Architektúra
Náklady
Soundslash

29. januára 2015
Michal Bystrický
bystricky@soundslash.com

Obsah

1 Úvod **3**

2 Aktuálny Stav Architektúry **3**
2.1 Hlavný Prípad Použitia 3
2.2 Typy Služieb 3
2.3 Architektúra a Hlavný UC 4
2.4 Opis Serverov 4

3 Testovacie Servery **5**
3.1 Fyzický Server 5
3.2 VPS 5

4 Minimálne Konfigurácie **6**
4.1 Všeobecné Ohraničenia 6
4.2 Optimálne konfigurácie 6

5 Cenové Ponuky **11**
5.1 VPSTech 11
5.2 WebSupport 11

6 Kalkulácia Nákladov a Služieb a Záver **12**

A MongoDB servers **15**

B Konfigurácia VPS **15**

C Skript na Meranie CPU, RAM a I/O **15**

D MongoDB Používanie Aplikácie **15**

1 Úvod

Tento dokument opisuje architektúru služby Soundslash a architektúru nasadenia. Jeho cieľom je vypočítať náklady na prevádzku služieb Soundslash pri danej architektúre. Opiera sa o dostupné metriky a zdroje vďaka ktorým počíta veľmi približnú kalkulácia nákladov pre jednotlivé služby.

2 Aktuálny Stav Architektúry

Táto sekcia opisuje aktuálny stav architektúry. Sekcia 2.1 opisuje hlavný prípad použitia na úrovni summary (strategic) cieľov a konsekventne Sekcia 2.3 architektúru za ním. Sekcia 2.2 sa venuje typom služieb z pohľadu architektúry. Sekcia 2.4 dodáva k hlavnému prípadu použitia ďalšie dôležité architektonické rozhodnutia.

2.1 Hlavný Prípad Použitia

Hlavný prípad použitia predstavuje vytvorenie internetového rádia a jeho manažment. Teda mohli by sme povedať, že na úrovní strategických cieľov sa tento prípad použitia bude volať "Správa rádia". Voľne ho môžeme opísať nasledovnými krokmi:

1. Organizácia vyberie "Vytvorenie rádia"
2. Systém vytvorí rádio
3. Systém spustí rádio
4. System umožní organizácii rádio spravovať
5. Systém umožní rádio počúvať
6. Organizácia nahrá média
7. Organizácia vytvára program
8. Organizácia robí LIVE vstupy
9. Poslucháč počúva rádio

2.2 Typy Služieb

Z pohľadu architektúry existujú 2 typy služieb:

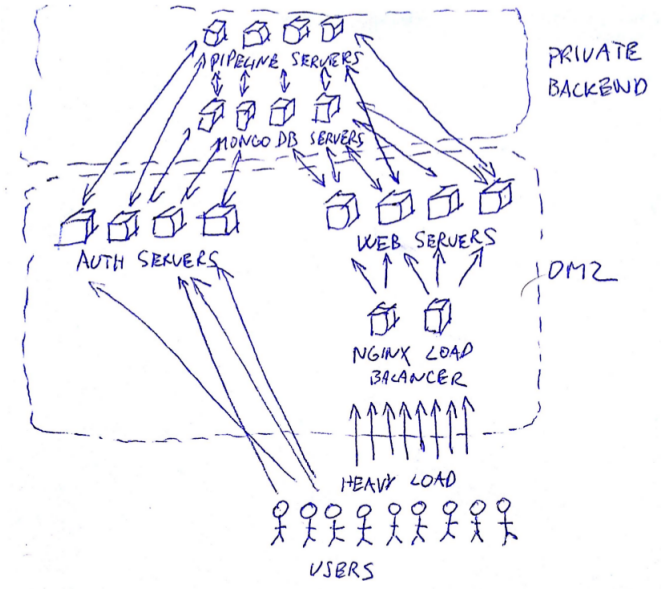
- re-enkódované,
- enkódované.

