

# High Availability a Linux

Věroš Kaplan  
<veros-kaplan@koren.cz>

prosinec 2003

# Obsah

<b>1</b>	<b>Bylo, nebylo...</b>	<b>3</b>
1.1	Co to je za ptákovinu? . . . . .	3
<b>2</b>	<b>Cluster a HA</b>	<b>5</b>
2.1	heartbeat? heartbeat! . . . . .	6
2.2	Návrh . . . . .	6
2.3	Instalace . . . . .	7
2.4	Konfigurace . . . . .	7
2.4.1	/etc/ha.d/authkeys . . . . .	7
2.4.2	/etc/ha.d/haresources . . . . .	8
2.4.3	/etc/ha.d/ha.cf . . . . .	8
2.4.4	Služby . . . . .	9
2.5	Sdílený filesystém . . . . .	9
2.5.1	CODA . . . . .	9
2.5.2	SW-RAID + NBD . . . . .	9
2.5.3	drbd . . . . .	10
2.5.4	SCSI či FibreChannel . . . . .	10
2.5.5	Vlastní replikace . . . . .	11
2.6	3...2...1...zážeh! . . . . .	11
2.6.1	Testování . . . . .	11
2.7	Problémy . . . . .	11
2.7.1	Problémy teoretické . . . . .	11
2.7.2	Problémy praktické (caveats) . . . . .	12
2.8	Vyzkoušená řešení . . . . .	13
2.8.1	Mail a web . . . . .	13
2.8.2	Mail a firewall . . . . .	13
2.8.3	Jiné reference . . . . .	13
<b>3</b>	<b>Závěr</b>	<b>15</b>

# Kapitola 1

## Bylo, nebylo...

Bylo, nebylo, v jednom městě stál domeček a v tom domečku bydlel Honza. Pracoval jako dobře placený správce IT v prosperující firmě. Na nějakém starším serveru si nainstaloval Linux (tu nejlepší distribuci) a na něm mailový démon (taky ten nejlepší). Server krásně fungoval, zaměstnanci firmy i pan ředitel odesílali a přijímali maily a všichni byli spokojení. Když pan ředitel zjistil, kolik stojí jiné komerční mailové systémy a spočítal si, kolik Honza firmě ušetřil, dal mu za odměnu služební letadlo. Honza však nezpychl a v nevyužitě pracovní době dál odpovídal na začátečnické dotazy z konference <linux@linux.cz>. Tak by to mohlo jít až dodnes, ale to by tahle pohádka byla moc krátká... (Nebřečte děti, Honzovi se nic nestane.)

Jednou Honza odpověděl na všechny začátečnické dotazy a najednou s údivem zjistil, že už deset minut žádný začátečnický dotaz nedorazil. Dokonce ani žádná flamerwar! Nevěřil svým očím a šel zkontrolovat svůj server. A opravdu, zjistil, že se serverem se děje opravdu něco divného. Po několika minutách přišel na to, že disk v serveru se odebral do věčných lovišť (přece jenom byl už server starší, že). Honza zalezl do skladu, našel v tajných zásobách nový disk, obnovil obsah disku ze zálohy a za tři hodiny mohl znovu odpovídat na začátečnické dotazy...

### 1.1 Co to je za ptákovinu?

Dobře, tak trochu vážněji.

Pokud se zamyslíme nad tím, co se všechno stalo, narazíme na několik problémů.

1. Honza obnovil data ze zálohy, tedy všechny změny od poslední zálohy se ztratily. Až si znovu stáhne maily, bude tam mít staré začátečnické dotazy, které už byly zodpovězené. Pan ředitel, který je teď na dovolené, naopak přijde o maily, které došly mezi zálohou a havárií systému. Až to zjistí, pravděpodobně Honzovi odebere služební letadlo.
2. V době, kdy nefungoval server, protože Honza hledal náhradní disk a obnovoval

za zálohy, nemohla celá firma číst maily. Pokud se to donese panu řediteli, Honza přijde o letadlo.

Pomineme-li fakt, že Honza je pečlivý správce, protože pravidelně zálohuje (kolik ze správců zálohuje?), musíme si položit otázku: „Jak mohl Honza následkům havárie předejít?“

- a) Použít nový server? – I nové železo občas může dělat problémy.
- b) Použít RAID? – RAID je sice pěkná věc, ale chrání pouze proti havárii disku. Kdyby neshořel disk, ale řadič disků, je Honza tam, kde byl před chvilkou.
- c) Postavit mail server jako High Available cluster? – Ano c) je správně!

