

Storage Over IP

Jan Haluza, Filip Staněk, Ivan Doležal

Technická zpráva CESNETu č. 5/2002

18. června 2002

Úvod

Technologie iSCSI je popsána jako Internet Draft viz [IETF]. Jak již název napovídá, technologie umožňuje dálkové propojení storage a výpočetních zařízení na úrovni SCSI protokolu internetem. Stejně jako u klasického SCSI je vysílatel příkazů označován jako initiator, jejich vykonavatel jako target. Síťová entita označovaná jako klient sestává z jednoho nebo více initiatorů a síťových rozhraní, server z jednoho nebo více targetů a síťových rozhraní. Každý uzel má světově unikátní označení. Obě strany spolu komunikují protokolem TCP, server naslouchá dle [IETF] na portu 5003.

Technologie má potenciál být využita v supervýpočetních clusterech CESNETu v Ostravě, Brnu, Praze, Plzni; pro FTP servery, video servery atd. zejména ve vysokorychlostní síti CESNET2 a v prostředí MAN.

iSCSI, platforma PC, Linux

Tato část zprávy se zaměří na vytvoření obou částí iSCSI na platformě PC v prostředí Linuxu. Poznamenejme pro úplnost, že initiator může být libovolné zařízení, předpokládá se však výpočetní systém. Naproti tomu target může být nejen obecný počítač, ale i vysoce specializované HW zařízení, např. Cisco SN5420 iSCSI router nebo diskové pole IBM TotalStorage IP Storage 200i.

Níže popsané příklady a pokusy byly prováděny na následujících sestavách:

HW:

Pentium III 500 MHz with 512 KB cache, 384 MB RAM, NetGear GA620 Gigabit Ethernet, SCSI controller Adaptec AHA-2940U2/W, hard drive Seagate ST136403LW (Type: Direct-Access, ANSI SCSI revision: 02, 80.000MB/s transfers)

SW:

sestava dále označovaná jako rave: Debian 3.0 (Woody), kernel 2.4.19-pre1-ac1

sestava dále označovaná jako termit7: Red Hat 7.2, kernel 2.4.18

sestava dále označovaná jako termit6: Debian 2.2 (Potato), kernel 2.4.19-pre1-ac1

Dostupná řešení

V době přípravy tohoto textu byly dostupné výsledky práce následujících projektů:

Intel iSCSI Reference Implementation Version 8

Pracuje nad Linux kernelem 2.4, po úpravách i nad 2.2. Podle vyjádření autorů moduly neběží spolehlivě v SMP módu (nebylo testováno). Je dostupná ve formě zdrojových textů na [Intel]. Nejedná se o finální produkt, firmou je označen jako studie a umístěn pod BSD licencí mimo její oficiální servery (viz URL v citaci). Implementace iniciatoru i targetu je obsažena v jednom distribuovaném balíku. Konfiguraci je potřeba provést úpravou zdrojových textů již v době překladu. Postup je částečně zdokumentován v příloženém souboru README, nicméně nyní shrneme zásadní kroky:

Na obou stranách

v souboru **iscsi.h**:

- Pokud je požadováno, aby aplikace pracovala na portu specifikovaném [IETF], je potřeba upravit tento řádek (bez úpravy používá toto řešení nestandardně port 5000).

```
#define ISCSI_PORT 5003
```

Překlad se vždy provede standardním `make`. Je potřeba mít k dispozici zdrojové texty jádra.

Na straně *initiator*

v souboru **initiator.h**:

- Je potřeba definovat počet targetů. Tato hodnota udává počet položek v souboru `initiator.c` (viz níže):

```
#define CONFIG_INITIATOR_NUM_TARGETS 1
```

v souboru **initiator.c**:

- Zde je potřeba zadat adresy a porty targetů, případně ISCSI jména targetů (nejsou-li zadána, budou objevenána). Zbylé parametry jsou nevyužity:

```
static INITIATOR_TARGET_T g_target[CONFIG_INITIATOR_NUM_TARGETS] = {
    {"195.113.113.25", ISCSI_PORT, "", NULL, 0};
};
```

Odstartování iniciatoru se provede dynamickým zavedením modulu `intel_iscsi.o`. Po odstartování se iniciator pokusí vytvořit virtuální SCSI řadič, spojit se s targety (podle zakompilovaných informací) a obsadit nejnižší volné SCSI zařízení.

