modulů). Pro použití IDE CD-R/RW mechanik je nutno zakompilovat i podporu IDE-SCSI. Také je důležité mít v adresářích `/usr/include/linux` a `/usr/include/asm` hlavičkové soubory jádra, se kterým se bude HSCSI překládat a používat. V souboru `Makefile` lze nastavit úroveň optimalizace výsledného překladu pro procesory, především rodiny *i686*. Dále stačí po rozbalení HSCSI balíku ve vzniklém adresáři zadat `make` a `make install` což je celá instalace. V distribuci Debian je ještě nutno přesunout symbolické odkazy `hs-server` a `hs-client` z adresáře `/etc/rc.d/init.d` do `/etc/init.d`.

Po instalaci jsou v systému tyto soubory:

Jaderné moduly:

```
/lib/modules/<verze_jádra>/kernel/drivers/scsi/hs-server.o  
/lib/modules/<verze_jádra>/kernel/drivers/scsi/hs-client.o
```

Konfigurační soubory:

```
/etc/hscsi/hs-server.conf  
/etc/hscsi/hs-client.conf
```

Instalační konfigurační soubor (obsahuje cesty k modulům a skriptům, je striktně vyžadován):

```
/etc/hscsi/install.config
```

Skripty pro použití serveru a klienta:

```
/sbin/hs-configure  
/sbin/hs-wrapper  
/sbin/hs-server  
/sbin/hs-client
```

Praktické zkušenosti

- Ověření celkové funkčnosti jak klienta, tak serveru. Jedno/více klientská (serverová) varianta. Testované verze byly na platformě Intel/Linux funkční, ale pouze na jádrech 2.4.18 a vyšších. Starší verze HSCSI podporují i starší jádra. Protože varianta nad IP vrstvou ještě nebyla implementována, byla pro testy použita Ethernetová varianta. Klient byl se serverem propojen přímo (pro výkonnostní testy na gigabitovém ethernetu) a přes přepínač (pro testy přístupu více klientů). Fungoval jak přístup více klientů na server, tak i konfigurace klienta, který měl připojen zařízení z více serverů.
- Ověření funkčnosti na bezdrátové síti standardu 802.11b
Testované verze sice byly schopny zajistit klientovi přístup na vzdálené zařízení přes bezdrátovou síť, ovšem při pokusech o přenos souvislého bloku dat docházelo k chybám, které vedly nejen k neúspěšnému přenosu, ale na straně klienta také způsobily nemožnost dále se zařízením pracovat.
- Ověření funkčnosti bezpečnostních vlastností (integrity i kryptování).
Bylo ověřeno fungování mechanismů kontroly integrity a kryptování přenosu na úrovni HSCSI protokolu. Implementace zahrnuje pouze jednu metodu jak pro kontrolu integrity (SHA1 HMAC), tak kryptování (128 bitový algoritmus AES). I přes varování autorů, že kryptování přenosu může vést k poškození dat, se toto nepodařilo prokázat. Bohužel kryptování zvýšilo zátěž strojů až na trojnásobek. Ve verzích starších než 20030930 se zajištění integrity při přenosu nezdařilo vůbec, pravděpodobně z důvodu špatné implementace a to i přesto, že tato vlastnost byla autory označována za zcela funkční. V poslední verzi (20030930) již kontrola integrity funguje zcela bez problémů, bez jakékoliv zmínky v *changelogu*.
- Fungování přístupu na SCSI disky.
Protokol HSCSI fungoval s SCSI disky bez nejmenších problémů.
- Fungování přístupu na IDE disky (jak pevné, tak optické).
IDE disk fungoval nativně, na rozdíl od IDE optické mechaniky, která se musela používat jako SCSI zařízení skrze emulační mezivrstvu IDE-SCSI jak na serveru, tak i klientovi (a to i pro CD-ROM mechaniku určenou pouze pro čtení). Fungoval i zápis na CD-RW mechaniku.
V této implementaci se vyskytuje problém u výměny média při již připojené mechanice, klient od serveru dostává neustále obsah předchozího média (platí pro všechny formy připojení CD mechanik).