## Kapitola 2

# Cluster a HA

Existuje několik různých výkladů slova cluster. V angličtině to značí shluk, v počítačové terminologii se slovo cluster používá pro několik počítačů, které zajišťují jednu činnost. V následujícím textu budu (poněkud nepřesně, leč snad srozumitelně) používat termín cluster pro to, co se obvykle označuje jako HA Cluster.

Anglický termín High Availability (HA) se do češtiny překládá nejčastěji jako „vysoká dostupnost“. Nevím, zdali existuje přesná definice, co si pod tím jeden může představit, neb každá firma o svých HA řešeních bude tvrdit něco jiného, nazývat a definovat je jinak. Pro intuitivní představu však můžeme říci, že HA řešení (=hardware + software + konfigurace) by mělo být odolné proti výpadkům – zejména proti nepředvídaným poruchám hardware.

Pěkným příkladem HA řešení je DNS – výpadek jednoho nameserveru by celosvětovou DNS vyřadit neměl.

Snažíme se tedy zajistit dostupnost služby i v případě poruch hardware. V Honzově případě je službou mail, ale může to být cokoli jiného, co je poskytované po síti klientům, například webové stránky, databáze (MySQL), či třeba slovník.

Protože potřebujeme zajistit dostupnost služby v případě výpadku hardware, musíme mít k dispozici alespoň dva počítače. Jeden z počítačů bude pracovat (nazývejme ho „pracující“) a druhý („dozorce“) na něho bude dávat pozor. Když se „pracující“ přestane fungovat, „dozorce“ to zjistí a začne dočasně pracovat místo něho. Když se „pracující“ zotaví, „dozorce“ může přestat pracovat a začne ho zase hlídat.

Kdyby Honza z úvodní pohádky použil HA řešení, při havárii jednoho počítače z clusteru by se nic vážného nestalo. Druhý počítač by převzal jeho úlohu, Honza by to zjistil a začal by problém řešit. Protože by pro ostatní zaměstnance firmy všechno fungovalo, nikdo by ho neotravoval s telefonáty typu „Tady Franta Běžný, nefunguje mi mail, můžete se na to přijít podívat?“ a mohl by v klidu porouchaný počítač opravit.

Že to zní to jako z pohádky? Ale kdepak, to už je realita...

## 2.1 heartbeat? heartbeat!

Nejjednodušší implementací třídního řešení s „pracujícím“ a „dozorcem“ je právě démon heartbeat. Původně pochází z <http://www.linux-ha.org/heartbeat/>, dnes se však vyskytuje v každé slušné distribuci (v distribuci Debian to je balík heartbeat). Vzhledem k tomu, že heartbeat jsem zkoušel pouze na Debianu (woody), budu dále používat cesty a nastavení, které jsou standardní na tomto systému.

Heartbeat umožňuje standardně propojit do clusteru právě dva počítače. Pro každý počítač je definována sada služeb, která na něm má v klidovém stavu běžet. Po startu démona se nejprve pokusí domluvit s démonem na druhém počítači v clusteru, pokud zjistí, že funguje, nastartuje služby, které mají běžet. Pak si pravidelně s démonem na druhém počítači vyměňují zprávy: „Jak se máš? Já se mám dobře.“ Pokud démon v průběhu činnosti zjistí, že druhý počítač mu dostatečně dlouho neodpověděl, rozhodne se jednat. Ověří si, že druhý počítač je opravdu mrtvý a potom na svém počítači nastartuje služby, které běžely na havarovaném počítači, případně přebere i jeho IP adresu. Havárie by se měla projevit maximálně několikaminutovým výpadkem – to je součet doby, než démon zjistí, že druhý počítač neodpovídá a doby potřebné pro start služeb. Poté, co se havarovaný počítač zotaví, jeho démon ohlásí: „Já už se mám zase dobře“, počká, až se jeho služby na zdravém počítači vypnou a přebere se IP adresa a služby se spustí znova na původním počítači.

Tolik princip, jak na to?

## 2.2 Návrh

Před tím, než začneme instalovat je dobré provést nějaký návrh.