Na straně *target*

v souboru **target.h**:

- Zadává se počet LUN, který odpovídá počtu symbolických unixových odkazů (`ln`) vytvořených v adresáři `/tmp` na bloková zařízení, která chceme sdílet (jak popsáno níže).

```
#define CONFIG_TARGET_NUM_LUNS 1
```

- Dále je potřeba zadat počet bloků těchto zařízení. Tento údaj je skalár, nikoliv pole. Všechna zařízení proto musejí mít stejnou velikost. V našem případě byla velikost nastavena dle vztahu

4.427 cylindrů * 16.065 512-bajtových jednotek na cylindr = 71.119.755 bloků (geometrie rozeznána programem fdisk používaným Linux distribucemi) Délka bloku může být nastavena na hodnoty 512, 1024 nebo 2048.

```
#define CONFIG_TARGET_NUM_BLOCKS 71119755
#define CONFIG_TARGET_BLOCK_LEN 512
```

- Zadání iSCSI jména (zde shodou okolností použito stejné jméno jako jméno počítače):

```
#define CONFIG_TARGET_NAME "termit7.vsb.cz"
```

Zpřístupnění jednotlivých SCSI zařízení je nyní doporučeno zajistit vytvořením symbolických odkazů v adresáři /tmp na sdílená zařízení, například

```
/tmp/iscsi_termit7.vsb.cz_lun_0 -> /dev/sdb
```

Nejsou-li tyto odkazy vytvořeny ručně, zobrazí se po odstartování targetu v adresáři /tmp soubory o velikosti aktuálně připojených SCSI disků.

Odstartování targetu se provede příkazem `ufsdisk_mmap` (jak bylo pro snížení latence [Bench01] prováděno při popsání pokusech), případně s vyloučením memory mappingu příkazem `ufsdisk`. Tento binární soubor obsahuje vše potřebné pro běh targetu. I zde je možné dodatečně specifikovat fyzické parametry disku, číslo portu a iSCSI jméno targetu.

Ukončení targetu se provede stisknutím Ctrl-C. Target odmítne provést ukončení, pokud je na něj připojen alespoň jeden initiator.

Při praktickém ověřování byly zjištěny tyto skutečnosti:

- Buď probíhá vývoj referenční implementace s překladačem pro 64bitovou platformu nebo pouze s historickými SCSI zařízeními. Důsledkem je zásadní omezení velikosti pevného disku na velikost proměnné `int`. Protože velikost použitého disku zjištěná programem fdisk byla přesně 36 413 314 560 B (což není celý násobek mocniny dvou), wraparoundem s rozsahem velikosti proměnné `int` vznikl adresovatelný prostor na disku o velikosti cca 1 958 MB. Úprava kódu tak, aby pracoval s proměnnými `long` místo `int`, však nebyla triviální. Překročení této hranice způsobovalo zhroucení filesystému. Bylo rozhodnuto pokračovat v pokusech s vědomím tohoto omezení a uvedenou hranici nepřekračovat.
- Jakmile velikost disku připojeného k target zařízení překročí hranici 2 097 152 cylindrů, začne Intel předávat iniciatoru počet cylindrů vydělený hodnotou (255*63).
- Ačkoliv je v dokumentaci k Intel targetu jedna část věnována zajištění interoperability s Cisco iniciátorem, popsání úpravy neúčinkují a je potřeba se pokusit o úpravy na straně Cisco iniciatoru (viz dále).
- Zápis na target se neprojevuje okamžitě. Pokud initiator přikáže zápis, data se na disku objeví až po ukončení iniciatoru i targetu. Do té doby data nejsou vůbec viditelná.
- Současné čtení dvou iniciatorů z dvou partition na téže disku dokončil správně pouze dříve vzešlý požadavek, operace pro pozdější požadavek skončila chybou. Současný zápis do různých partition proběhl v pořádku. Současný zápis do téže partition sice formálně neskončil chybou, data ale nebyla zapsána. Protože operace dvou iniciatorů nad zařízením s jedním LUN selhávaly, byla

provedena další série pokusů s přístupem k těmto zařízením, kterému však byla přidělena dvě různá LUN:

```
#define CONFIG_TARGET_NUM_LUNS      2

/tmp/iscsi_termit7.vsb.cz_lun_0 -> /dev/sdb
/tmp/iscsi_termit7.vsb.cz_lun_1 -> /dev/sdb
```

Při pokusech se dvěma různými initiatory, kdy každý z nich využíval jiný LUN pro totéž zařízení, již probíhalo čtení i zápis v pořádku.