- Fungování přístupu na USB zařízení.
Bylo testováno zařízení třídy USB-Storage (konkrétně CD-RW mechanika Sony). Opět bylo nutné použít IDE-SCSI mezivrstvy na obou stranách (klient/server). Fungoval i zápis na RW médium. Dále bylo samozřejmě nutno zakompilovat do jádra podporu USB Mass Storage zařízení (včetně podpory této konkrétní mechaniky, resp. jejího řadiče).
- Fungování přístupu na SCSI pásku.
Bylo testováno zálohování a obnova ze vzdálené SCSI pásky. Záloha i obnova fungovala bez chyb, ovšem při pokusu o smazání zhruba 40 GB pásky (což byla časově náročná operace) se vyskytla chyba s frontou SCSI příkazů, která vedla až k zhroucení klienta. Bylo třeba restartovat celý OS. Server se dal ukončit a spustit znovu bez restartování.
- Fungování klienta v prostředí Microsoft Windows 2000 Professional
Byla testována funkčnost prvního klienta pro jiný operační systém než Linux. Klient má podstatně omezenější možnosti, než Linuxový ekvivalent, zcela mu chybí možnost kontroly integrity i kryptování přenosu, lze se připojit pouze na jeden server, trpí nekorektním ukončením spojení se serverem (pouze u pevného disku, CD-ROM funguje korektně). Celkově je vidět, že se jedná o první verzi, která zdaleka není tak daleko jako primární Linuxová.
- Ověření fungování klienta na mobilním zařízení
Bohužel se nepodařilo na mobilním zařízení (PDA) přeložit jádro s podporou HSCSI a tedy tato možnost nebyla ověřena. Navíc připojení mělo být realizováno pomocí bezdrátové sítě, která se ukázala jako nevhodná.

Výkonnostní testy

Zatímco ověřování funkčnosti bylo prováděno kombinovaně na všem výše uvedeném HW, výkonnostní testy byly prováděny pouze na SMP serverech s gigabitovými ethernet adaptéry. Propojení bylo přímé optickým vláknem.

Použit byl program na měření výkonnosti diskových operací IOzone [6]. Kvůli 2 GB RAM, byl použit vzorek o velikosti 4 GB. Testovaly se velikosti přenášených bloků v rozpětí 4 kB až 16 MB. Byla použita funkce *close()* a výstup byl zapsán do binárního formátu MS Excel.

Parametry pro IOzone:

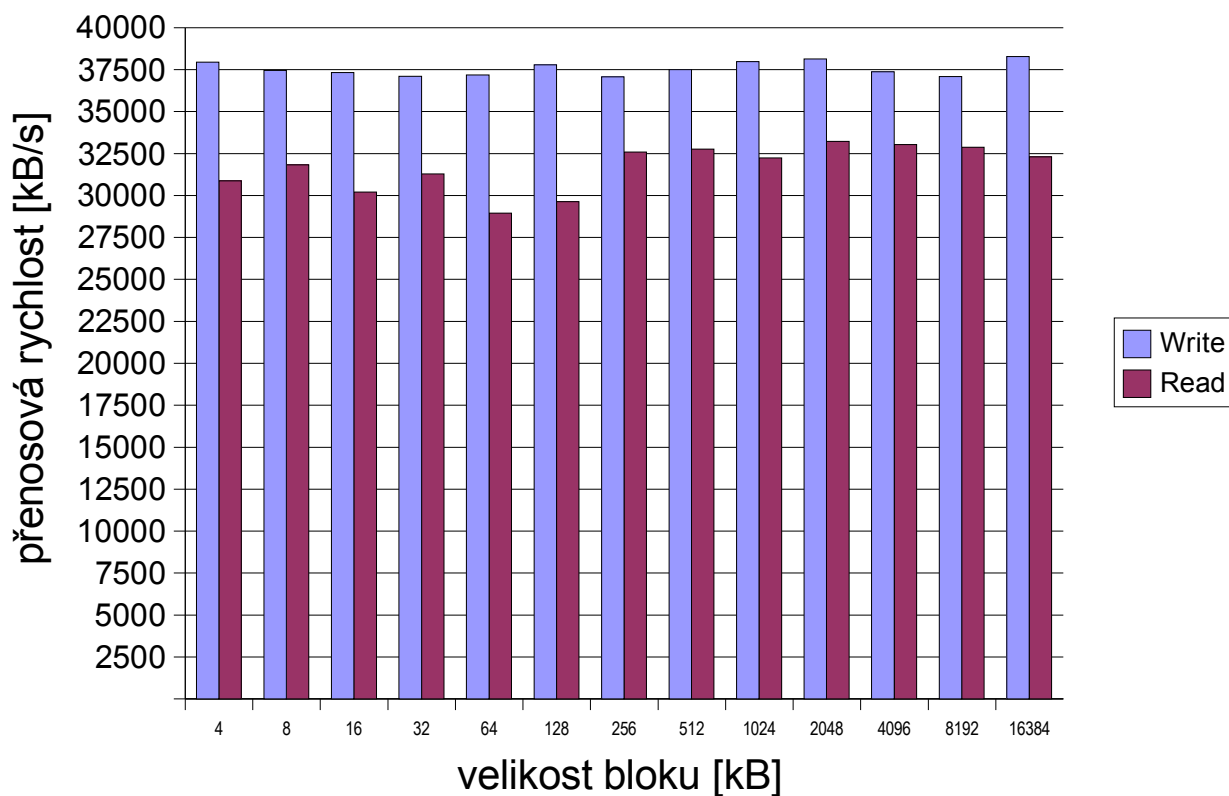
```
iozone -Rb hscsi.wks -n 4g -g 4g -z -c -a -i 0 -i 1
```