Ak ide o re-enkódovaný stream, média sa dekódujú, mixujú a enkódujú za behu. Takto napríklad môžeme primixovať LIVE stopu priamo z mikrofónu, alebo realizovať fade efekt na audio stopu. Pri enkódovaných nie je potreba tohto processingu, pretože už sú média spracované a posielajú sa priamo poslucháčom. Avšak, pri enkódovanom type nie je možné mixovať alebo inak upravovať stream.

2.3 Architektúra a Hlavný UC

Teraz si prejdeme architektúru za týmto prípadom použitia. Služba Soundslash je škálovateľná. Na Obrázku 1 môžeme vidieť architektúru a rozloženie záťaže. Treba si uvedomiť, že každý komunikuje s každým na lokálnej sieti (nie nevyhnutne), okrem Web a Auth serverov. Ako identifikátor aplikácia vie svoju lokálnu IP a verejnú IP.

DMZ nie je prístupná z Internetu, teda používatelia sa pripájajú na load balancer a Icecast streaming servery. Ďalej, load balancer rozloží záťaž na web servery, kde sa realizuje aj encoding audio súborov. Ak je encoding dokončený pridá sa tento audio súbor do databázy MongoDB. Média spolu s programom sa ukladajú do databázy.

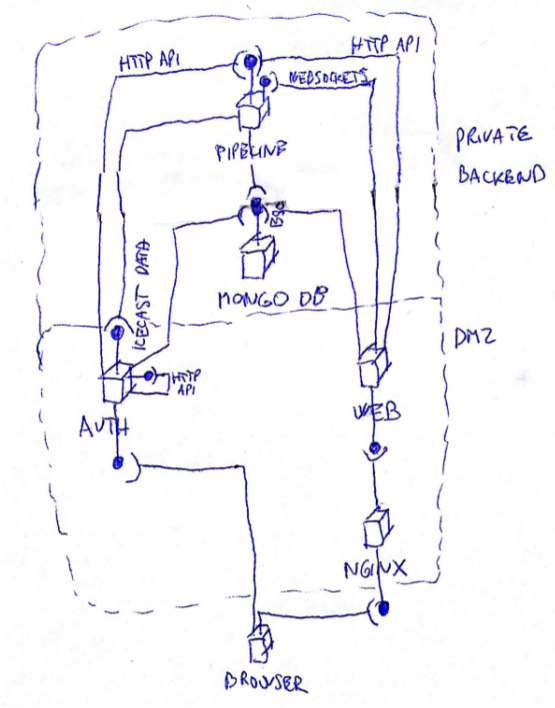


Obr. 1: Architektúra služby Soundslash

Asynchrónne z tejto databázy podľa programu, sú média vyberané, (dekódované, mixované, enkódované—v závislosti od typu služby) a posielané do Auth serverov. Na Auth serveroch beží streaming server Icecast, ktorý aktualizuje tabuľku streamov k serverom v databáze. Teda, ak už je poslucháčov mnoho Auth aktualizuje túto tabuľku. Následne pri pripojení ďalšieho poslucháča podľa tabuľky buď (1) pošle Web server požiadavku na Pipeline o pripojení ďalšieho Auth servera (2) alebo v prípade, že Auth server nie je plný, tak ho pripojí na tento server.

Teda, Pipeline realizuje samotné škálovanie (pripájanie) Auth serverov, pretože on realizuje samotný stream. Aby používateľ mohol ovládať stream, Pipeline má 2 interfejsi, ako môžeme vidieť na Obrázku 2. Prvý HTTP API prijíma správy, ktoré ovládajú stream, druhý Websockets slúži na posielanie LIVE dát z mikrofónu.

Na serveroch Auth beží Icecast streaming server, ktorý počúva a prijíma stream, ktorý ďalej distribuuje poslucháčom. Server Icecast cez HTTP API komunikuje s Auth serverom o streamoch, teda, aktualizuje mu nasledovné:



Obr. 2: Interfejsi služby Soundslash

- pridanie nového streamu,
- odpojenie streamu,
- pripojenie poslucháča,
- odpojenie poslucháča.