Tato implementace je nejfunkčnější ze všech zkoumaných. Další testy proto byly provedeny právě na ní.

linux-iscsi (Cisco)

Jde pouze o implementaci iniciatora, proto se Cisco implementaci budeme věnovat pouze stručně:

Její zdrojové texty lze získat nejen na domovských stránkách projektu [Cisco], ale je i součástí kernelu distribuce Red Hat Linux od verze 7.2, kde je připraven též instalační balíček.

Správný běh je zajišťován pomocí démonu `iscsid`. Ten je řízen skriptem `/etc/init.d/iscsi`. Démon zavádí a uvolňuje z jádra potřebné moduly, konfiguruje je dle `/etc/iscsi.conf` a zajišťuje un-mountování disků podle `/etc/fstab.iscsi`. Unikátní jméno iniciatoru je uloženo v souboru `/etc/initiatorname.iscsi`.

Praktické zkušenosti:

- Implementaci Cisco nelze použít pro spolupráci s targetem Intel: iniciator chybně pracuje s geometrií disku. I když se úpravou zdrojového textu Cisco iniciatoru podařilo výše zmíněný efekt Intel optimalizace odstranit, Intel target přesto při komunikaci indikoval chyby. Ty byly způsobeny i faktem, že Cisco iniciator vysílal i nestandardní požadavky (např. textové řetězce o výrobci software).

Bylo by zajímavé prozkoumat tuto implementaci s firemním HW řešením, to však není v současné době autorům k dispozici.

UNH iSCSI Initiator and Target Reference Implementation ref20.2

Implementace byla jejími tvůrci testována na Linux kernelu 2.4.15. Příložená podrobná dokumentace objasní, že vzniknuvší moduly `iscsi_initiator.o`, `scsi_target.o` a `iscsi_target.o` je potřeba dynamicky zavést do jádra. Toto zavedení i následné konfigurování příloženými utilitami `iscsi_config` a `iscsi_manage` provádějí příložené shellové skripty, které je ale potřeba upravit. Obdobnou službu provedou i skripty pro odinstalování.

Naše skripty vypadaly např. takto:

iscsi_target_install

```
insmod scsi_target.o

insmod iscsi_target.o

../common/iscsi_manage target force v=1100

../common/iscsi_manage target setp TargetName="termit7.vsb.cz"
```

iscsi_initiator_install

```
/sbin/insmod -f iscsi_initiator.o

../common/iscsi_manage init set InitiatorName="rave.vsb.cz" host=1

../common/iscsi_manage init set TargetName="termit7.vsb.cz" host=1

./iscsi_config up host=1 ip=195.113.113.25 port=5003
```

Detailní popis zde nebudeme uvádět, zmíníme se pouze o některých zajímavých parametrech utility `iscsi_manage`: „v“ udává verzi internet draftu, podle které se obě strany mají chovat (800, 900, 1100 pro verze 8, 9, 11). Zavádějící je parametr „host“, který udává číslo, jaké má v Linux SCSI subsystému adaptér „iSCSI initiator“ (viz také [TLDP]). Protože je tento virtuální adaptér zaváděn jako poslední, mají všechny fyzické adaptéry nižší číslo.

Praktické zkušenosti:

- V době psaní tohoto příspěvku byl kód dostupný na [UNH] v podstatě nefunkční. Po navázání spojení dochází k zamrznutí jádra operačního systému na target serveru.

Ačkoliv byl tento kód nefunkční, bude zajímavé se k němu po čase vrátit.

Linux isns for iscsi

Projekt [linuxisns] neimplementuje použitelným způsobem ani initiator, ani target, pouze jmenné služby pro iSCSI (obdoba DNS). Je zde zmíněn pouze pro úplnost.