Měření byla provedena několikrát a výsledky zprůměrovány, podmínky počas jednotlivých měření se nijak neměnily. Předmětem testů bylo zjistit opravdový výkon s použitím HSCSI v různých konfiguracích (*SMP*, *nonSMP*) a hlavně výkonnostní dopad možností kryptování a zajištění integrity.

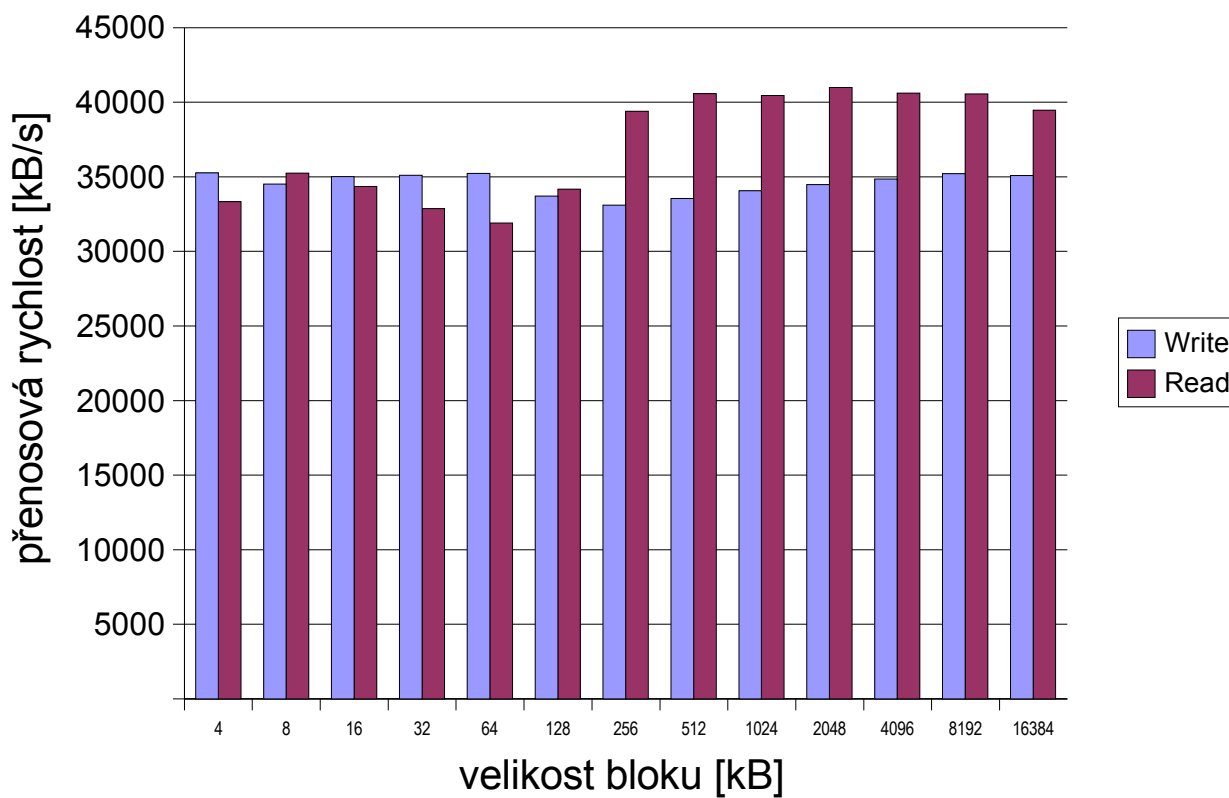
Komentáře k jednotlivým testům:

1. jádro 2.4.22, nastavení velikosti datového okna pro gigabit ethernet, volba pro server
PKT_WINDOW_SIZE: 32 , pro klienta SG_TABLE_SIZE: 32
2. jádro 2.4.22, ponechání velikosti datového okna pro 100 Mbit ethernet, PKT_WINDOW_SIZE: 5,
SG_TABLE_SIZE: 16
3. jádro 2.4.22, gigabit ethernet, kontrola integrity, volba pro server VOL_OPT: 1403:0, pro klienta
VOL_OPT: 1403
4. jádro 2.4.22, gigabit ethernet, kryptování přenosu, volba pro server VOL_OPT: 1602:0, pro klienta
VOL_OPT: 1602
5. jádro 2.4.20-20.7 (Redhat), gigabit ethernet
6. jádro 2.4.20-20.7 (Redhat), gigabit ethernet, kryptování přenosu, volba pro server
VOL_OPT: 1602:0, pro klienta VOL_OPT: 1602
7. jádro 2.4.22 *SMP*, gigabit ethernet, zvýšení počtu vláken pro zápis a čtení, volby pro server i
klienta MULTI_RCV_THREAD: 4, MULTI_XMIT_THREAD: 6

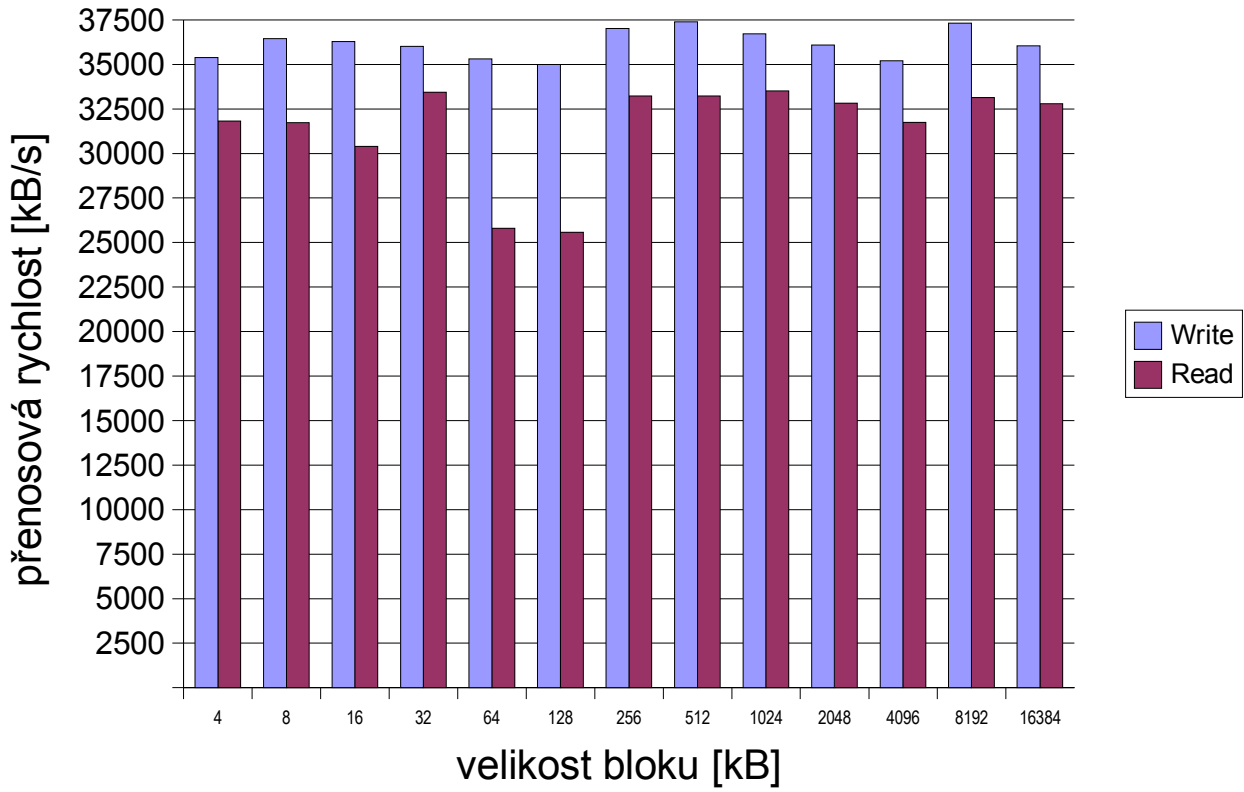
HyperSCSI - 1



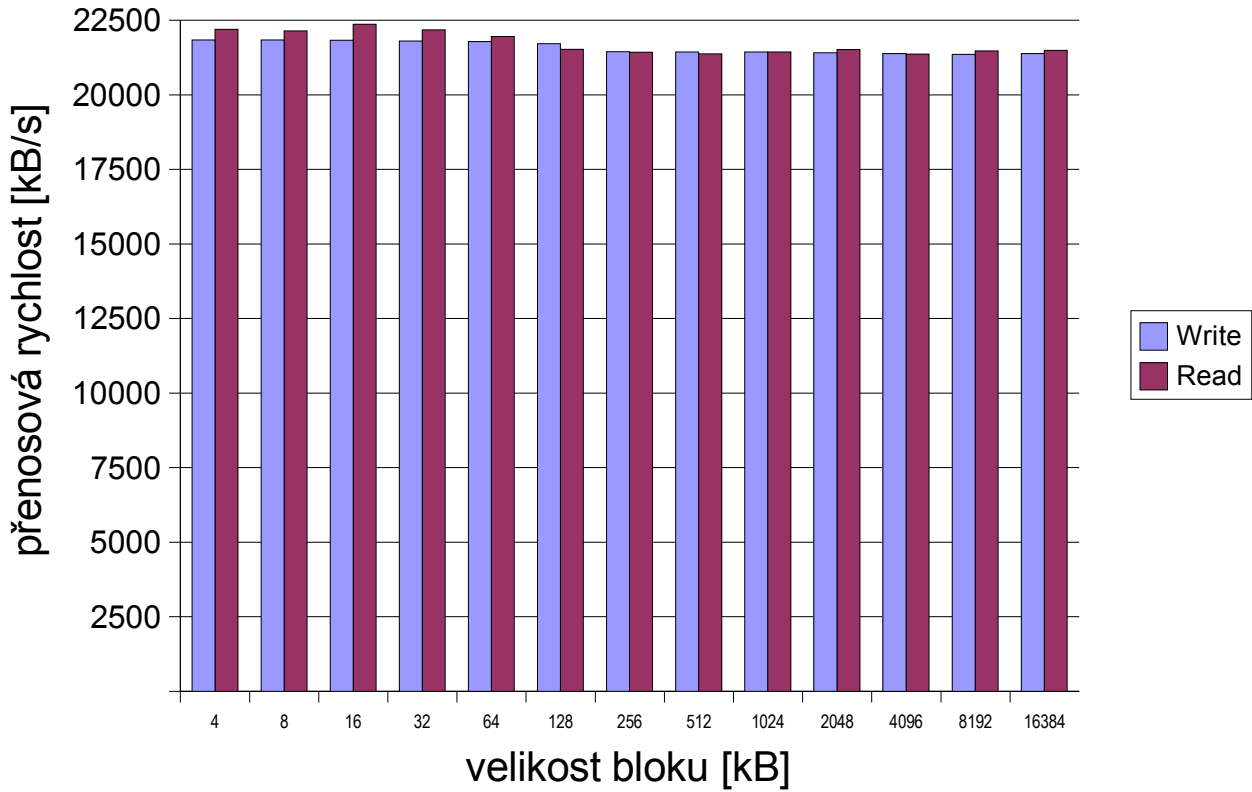
HyperSCSI - 2



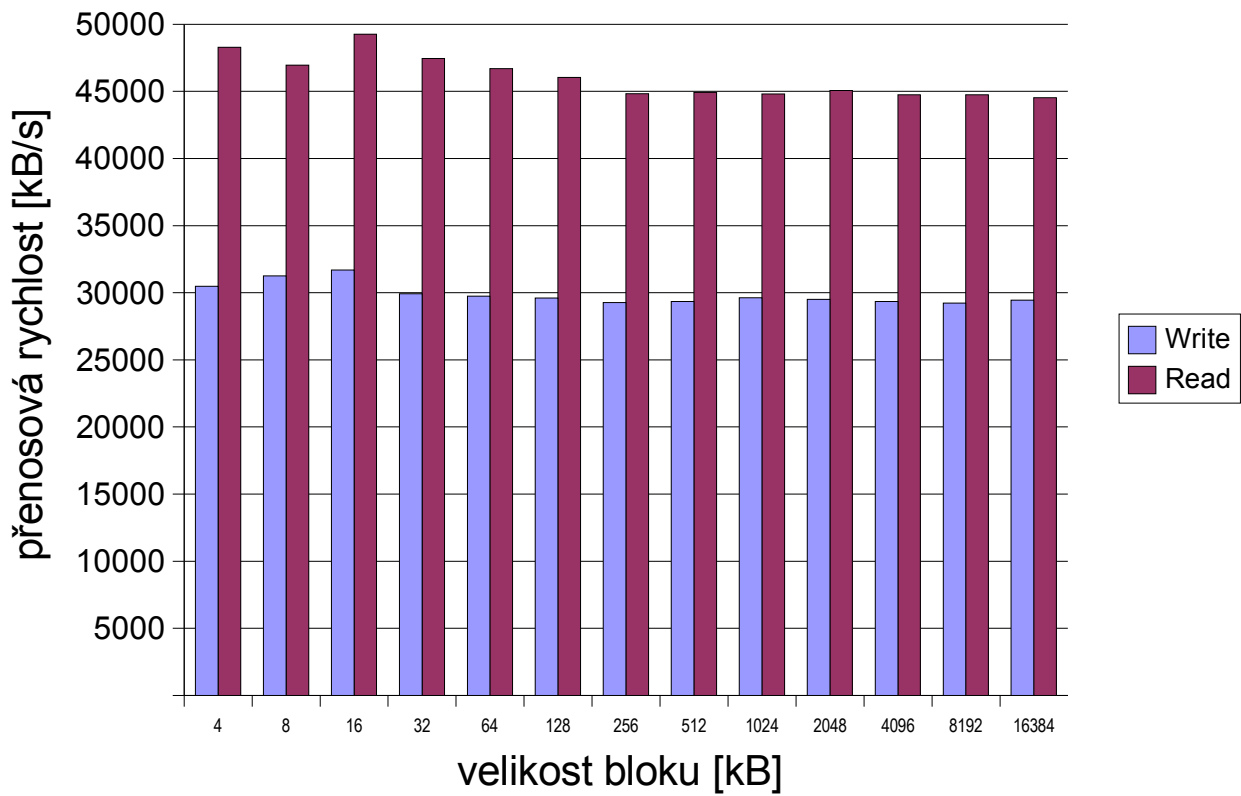
HyperSCSI - 3