Víme, že jeden démon heartbeat umí obsloužit cluster pouze o dvou počítačích. Určíme si službu či démona, který má být trvale dostupný. Aby byla služba dostupná i při migraci na druhý server, je potřeba kromě běžných IP adres obou počítačů vybrat i další IP adresu, na které by měla být služba dostupná pro ostatní počítače. O tuto IP adresu se budou oba počítače v případě havárie „přetahovat“.

Posílání a odesílání zpráv „Jak se máš? Já se mám dobře.“ se občas také říká heartbeat či heartbeating. Démon heartbeat podporuje tři typy těchto zpráv: heartbeat zasílaný jako UDP datagramy, multicast či heartbeat po sériové lince. I tady musíme rozhodnout, kterou volbu (či jejich kombinaci) použít.

Příklad: mějme dva počítače **bob** (192.168.0.1/24) a **bobek** (192.168.0.2/24). Chceme stejně jako Honza postavit mail server, zajímavé služby tedy budou SMTP, POP3 a IMAP4. Tyto služby by měly běžet na adrese 192.168.0.250, v klidovém stavu poběží všechny služby na počítači **bob**. Služby se startují a zastavují startovacími skripty `/etc/init.d/mailler`, `/etc/init.d/pop3` a `/etc/init.d/imapd`. Velmi doporučuji najít nějaký sériový kabel (třeba ty, co dodává HP ke každému svému switchi :-)) a použít ho také jako linku pro heartbeating. Připojíme ho na obou počítačích na port `/dev/ttyS0`.

---

## 2.3 Instalace

Poté, co si ujasníme, co vlastně chceme, můžeme démona stáhnout a nainstalovat. Na obou počítačích provedeme instalaci balíčku příkazem

```
# apt-get install heartbeat
```

Budeme-li instalovat pomocí Debianího balíčkovacího systému, bude nám tento klást různé záludné otázky, k nim patřící odpovědi posléze zapíše do konfiguračních souborů. Správné odpovědi na tyto otázky ponechám laskavému čtenáři za domácí úlohu, po přečtení tohoto článku by toho měl být schopen.

Démon se má občas po instalaci (či instalacích, deinstalacích a jiných experimentech) tendenci spouštět, proto je vhodné jej před změnami konfigurace vypnout. Protože je heartbeat standardní služba, můžeme ho vypnout běžným startovacím skriptem:

```
# /etc/init.d/heartbeat stop
```

## 2.4 Konfigurace

Téměř celá konfigurace démona heartbeat je uložena v adresáři `/etc/ha.d/`, výjimkou jsou pouze startovací skripty služeb. Heartbeat při hledání startovacích skriptů hledá v adresářích příslušných dané distribuci (tedy `/etc/init.d/`) a potom ve vlastním adresáři `/etc/ha.d/resource.d/`. Některé služby však nemají svůj skript v `/etc/init.d/` a musíme ho umístit do druhého adresáře – některé skripty, například pro IP takeover (to je to přebrání IP adresy), jsou již v tomto adresáři připraveny.

Všechny standardně nainstalované konfigurační soubory obsahují dostatečné množství dobře napsaných komentářů, takže není velký problém vyladit si konfiguraci podle svého.

### 2.4.1 `/etc/ha.d/authkeys`

Obě instance démona při heartbeatu mohou používat autentizaci pro ověření, že si opravdu povídají se správným partnerem. V souboru `authkeys` je definováno, který způsob ověřování bude použit. Na výběr jsou 3 varianty: žádné ověřování (pouze CRC), sha1 a md5. Pro sha1 i md5 je musí být zadán klíč, který se použije pro generování autentizačního hashe. Na obou počítačích je zapotřebí zadat stejný způsob autentizace i klíč. V případě instalace na Debianu tento soubor nemusíme upravovat, při instalaci balíčku je automaticky upraven.