Auth server potom na základe týchto informácii aktualizuje tabuľku o dostuponosti Icecast streaming serverov, ktorej príklad môžeme vidieť v prílohe A. Atribút level je od stupňa 0 do 10, kde 10 predstavuje úplne plný server a 0 práve vytvorený. Tento atribút je aktualizovaný pri Pipeline podľa CPU a pri Auth podľa poslucháčov.

2.4 Opis Serverov

Pipeline server. Hlavná úloha je práca so streamom, teda dekódovanie, mixovanie, enkódovania a pripájanie nových Icecast streaming serverov. Tento server si sám aktualizuje tabuľku o vyťaženosti (atribút `level`). Ak je moc vyťažený ďalšie streamy nespúšťa. Web server potom hľadá iný menej-vyťažený, ak je nenájdený, používateľovi je oznámený problém o nedostatku serverov.

Metódy, ktoré je možné spúšťať na Pipeline HTTP API:

```
(r"/live.json", LiveHandler),
(r"/updates.json", UpdatesHandler),
(r"/start-streaming.json", StartStreamHandler),
```

```
(r"/restart.json", RestartStreamHandler),
(r"/scale.json", ScaleHandler),
(r"/is-alive.json", IsAliveHandler),
(r"/playlist-update.json", PlaylistUpdateHandler),
(r"/run-command.json", RunCommandHandler)
```

Posledná metóda umožňuje spúšťať príkazy z webového rozhrania a to sú nasledovné:

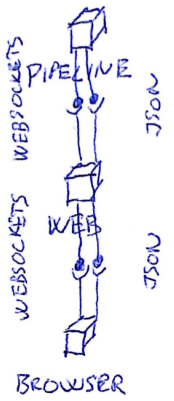
- use, dump_dot_file, start, stop, next, scale, rescale, playlist

Auth server. Jeho hlavná úloha je autentifikácia poslucháčov. Icecast streaming server sa pýta priamo tohto servera cez HTTP API, či poslucháčovi pustiť stream alebo nie. Aktualizuje tabuľku o Icecast streaming serveroch a ak Pipeline prestane posielať stream, opýta sa ho, či je dostupný, ak nie je, tak ho označí za nedostupný.

Metódy, ktoré je možné spúšťať na Auth HTTP API:

```
(r"/mount_add", MountAddHandler),
(r"/mount_remove", MountRemoveHandler),
(r"/listener_add", ListenerAddHandler),
(r"/listener_remove", ListenerRemoveHandler),
```

Web server. Ak Pipeline nie je dostupný označí ho za nedostupný. Získava dáta o dosupnosti z tabuľky o dostupnosti z databázy (server MongoDB). Odosiela správu na spustenie nového streamu, alebo na spustenie nového Icecast streaming servera, ak je to potrebné.



Obr. 3: Detail Web server

Na Obrázku 3 môžeme vidieť detail, ako Web server komunikuje s prehliadačom. Web server poskytuje iba datovú vrstvu nad databázou (JSON), teda celý obsah je generovaný na strane klienta. Ide o tučného klieta, kde URL routing, generovanie frontendu a logika aplikácie je presunutá ku klientovi. Server iba odpovedá na požiadavky,

ktoré nevyhnutne potrebujú databázu alebo pripojenie k Pipeline. Websockets interfejs je požitý na aktualizáciu piesní pomocou vzoru Observer a tiež na odosielanie dát z mikrofónu.

3 Testovacie Servery

Táto sekcia opisuje servery na ktorých budú robené testy a tiež diskutuje minimálne konfigurácie pre jednotlivé typy serverov. Sekcia 3.1 predstavuje fyzický server, na ktorom bude umiestnený virtuálny server (VPS) opísaný v Sekcii 3.2.

3.1 Fyzický Server

Fyzický server má 2x CPU po 8 vláknach, čo je spolu 16 jadier. Na stroji je 16x VPS virtualizované na technológii KVM. Tabuľka 2 prehľadne zobrazuje konfiguráciu fyzického servera. Na tomto fyzickom serveri je umiestnená 1x VPS opísaná Sekcii 3.2.