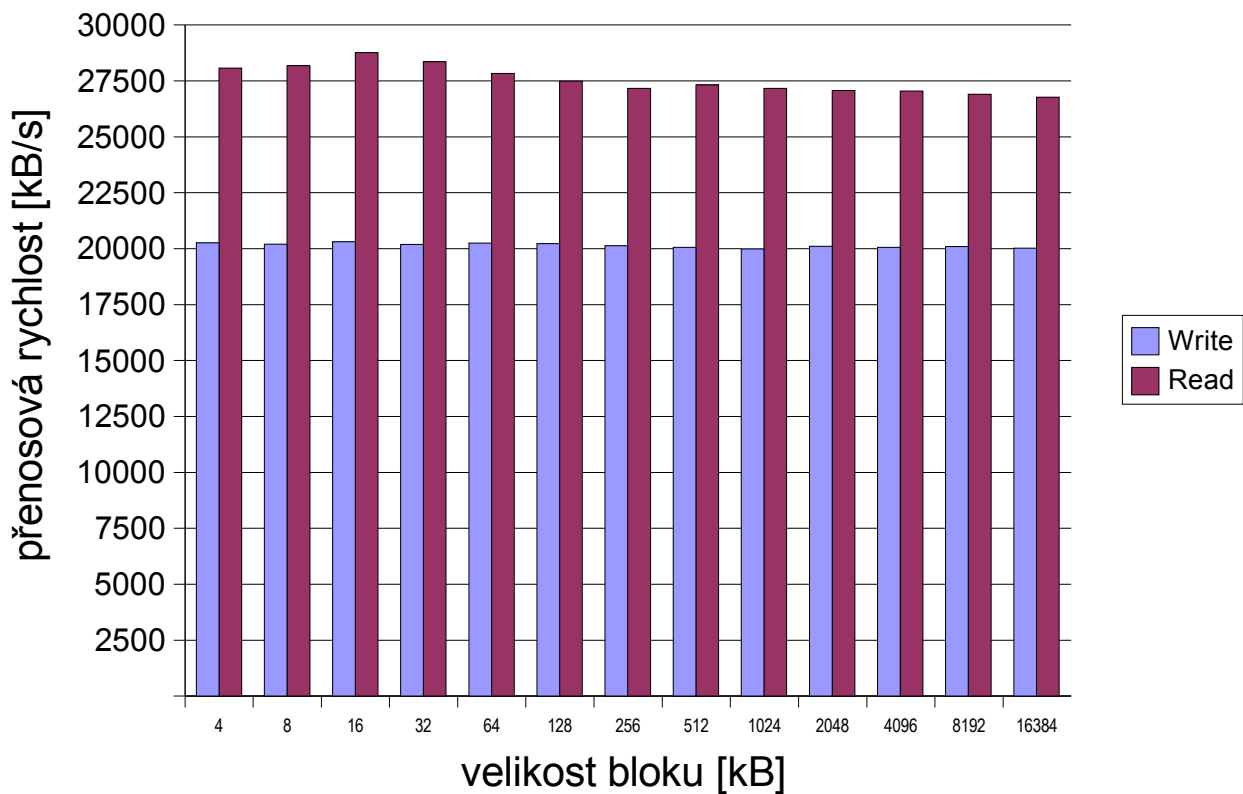
HyperSCSI - 4



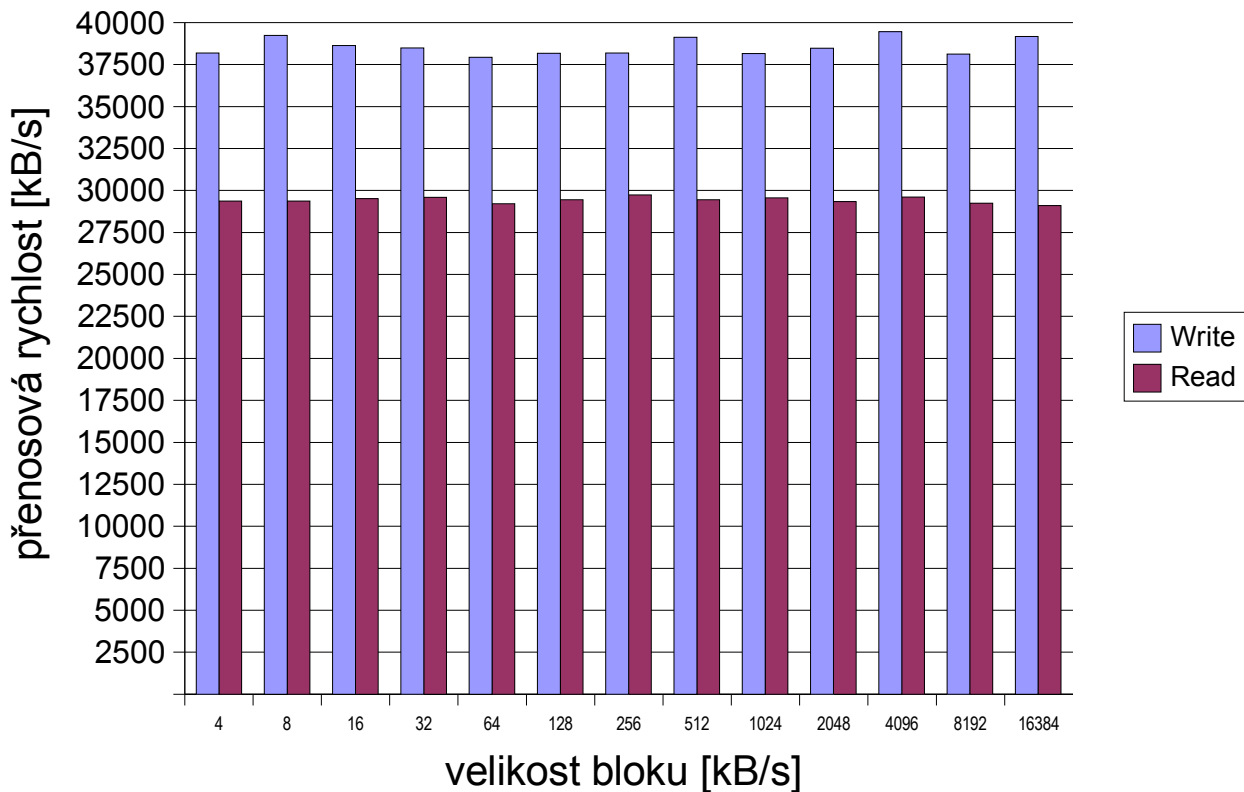
HyperSCSI - 5



HyperSCSI - 6



HyperSCSI - 7



Z grafů je vidět, že zatímco kontrola integrity přenášených dat nemá na propustnost nějak výrazný vliv (grafy 1 a 3), tak kryptování přenášených dat již znatelně výkon ovlivňuje (grafy 1 a 4). Dále je zajímavé, že na gigabitovém ethernetu s ponecháním velikostí příjmajícího a odesílajícího okna s hodnotami pro 100Mbit ethernet (tedy okna menší velikosti) dochází u středních (256 kB) a větších bloků dat k vyšší propustnosti (grafy 1 a 2). Potvrzení faktu zjištěného při měření iSCSI, že optimalizovaná jádra firmy Redhat mají lepší výsledky než jádra standardní, rovněž stojí za pozornost (grafy 1 a 5). Výsledky porovnání *SMP* a *nonSMP* verzí jader je tak trochu neočekávané, protože lehké zvýšení propustnosti při zápisu je spojeno s poklesem propustnosti čtení (grafy 1 a 7). Nicméně u *SMP* jádra byla vidět menší zátěž CPU (zhruba o 20 %).