## 2.4.2 /etc/ha.d/haresources

Do tohoto souboru se zapisují služby (resp. startovací skripty), které mají na obou strojích běžet. Soubor obsahuje dvě řádky se jmény strojů (Definice: jméno stroje je to, co dostanete po výpisu `uname -n`), za jménem stroje jsou uvedeny služby (startovací skripty), které dotyčný stroj v klidovém stavu obhospodařuje. Jména startovacích skriptů mohou být následována parametry oddělenými čtyřtečkou (: :). Doporučuje se, aby soubor `haresources` byl na obou počítačích shodný. (V praxi to není potřeba, je však velmi dobré vědět, co člověk dělá.) Soubor `haresources` pro náš příklad může vypadat například následovně (pro oba počítače):

```
bob mailer pop3 imapd IPaddr::192.168.0.250/24
bobek
```

Po startu démona `heartbeat` se tento pokusí navázat spojení s druhým počítačem v clusteru, pokud se mu to v určené době podaří, zkusí spustit všechny startovací skripty uvedené v souboru `haresources`. Startovací skripty se hledají v adresářích `/etc/ha.d/resource.d/` a `/etc/init.d/`. Démon na počítači **bob** tedy postupně spustí následující příkazy

```
# /etc/init.d/mailer start
# /etc/init.d/pop3 start
# /etc/init.d/imapd start
# /etc/ha.d/resource.d/IPaddr 192.168.0.250/24 start
```

První tři příkazy jsou zřejmé a tedy nezajímavé, poslední příkaz volá skript `IPaddr`, který je dodáván s balíčkem `heartbeat`. Tento skript způsobí přidání IP adresy na vhodné síťové rozhraní a rozeslání ARP paketu s oznámením o změně MAC adresy do připojené sítě. („Vhodné síťové rozhraní“ znamená patřící do stejné sítě jako jiné již zkonfigurované rozhraní, podrobnější pravidla jsou ve vzorovém konfiguračním souboru `haresources`.) Tímto se pro obsluhované počítače zpřístupní adresa `192.168.0.250/24`, na které poslouchá `mailer`. Počítač **bobek** nespustí nic, neb to po něm v konfiguračním souboru není požadováno.

Při vypínání služeb probíhají operace analogicky, místo `start` se použije `stop`. V případě výpadku počítače `bob` to po určitém timeoutu počítač `bobek` zjistí a nahodí služby, které dosud běžely na počítači `bob` – tj. zavolá opět ony čtyři startovací skripty. Po obnovení počítače `bob` se oba počítače domluví, `bobek` služby (opět) pomocí startovacích skriptů vypne a `bob` je na sobě spustí. Primitivní, ale účinné, že?

I soubor `haresources` se na Debianu upravuje automaticky po instalaci balíčku a nemusíme do něho zasahovat.

## 2.4.3 /etc/ha.d/ha.cf

V tomto souboru se nastavuje ostatní konfigurace démona `heartbeat`, například, zdali bude `heartbeating` probíhat přes UDP či sériovou linku, nastavení UDP paketů či sériové linky, nastavení logovacích souborů, různé timeouty a rozličná další nastavení.

Vzhledem k tomu, že konfigurace je velmi dobře okomentovaná přímo v souboru, domnívám se, že ji zde není nutné popisovat. I tento soubor na Debianu se upravuje automaticky po instalaci balíčku.

#### 2.4.4 Služby

Kromě konfigurace démona heartbeat je samozřejmě ještě velmi vhodné nastavit i normální služby, např. zmiňovaný SMTP démon. Protože zálohované služby budou startovány až heartbeatem, je vhodné ujistit se, že jsou odstraněny z běžných startovacích posloupností (adresáře `/etc/rcX.d`).

### 2.5 Sdílený filesystém

Pro většinu služeb je potřeba použít nějaký způsob sdílení dat. Typickým příkladem je mailserver a uživatelské mailboxy: v klidovém stavu potřebujeme, aby na obou počítačích byl aktuální stav mailboxů. Jen tak je možné zajistit, že v případě výpadku neztratíme žádnou informaci.

Potřebujeme tedy při konfiguraci clusteru způsobit stav, kdy se v klidovém stavu obsah filesystému na jednom počítači replikuje (česky: propisuje :-) na druhý počítač. Při havárii začne přeživší počítač využívat svoji kopii disku čímž je zajištěno, že pracuje s poslední zdravou verzí dat. Při oživení havarovaného počítače je potřeba opět sesynchronizovat oba disky.

#### 2.5.1 CODA

Pro replikování disků mohou být výhodné síťové filesystémy, které jsou distribuované a odolné proti výpadkům, například CODA. Vzhledem k tomu, že se mi ji po několika dnech experimentování včetně čtení manuálů nepovedlo uspokojivě rozchodit, nepoužil jsem ji. Předpokládám ale, že CODA by mohla být dobré řešení, nejspíš jsem jí nevěnoval dostatek času. (Možná časem...)