CPU	2x 2266.746 MHz
Cores	2x8=16

Tabuľka 1: Fyzický server

3.2 VPS

Tabuľka 2 prehľadne zobrazuje minimálnu konfiguráciu spustenia služby na 1 VPS. Pozri Sekciu 4 pre viac informácii.

Cores	2
RAM	512 MB
HDD	40 GB

Tabuľka 2: VPS

4 Minimálne Konfigurácie

Táto sekcia sa venuje minimálnym konfiguráciam. Sekcia 4.1 diskutuje o všeobecných ohraničeniach. Sekcia 4.2 následne opisuje optimálne konfigurácie pre jednotlivé servery.

4.1 Všeobecné Ohraničenia

CPU. Minimálna konfigurácia CPU je minimálne 2 jadrá (pokiaľ máme celú aplikáciu na na 1 VPS), pretože server Pipeline si aktualizuje svoje vyťaženie na základe CPU a teda počas processingu médii by sa mohol označiť ako plne využitý a už ďalej nespúšťať nové rádiá. Teda hneď po vytvorení rádia (po processingu prvej audio stopy) by nebolo spustené.

HDD. Operačný systém spolu so softvérom, aplikáciou a knižnicami je veľký do 5 GB, ako je vidno v prílohe B. Teda minimálna veľkosť na diskový priestor je 5 GB (bez akejkoľvek databázy).

RAM. Podľa prílohy B minimum pre spustenie aplikácie je 256 MB RAM.

Traffic. Sieťová prevádzka závisí od počtu poslucháčov. Pri 8 hodinovom počúvaní denne, 1 rádio, 10 poslucháčov, 128 kbps za mesiac je to 2.3 GB. Tabuľka 3 prehľadne zobrazuje ďalšie variácie.

Počet rádií	Poslucháčov	Čas počúvania [h/d]	Bitrate [kbps]	Traffic [GB/mes]
1	1	8	128	0.225
1	300	8	128	66
10	1	8	128	2.3
10	100	8	128	220
100	100	8	128	2 200
100	300	8	128	6 600
100	300	8	256	132 000

Tabuľka 3: Traffic

4.2 Optimálne konfigurácie

Táto sekcia rozoberá optimálne konfigurácie v závislosti od typov serverov.

Pipeline. Pipeline je server, kde beží mnoho GStreamervlákien, preto vhodným nastavením by bola konfigurácia s viacerími jadrami. Pre VPS zostavu sme vykonali test, ktorého výsledky možno vidieť v Tabuľke 4. Treba upozorniť, že testovacia zostava je 2 jadrová, teda ak jedno jadro je vyťažené na 100% a druhé na 0%, výsledne bude 50% záťaž. Opäť, ak budeme mať silnejšie alebo slabšie fyzické CPU, záťaž sa môže líšiť.

Mohli by sme predpokladať, že 8% CPU je vyžadovaného na 1 reencoding stream pri danej zostave. RAM stúpala o cca 5%, čo je cca 26 MB. Základ pamäte RAM je 59%, čo predstavuje 300 MB. Nasledovná tabuľka prehľadne opisuje požiadavky na 1 re-encoding stream.

CPU	8%
RAM	26 MB

Daným pomerom získame optimálnu konfiguráciu. Na dané CPU sa zmestí cca 10-12 re-encoding streamov, pričom pre 12 streamov potrebujeme 300+(12*26) = 612 MB RAM. Preto optimálna zostava pre Pipeline server je uvedená v Tabuľke 5.