Závěry

Bohužel nelze jednoznačně porovnat iSCSI a HyperSCSI protokol, protože zatímco iSCSI server je dostatečně implementován pouze jako specializovaný HW, HyperSCSI server pouze jako software. Nicméně, lze vyslovit následující závěry:

1. Protokol HyperSCSI over Ethernet se již nyní jeví ve své linuxové verzi jako stabilní a použitelný na vytvoření lokálního SAN-u. Jeho hlavní nevýhodou ovšem nadále zůstává právě ono omezení na Ethernet a tedy celkovou rozlehlost případného řešení. Naopak proti řešením postaveným na iSCSI či FC může být velice příznivá cena za zřízení. Proti iSCSI navíc nabízí vyšší výkon při menší zátěži. Mohl by tedy být vhodným doplňkem menších, na linuxu postavených výpočetních *clusterů*.
2. Bezpečnost přenosu je sice dobře navržena i implementována, ovšem za cenu podstatně vyšších nároků na hardware (alespoň v momentálně implementované variantě s kryptováním AES). Kryptování přenosu bude mít větší opodstatnění u IP verze.

Literatura a odkazy

- [1] <http://www.cesnet.cz/doc/techzpravy/2002/iscsi/iscsi.pdf>
- [2] <http://www.cesnet.cz/doc/techzpravy/2002/ipstorage/ipstorage.pdf>
- [3] <http://www.cesnet.cz/doc/techzpravy/2002/ipstorage3/ipstorage3.pdf>
- [3] <http://nst.dsi.a-star.edu.sg/mcsa/hyperscsi/nspd2.pdf>
- [4] <http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt>
- [5] <http://nst.dsi.a-star.edu.sg/mcsa/hyperscsi/download.html>
- [6] <http://www.iozone.org>

Přílohy – příklad konfigurace

```
[HYPERSCSI-SERVER-CONFIG-VERSION-20030930]
# Sample Config File for HyperSCSI Server - Modify Before Use!
# Optimised for Fast Ethernet
# Last Updated - 03 July 2003
```

```
[ADD]
```

```
[MODULE_DEF]
# For GE, try PKT_WINDOW_SIZE 32
PKT_WINDOW_SIZE: 32
##PKT_WINDOW_SIZE:5
MULTI_RCV_THREAD: 2
MULTI_XMIT_THREAD:3
REXMIT_COUNT: 2
DIRECT_MC: 0

[VOL_DEF]
# Example of supported device type
# VOL_1: SDA SCSI-DISK 0 0 0 0
# VOL_1: HDA IDE-DISK 0 0 0 0
# VOL_1: SCD0 SCSI-CDROM 0 0 0 0
# VOL_1: SCD0 IDE-CDROM 0 0 0 0
# VOL_1 ST0 SCSI-TAPE 0 0 0 0
# VOL_1: SDA USB-DISK 0 0 0 0
# VOL_1: MD0 RAID 0 0 0 0
# VOL_1: LVMA LVM 0 0 0 0
# VOL_1: SGO GENERIC 0 0 0 0
#
# For DEVFS, simply define VOL_1 SDA is enough
# VOL_1: SDA
###VOL_1: SDA SCSI-DISK 0 0 0 0
VOL_1: HDA IDE-DISK 0 0 0 0
```

```
[NETWORK_DEF]
LAN_1: ETH2
```

```
[GROUP_DEF]
GROUP_NAME: CESNET
PASSWORD: HESLO
NET: LAN_1
IP_ON: 0
VOL_NAME: VOL_1
VOL_OPT: 0:0
```

```
[END]
```

```
[HYPERSCSI-CLIENT-CONFIG-VERSION-20030930]
# Sample Config File for HyperSCSI Client - Modify Before Use!
# Optimised for Fast Ethernet
# Last Updated - 02 July 2003
```

```
[ADD]
```

```
[MODULE_DEF]
# For GE, try SG_TABLE_SIZE 32
SG_TABLE_SIZE: 32
###SG_TABLE_SIZE: 16
MULTI_RCV_THREAD: 2
MULTI_XMIT_THREAD: 3
REXMIT_COUNT: 2
DIRECT_MC: 0
```

```
[NETWORK_DEF]
LAN_1: ETH2
```

```
[GROUP_DEF]
GROUP_NAME: CESNET
PASSWORD: HESLO
NET: LAN_1
TARGET_IP: 192.168.2.16
VOL_ID: 0
VOL_OPT: 0
```

```
[END]
```