#### 2.5.2 SW-RAID + NBD

Pavel Machek do jádra Linuxu implementoval Network Block Device. Jde o blokové zařízení, které umožňuje přistupovat přes síť k jinému blokové zařízení. Jeden z prvních pokusů o sdílení disku jsem prováděl právě se softwarovým RAIDem (RAID1) a NBD, tato varianta je doporučovaná i na stránkách projektu HA Linux. SW-RAIDu však při startu dělalo problémy se spojit dohromady s druhým diskem na vzdáleném počítači. Poté, co jsem našel projekt drbd, který implementuje všechny potřebné vlastnosti a je určen přímo pro heartbeat, experimentů s SW-RAIDem i NBD jsem zanechal.

Nechcete-li používat různé exotické jáderné moduly, je SW-RAID a NBD jednoduchá a možná i funkční varianta.

### 2.5.3 drbd

DRBD je patch do jádra od Philipa Reisnera. Jde o jakousi variaci na síťový RAID-1.

Po kompilaci a instalaci DRBD (`make; make install`) bude vytvořen nový modul do jádra `drbd.o`, konfigurační soubory `/etc/drbd.conf`, skript pro heartbeat `/etc/ha.d/resource.d/datadisk`, obecný startovací skript `/etc/init.d/drbd`, program pro manipulaci se zařízeními `/sbin/drbdsetup` a trochu dokumentace. Po vložení modulu `drbd` do jádra vznikne blokové zařízení `/dev/nb0`. Modifikací konfiguračního souboru toto zařízení namapujeme na jiné blokové zařízení, např. `/dev/hdc1` a spárujeme jej se zařízením na druhém počítači. Zařízení se budou chvíli synchronizovat (postup synchronizace, stejně jako celkový stav zařízení můžeme sledovat v souboru `/proc/drbd`).

Po synchronizaci obou nositelských zařízení je pár zařízení `/dev/nbX` připraven k použití. Vždy jedno ze zařízení může být primární, kdy je možné jej používat pro zápis i pro čtení, druhé zařízení je možné použít pouze pro čtení. Všechny změny na primárním zařízení se přitom okamžitě posílají na zařízení sekundární, kde se provádí. Chceme-li použít sekundární zařízení pro zápis, musí se nejprve primární zařízení přestat využívat (umountovat), pak prohlásit za sekundární a stávající sekundární zařízení prohlásit za primární. Všechny změny stavu Primární/Sekundární probíhají automaticky (automagicky?) v ovladači. Uživatel musí pouze zajistit, aby se nepokoušel namountovat zařízení na obou počítačích pro zápis. Kromě stavů Primární/Sekundární existuje i třetí stav, kdy zařízení nemá připojené svoje „dvojče“. Pro přechod z tohoto stavu do primárního je potřeba provést replikaci dat – ovladač při korektním rozpojení umí poznat, která data se změnila a provést pouze rychlou replikaci, při nekorektním rozbití je potřeba počkat na synchronizaci celého pole.

Celý ovladač je chystán pro využití démonem heartbeat, proto je do něho jednoduše integrovatelný.

Možnost stažení balíčku DRBD je na stránkách projekty <http://www.drbd.org/>, pro velký rozsah a (domnívám se) malou informační hodnotu, konfiguraci nebudu popisovat zde. V případě zájmu ji mohu popsát jako další článek.

Poznámky: modul `drbd` provozují dost dlouhou dobu i na SMP stroji, takže by mohl být SMP safe. Naopak, bez úprav se nesnese s NBD, protože používá stejné major číslo zařízení. Když jsem DRBD instaloval, ještě neexistovala samostatná stránka projektu a pro jádra 2.4 se zdrojové kódy stahovaly z CVS. Jak je to dnes nevím.

### 2.5.4 SCSI či FibreChannel

Máme-li dost peněz či zkušenosti, můžeme místo „partyzánských“ metod sdílení disku použít „civilizovanou“ sběrnici, která je na sdílená zařízení stavěná. Bohužel s ní zatím nemám praktické zkušenosti. (Nějaký sponzor? :-)

## 2.5.5 Vlastní replikace

Některé programy umožňují replikaci vlastními prostředky, například. OpenLDAP server. Budeme-li chtít nasadit vysoce dostupný LDAP server, je pravděpodobně nej-jednodušší variantou použít heartbeat pro IP takeover, slapd a slurpd pro poskytování a replikaci dat.