Počet reencoding	Počet encoding	Dĺžka testu [min]	CPU AVG [%]	CPU MAX [%]	RAM [%]
1	0	2	7.308	12	59.2
1	1	2	9.233	14	64.5
2	1	2	15.3	23	66.8
2	2	2	16.6	24	69.5
3	2	2	22.7	30	74
4	2	2	28.9	41	77.4
5	2	2	33	47	81
0	1	2	1.6	4	65.1
0	2	2	2.9	8	66.1
0	3	2	1.76	4	67.6
0	4	2	4	8	69.2
0	5	2	2.05	6	71
0	6	2	2.03	5	72.3

Tabuľka 4: Pipeline server test

Cores	2
RAM	756 MB
HDD	5 GB
Počet re-encoding	10-12
Percentuálne 1 re-encoding	0.1 %

Tabuľka 5: Optimálna konfigurácia Pipeline servera (re-encoding 64 kbps)

Ďalej by sme mohli predpokladať, že 1.5% CPU je vyžadovaného na 1 encoded stream pri danej zostave. Pamäť RAM sa zvyšuje o cca 2%, čo je 11 MB. Základ bol 65%, čo je 333 MB RAM. Nasledovná tabuľka opisuje požiadavky na 1 encoded stream.

CPU	1.5%
RAM	11 MB

Podľa tohto pomeru, pre encoded stream sa zmestí na daný VPS server 60-65 streamov, pričom pamäť RAM v závislosti od tohto by bola 333+(65*11) = 1024 MB RAM. Mohli by sme špekulovať, že 756 MB RAM je tiež akceptovateľné, nakoľko merania pamäte RAM obsahujú aj cache RAM. Preto optimálna zostava pre Pipeline server je uvedená v Tabuľke 9.

Cores	2
RAM	1024 MB (756 MB)
HDD	5 GB
Počet encoded	60-65
Percentuálne 1 encoded	0.017 %

Tabuľka 6: Optimálna konfigurácia Pipeline servera (encoded 64 kbps)

Auth. Server Auth úzko spolupracuje s Icecast streaming serverom pomocou HTTP API, ktoré Auth poskytuje všetko na 1 node. Preto musíme rátať s určitým prídavným zaťažením (overhead-om) okrem Icecast zaťaženia, ktoré je zanedbateľné, pretože ide o requesty do databázy. Odrazíme sa od testov [1] a [2], limitujúce faktory sú nasledovné:

- Sources. Každý source pridáva cca 1 MB RAM (672 KB).
- Traffic vs Listeners. Poslucháči nemajú vplyv na hardwarovú zostavu, práve traffic je limitujúci faktor. V teste, pri 14 000 poslucháčov narazili práve na limit trafficu.

Pipeline posiela dáta do sources, ktoré sú po 8 poslucháčov (čím menej poslucháčov v sources, tým presnejšie využitie sources). Ak sa na stream pripojí viac ako 8 poslucháčov Pipeline začne posielať audio dáta do ďalšieho source. Teda, ak rátame s nasledovným limitom, pre 512 MB RAM, kde základ je 300 MB sa nám zmestí cca 200 sources po 8 poslucháčov.

Ak by ste mali 10 Mbit/s sieť (= 1 250 kB/s = 1.25 MB) upload. Do tohto rozsahu sa nám zmestí 78 streamov pri 128 kbit/s. Tabuľka 7 zobrazuje ďalšie variácie.

Ako je možné vidieť z tabuľky pre VPS, ktorá má 512 MB RAM a 128 kbit/s streamoch, by sme vedeli pri základe 300 MB RAM vytvoriť 200 sources po 8 poslucháčov, ktoré by obslúžili 1 600 poslucháčov, museli by sme však mať cca 3x 100 Mbit/s upload. Teda, ako je vidieť z tabuľky, naozaj, limitujúci faktor je primárne traffic.

[1] http://icecast.org/loadtest/1/
[2] http://icecast.org/loadtest/2/

	10 Mbit/s	100 Mbit/s	1 Gbit/s	10 Gbit/s
64 kb/s (8 kB)	156 19/9/4	1 562 195/97/48	15 625 1 953/976/488	156 250 19 531/9 765/4 882
128 kb/s (16 kB)	78 9/4/2	781 97/48/24	7 812 976/488/244	78 125 9 765/4 882/2 441
256 kb/s (32 kB)	39 4/2/1	390 48/24/12	3 906 488/244/122	39 062 4 882/2 441/1 220

Tabuľka 7: Počet maximum konkurentných streamov pri rôznych rozsahoch siete. V druhom riadku môžeme vidieť koľko sources po 8/16/32 poslucháčov, čo sú aj požiadavky na pamäť RAM v MB.