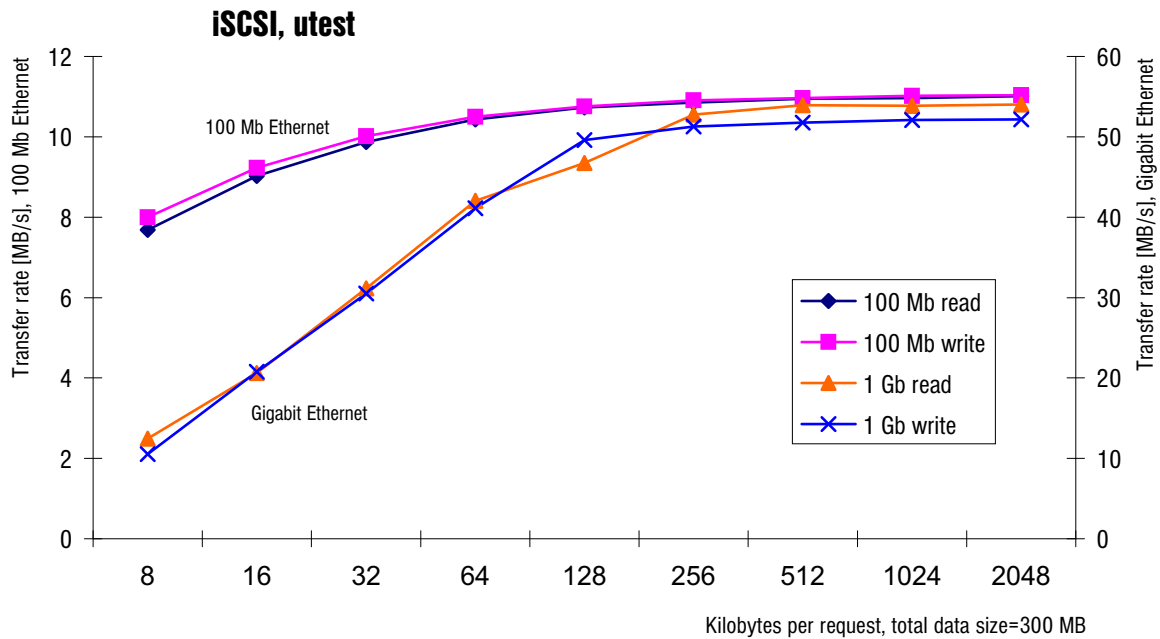
Výkon

Měření výkonu byla prováděna pouze s implementací Intel jako s jedinou funkční implementací.

Jako initiator byl použit počítač rave, targetem byl systém termit7, pro experimenty byl dále použit jako druhý initiator termit6. Souborový systém byl ext2.

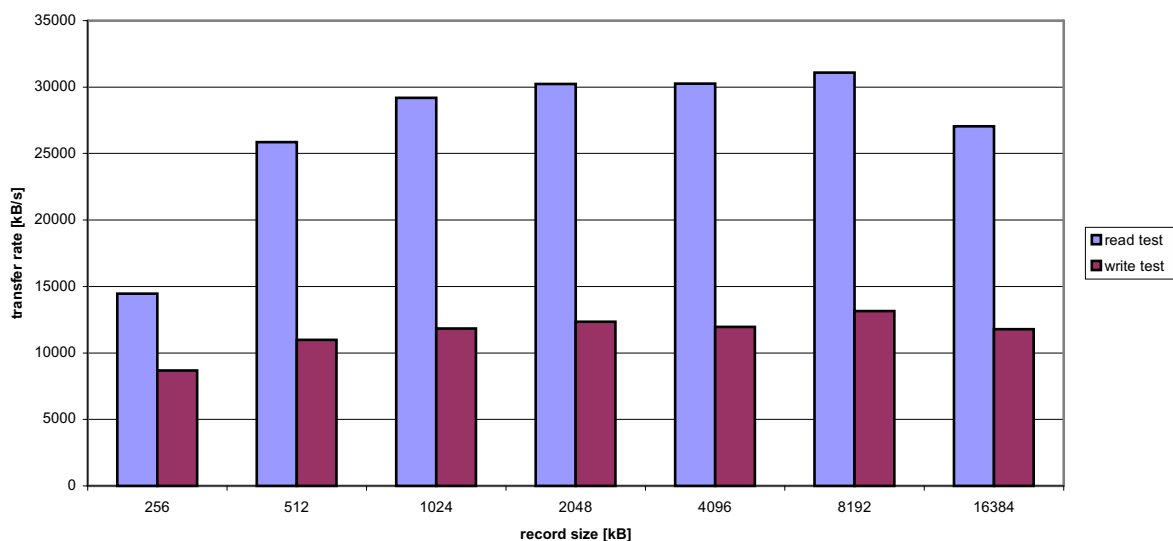
Pro měření výkonu iSCSI byla použita jednak utilita `utest` dodaná spolu se zdrojovými texty Intel implementace, jednak standardní benchmark IOzone [iozone].