## 2.6 3...2...1...zážeh!

Máme-li nachystané všechno, co potřebujeme ke startu prvního clusteru, můžeme jej zkusit zapnout. Na obou počítačích spustíme démona heartbeat běžným způsobem

```
# /etc/init.d/heartbeat start
```

a budeme pozorovat, co se děje. Démon nevypisuje startované služby na standardní výstup, jeho stav a případné chyby je možné dohledat v logovacích souborech.

### 2.6.1 Testování

Pokud na obou počítačích nastartovaly správné služby, můžeme zkusit pro jeden z počítačů simulovat vypnutí. Měkkým způsobem vypnutí je pouhé vypnutí démona heartbeat

```
# /etc/init.d/heartbeat stop
```

V krátké době by počítač, který zůstal „naživu“ měl převzít služby svého „mrtvého“ kolegy. Stalo-li se tak, můžeme vyzkoušet opětovné zapnutí „mrtvého“ počítače. Služby by opět měly přemigrovat na původní počítače. Funguje-li i opětné nahození, můžeme se pokusit shodit druhý počítač a pak i tvrdé vypnutí pomocí síťového vypínače. Občas může být záludné i současné nahození obou počítačů najednou, je proto vhodné vyzkoušet i to, jak se bude cluster chovat po výpadku elektřiny a jejím obnovení.

Poté, co jsme odladili veškerou myslitelnou konfiguraci, můžeme se jenom radovat, jak jsme šikovní a jít žádat svého zaměstnavatele o zvýšení platu (osobám samostatně výdělečně činným se omlouvám).

## 2.7 Problémy

### 2.7.1 Problémy teoretické

Jak říká klasik, každá kladná věc má i svou stinnou stránku, je zřejmé, že i řešení s heartbeatem budou nějakým způsobem omezená.

- Je zřejmé, že pokud se najednou porouchají dva počítače, nepomůže nám ani heartbeat. Dá se však ukázat, že porucha dvou počítačů najednou je (při předpokladech nezávislosti) méně pravděpodobná než porucha jednoho. (Že by úloha do nových skript z pravděpodobnosti?)
- Heartbeat umožňuje clusterovat pouze dva počítače. V manuálu se tvrdí, že je rozšiřitelný i nad dva počítače, nezkoušel jsem to. I kdyby rozšiřitelný nebyl, neměl by být problém monitorovat například tři počítače čtvrtým, stačí vhodně nakonfigurovat tři instance heartbeatu na záložním počítači. V praxi jsem toto řešení nezkoušel.
- Pokud spadne heartbeatová linka, démon může být zmaten. Je proto lepší do obou počítačů přidat jednu síťovou kartu s vyhrazenou IP adresou, která bude zajišťovat propojení obou počítačů přes křížený kabel. Používáte-li drbd, je možné tuto druhou linku použít i pro synchronizaci obsahu disku (rekord asi 8.5MiB/s). Praktickou realizaci ponechávám jako domácí úkol laskavému čtenáři.

## 2.7.2 Problémy praktické (caveats)

Jak praví Murphy: „I tam, kde není žádný problém, je jeden malý, který by rád ven.“ Při reálném nasazení heartbeatu můžete narazit na problémy, které jsou skryty v počátečním nadšení pro HA:

- Nevýhodou clusteringu je fakt, že všechno musí být zdvojené. V praxi to řeším tak, že dva stroje s rezervou výkonu „spáruji“. Například mail server a HTTP proxy server hlídají navzájem jeden druhého. Když vypadne mailer, proxy server začne přijímat a odesílat maily, v případě výpadku proxy serveru jeho služby dočasně zastane mailer. Ve většině času je každý jeden počítač zatížen pouze jednou službou.
- Aby služby byly opravdu identické, je potřeba udržovat stejné konfigurační soubory na obou strojích. Zatím to řeším pomocí scp, další možnost je rsync či sup. Zajímavou možností, která mne napadla je použít User Mode Linux (UML) pro běh clusterovaného počítače. Velmi by se tím mohla zjednodušit správa konfiguračních souborů. S UML zatím nemám žádné zkušenosti, takže myšlenky jsou pouze v teoretické rovině.
- Je zřejmé, že heartbeat je možné nasadit pouze pro některé demony – zejména pro služby, které jsou založené na pouze krátkodobých spojeních a démoni svůj stav ukládají na disk. I přesto však jde o dobrou možnost, jak zvýšit odolnost vašich serverů a služeb, neboť většina síťových služeb této podmínce vyhovuje. Je možné nasadit heartbeat relativně jednoduše pro mail server, sambu, LDAP. Pokud chcete ale zálohovat například freeciv server, musíte vynaložit větší množství úsilí.

- Budete-li clusterovat mailserver, nezapomeňte, že některé programy (cron) mají ve zvyku posílat maily správci systému. Je proto velmi vhodné, aby i na neaktivním počítači byl `/usr/bin/sendmail`, který bude i v případě neaktivity přeposílat maily na druhý stroj. V praxi jsem to vyřešil vhodně sestavenými skripty, které mění konfiguraci qmailu a pak jej restartují.

## 2.8 Vyzkoušená řešení

### 2.8.1 Mail a web

První stroj	Druhý stroj
Mailer	Websvrer, mailing list, databáze
qmail (SMTP, POP)	Apache, Mailman, MySQL
Courier IMAP (IMAP)	

Oba stroje jsou značkové repasované počítače, všechny disky jsou SCSI v RAIDu1. Replikované disky pro mailboxy, webové stránky a databázi jsou řešeny přes drbd. Do počítačů je přidána další síťová karta, která slouží jako heartbeat a zároveň se používá pro replikaci disků. Původně byl první počítač nainstalován samostatně, po měsíci provozu jsem jej spojil do clusteru s druhým. Toto řešení chodí bez problémů už několik měsíců.

Našli jsme pro něho s kolegou i další uplatnění – při restartech počítačů (třeba kvůli výměně jádra) se koncovým uživatelům neprojeví žádné velké přerušení dostupnosti služby.

### 2.8.2 Mail a firewall

První stroj	Druhý stroj
Mailer	Firewall, proxy server
qmail (SMTP, POP3)	

Po počátečních problémech (stroje při převozu a havárii proletěly zadním okénkem od auta) a drobných problémech s konfigurací i tento pár také funguje ke spokojenosti jeho uživatelů.

### 2.8.3 Jiné reference

V jedné nejmenované české velké firmě mají na Linuxu postavené dvě replikované databáze. Na dvou velkých počítačích běží dva databázové stroje, které sbírají data z výroby. Disky jsou připojené ze sdíleného diskového pole přes SCSI (či FibreChannel?). Při poruše jednoho z počítačů se diskové pole přemountuje do druhého, přebere se jeho IP adresa a obě databáze fungují dále.

Informační systém Masarykovy univerzity <http://is.muni.cz/> používá pro distribuci zátěže na farmu Linuxových strojů počítač, který přijímá HTTP požadavky a přeposílá je dále (LVS). Tento stroj je také zálohován heartbeatem. (Dle vyjádření Yenyi Kasprzaka).

Na Internetu je možné najít další reference (Samba, LDAP, apod.), dobrou startovní adresou může být např. <http://www.linux-ha.org/> či přímo URL démona heartbeat <http://www.linux-ha.org/heartbeat/>.

## Kapitola 3

### Závěr

I přes popsané nevýhody se domnívám, že nasazení heartbeatu je velmi lákavá a výhodná varianta. V praxi jej již mám nasazen přibližně půl roku a věřím, že mi ušetřil mnoho času i nervů.

Na tomto místě bych rád poděkoval Petrovi „Klokanovi“ Přidalovi za cenné připomínky k článku.

---

Tento článek vyšel na pokračování :-( na serveru `root.cz` na přelomu roku 2003 a 2004. Komentáře k článku obsahují další náměty, proto je velmi doporučuji k přečtení.

- První díl <http://www.root.cz/clanek/1973>
- Druhý díl <http://www.root.cz/clanek/1973>
- Třetí díl <http://www.root.cz/clanek/1985>
- Poslední díl <http://www.root.cz/clanek/1994>

PS: A nejsem Johanka... :-)