Tabuľka 8 poskytuje iný pohľad podľa poslucháčov. Základ 200 MB by sme zarátali pri novom stroji, teda každých cca 750 (256 MB RAM), alebo 2 500 (512 MB RAM), alebo 5 000 poslucháčov (1 024 MB RAM). Teda pri prvej najneoptimálnejšej alternatíve by na 1 poslucháča vychádzalo zo zostavi 0.002 percenta, pri druhej 0.0004 % a pri tretej 0.0002 %. V Tabuľke 9 možno vidieť optimálnu konfiguráciu Auth servera.

Poslucháči	Traffic [Mbit/s]	Počet sources/RAM [MB]
1	0.128	1
8	1.024	1
50	6.4	7
100	12.8	13
300	38.4	38
750	96	93
1 000	128	125
2 500	320	312
5 000	640	625
10 000	1 280	1 250

Tabuľka 8: Požiadavky na traffic a RAM podľa poslucháčov pre 128 kbit/s stream.

Web. Ako sme už pokázali aplikácia je hrubý klient, preto po vybraní dát z databázy nie je takmer vôbec s nimi manipulované ďalej. Avšak, signifikantná záťaž je na processing médií, ktorej sa budeme venovať.

Pre jedno vlákno a pre danú VPS zostavu sme testovali nahrávanie a processing médií v Tabuľke 10. Mohli by sme predpokladať, že na danej zostave 5 minútové audio sa processuje do 30 sekúnd. Preto ak jeden album, ktorý obsahuje 10 audio stôp, je nahrávaný, dĺžka trvania nahrávania je cca 5 minút.

Teda daná VPS zostava pri 1 vlákne dokáže spracovať za 1 deň cca 2 800 médií (kde každá audio stopa je dlhá približne 5 minút). Ak počítame, že jedno rádio nahrá 1 album za 2 dni (čo je cca 10 audio stôp), tak za mesiac nahrá 150 médií, čo je 5%. Preto

Cores	1	1	1
RAM	256 MB	512 MB	1 024 MB
HDD	5 GB	5 GB	5 GB
Traffic (64 kbps)	48 Mbit/s	160 Mbit/s	320 Mbit/s
Traffic (128 kbps)	96 Mbit/s	320 Mbit/s	640 Mbit/s
Traffic (256 kbps)	192 Mbit/s	640 Mbit/s	1 280 Mbit/s
Počet poslucháčov	750	2 500	5 000
Percentuálne 1 poslucháč	0.002 %	0.0004 %	0.0002 %

Tabuľka 9: Optimálna konfigurácia Auth servera

Názov	Veľkosť [M]	Dĺžka [m:s]	Čas [sec]
Skin & Bones.mp3	3.0	4:17	20
Home.mp3	5.4	5:50	28
Always In My Head.mp3	8.3	3:36	21
Politik.mp3	16	6:53	42

Tabuľka 10: Testovanie processingu médií (aj s nahrávaním 10 Mbit/s upload do 1-4 sec a spustením streamu do 1 sec)

optimálna zostava je uvedená v Tabuľke 11.

Cores	1
RAM	512 MB
HDD	5 GB
Percentuálne 1 rádio/mesiac	5%

Tabuľka 11: Optimálna konfigurácia Web servera

MongoDB. K databáze pristupujú Web, Pipeline a Auth servery. Indexované kľúče sú v pamäti RAM, ktorá teda koreluje s počtom dát, ktoré majú index. Dáta sú na disku. Preto, limitujúce faktory sú:

- Počet requestov.
- Veľkosť disku. Pre jedno rádio odhad podľa oddychovka.eu 30 GB.
- Veľkosť RAM podľa veľkosti indexov.