utest nejprve vytvoří v paměti zadaný objem náhodných dat, který pak přeneše z iniciatora na target. Součástí tohoto programu je i zavedení modulu iniciatora do jádra. Programem lze změřit zpoždění odezvy targetu a přenosovou rychlost při různých velikostech požadavků na target; první možnost nebyla zkoumána. Měření bylo pro zajímavost provedeno na 100 Mb Ethernetu a gigabitové síti. Průchodnost sítě byla změřena v obou případech programem netperf pro velikost požadavku 16 kB a je v grafu vyznačena čárkovaně. Zatížení targetu dosahovalo během měření na gigabitové síti ve špičkách až 70%.



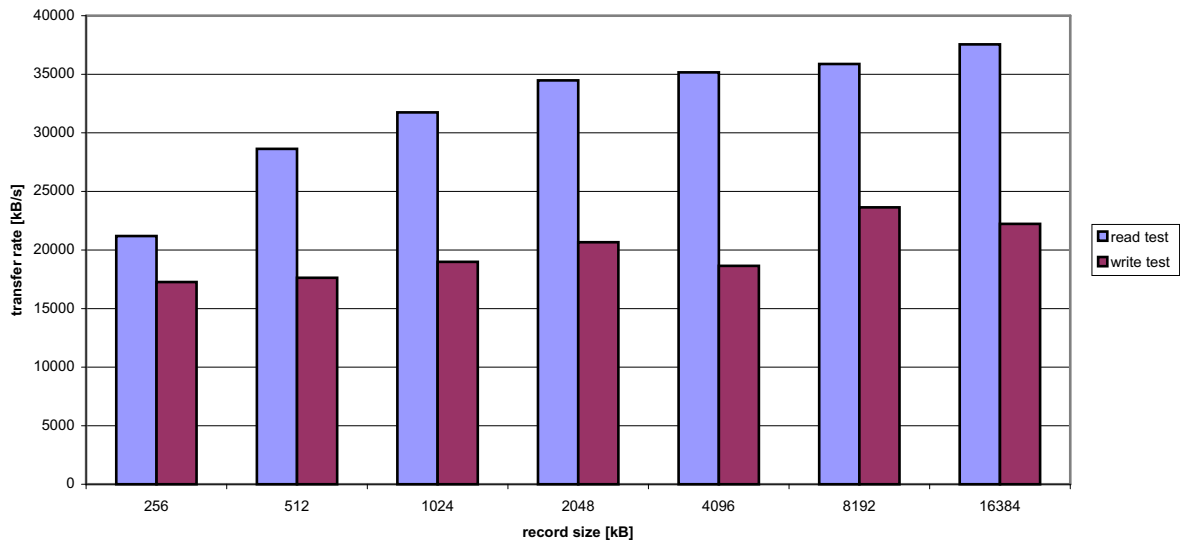
Další testy však byly provedeny pomocí standardního benchmarku IOzone. Též celkový objem přenášených dat se změnil a to z 300 MB na 1 GB.

iSCSI random read / write test (1GB file), IOzone



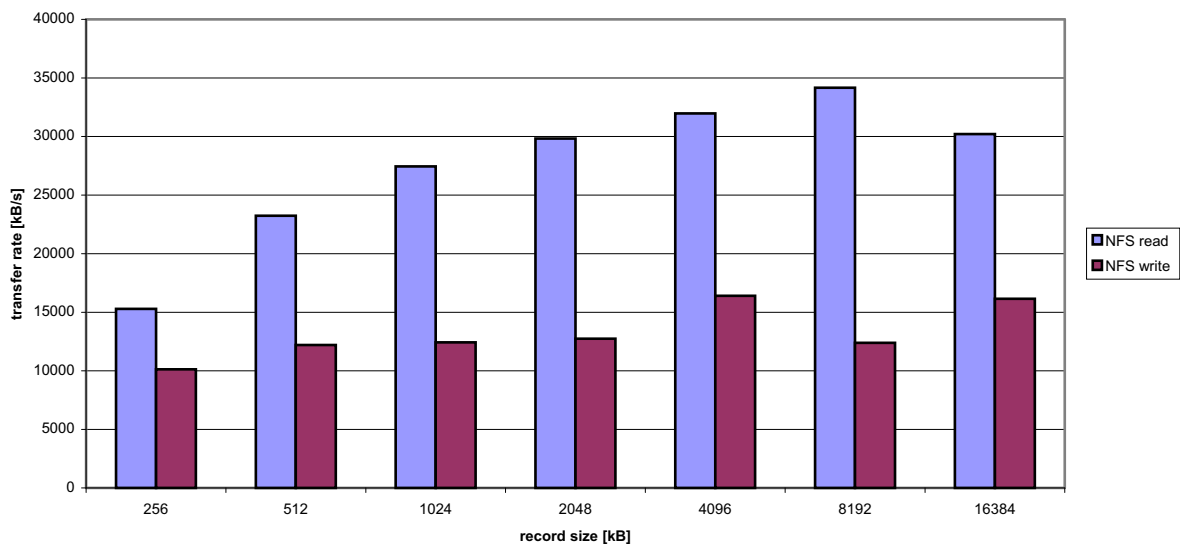
V této souvislosti bylo zajímavé porovnat přenosové rychlosti, kterých je dosahováno na hardware targetu v případě lokálních operací.

Local disk random read/write test (1GB file), IOzone



Na stejném HW bylo rovněž provedeno měření výkonnosti NFS (jiné strategie pro práci s daty na vzdálené storage):

NFS random read / write test (1GB file), IOzone



Výsledky všech měření shrnuje následující tabulka. Levá polovina obsahuje absolutní naměřené hodnoty (vždy v kB/s; velikost záznamu je udána v kB), pravá dává do poměru rychlost dosaženou při iSCSI nebo NFS provádění čtení/zápisu ku lokálnímu čtení/zápisu.

record size	256	512	1024	2048	4096	8192	16384		256	512	1024	2048	4096	8192	16384
iSCSI read	14473	25849	29185	30233	30254	31080	27051	iSCSI/L	68%	90%	92%	88%	86%	87%	72%
local read	21193	28629	31751	34477	35170	35888	37550								
NFS read	15291	23236	27448	29824	31973	34167	30208	NFS/L	72%	81%	86%	87%	91%	95%	80%
iSCSI write	8665	10975	11833	12344	11952	13141	11799	iSCSI/L	50%	62%	62%	60%	64%	56%	53%
local write	17269	17633	18981	20660	18642	23639	22230								
NFS write	10124	12203	12425	12742	16412	12390	16152	NFS/L	59%	69%	65%	62%	88%	52%	73%

Závěry

1. V době příprav této zprávy byla autorům práce známa tři čistě softwarová částečná či úplná řešení iSCSI na platformě PC, GNU/Linux. Ani jedno z nich se prozatím neukázalo jako vhodné pro rutinní použití. Tato oblast je nicméně dosud teprve ve vývoji.
2. Jak je patrné z výše uvedené tabulky, při práci s pevným diskem prostřednictvím iSCSI dochází u velmi malých požadavků k výrazné degradaci; s požadavky mezi 0,5 a 1 kB si však radí až o 9 procent lépe nežli NFS.
3. NFS naopak vykazuje lepší přenosové rychlosti u požadavků na velké objemy dat.

Literatura

[IETF] „iSCSI, Internet Draft“, <<http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-12.txt>>, April 17, 2002

[Intel] „Intel iSCSI Reference Implementation Version 8“, <<http://sourceforge.net/projects/intel-iscsi/>>, December 2001

[Bench01] „mmap Latency vs. Kernel Version“, <<http://cs.nmu.edu/~benchmark/index.php?page=mmap>>, Linux Benchmark Group, 2001

[Cisco] „linux-iscsi“, <<http://linux-iscsi.sourceforge.net/>>, May 2002

[UNH] „InterOperability Lab iSCSI Consortium“, <<http://www.iol.unh.edu/consortiums/iscsi/>>, May 15 2002

[TLDP] „The Linux 2.4 SCSI subsystem HOWTO“, <<http://www.tldp.org/HOWTO/SCSI-2.4-HOWTO/scsiaddr.html>>, The Linux Documentation Project, June 2002

[linuxisns] „Linux isns for iscsi“, <<http://sourceforge.net/projects/linuxisns/>>, June 2002

[iozone] „IOzone Filesystem Benchmark“, <<http://www.iozone.org/>>, June 2002