Pri používaní systému sme namerali 300 queries za cca 5 minút, čo je 1 query/s. Najdlhšia query bola 75 ms pri malých dátach (10 malých rádií). Tendencia je udržať dĺžku query do 100 ms pomocou indexov. Preto, optimálna konfigurácia MongoDB servera je v Tabuľke 12. Testy sú v prílohe D.

Cores	1
RAM	512 MB
HDD	5+(30*10) = 305 GB
Percentuálne 1 rádio/mesiac	0.1%

Tabuľka 12: Optimálna konfigurácia MongoDB servera

5 Cenové Ponuky

Táto sekcia opisuje cenové ponuky pre servery.

5.1 VPSTech

V Tabuľke 13 je cenová ponuka VPSTech.

Názov	Konfigurácia	Cena
Server Pipeline	Cores: 2 RAM: 756 MB HDD: 5GB	12 E/mes
Server Auth	Cores: 1 RAM: 256 MB HDD: 5GB Traffic: 48 Mbit/s	6 E/mes
Server Web	Cores: 1 RAM: 512 MB HDD: 5GB	8 E/mes
Server MongoDB	Cores: 1 RAM: 512 MB HDD: 305 GB	44 E/mes
Server MongoDB variácia 1	Cores: 1 RAM: 512 MB HDD: 35 GB	17 E/mes
Core upgrade	1 to 2	1.5 E/mes
Core upgrade	2 to 4	2 E/mes
RAM upgrade	+256 MB RAM	2 E/mes
RAM upgrade	+512 MB RAM	4 E/mes
HDD upgrade	+30 GB HDD	9 E/mes

Tabuľka 13: VPSTech cenová ponuka

5.2 WebSupport

V Tabuľke 14 je cenová ponuka WebSupport. WebSupport neumožňuje rezervovať traffic a navyše maximálne HDD je 200 GB.

Názov	Konfigurácia	Cena
Server Pipeline	Cores: 2 RAM: 756 MB HDD: 5GB	13.28 E/mes
Server Auth	Cores: 1 RAM: 256 MB HDD: 5GB Traffic: 48 Mbit/s (!)	6.61 E/mes
Server Web	Cores: 1 RAM: 512 MB HDD: 5GB	9.46 E/mes
Server MongoDB	Cores: 1 RAM: 512 MB HDD: 305 GB (200)	80 E/mes
Core upgrade	1 to 2	1.36 E/mes
Core upgrade	2 to 4	2.21 E/mes
RAM upgrade	+256 MB RAM	3.12 E/mes
RAM upgrade	+512 MB RAM	5.83 E/mes
HDD upgrade	+30 GB HDD	12.94 E/mes

Tabuľka 14: WebSupport cenová ponuka (28.1.2015)

6 Kalkulácia Nákladov a Služieb a Záver

Táto kapitola počíta podľa predchádzajúcich meraní a výpočtov náklady na prevádzku služby Soundslash.

Re-encoding stream. Nasleduje vzorec na výpočet nákladov za mesiac:

```
  0.1 * Pipeline
+ 0.002 * pocet posluchacov * Auth
+ 0.05 * Web
+ 0.1 * MongoDB
```

Encoded stream. Nasleduje vzorec na výpočet nákladov za mesiac:

```
  0.017 * Pipeline
+ 0.002 * pocet posluchacov * Auth
+ 0.05 * Web
+ 0.1 * MongoDB
```

Podľa vzorcov v Tabuľke 16 a cenovej ponuky môžeme vidieť kalkuláciu služieb podľa počtu poslucháčov a typu streamu (re-encoding, encoded).

Typ	Počet poslucháčov	HDD [GB]	Cena [E/mes]
Re-encoding	10	3	2.16
Re-encoding	10	30	9.72
Re-encoding	10	300	45.72
Re-encoding	30	3	2.40
Re-encoding	30	30	9.96
Re-encoding	30	300	45.96
Re-encoding	50	3	2.64
Re-encoding	50	30	10.02
Re-encoding	50	300	46.02
Re-encoding	100	3	3.24
Re-encoding	100	30	10.8
Re-encoding	100	300	46.8
Re-encoding	500	30	15.61
Re-encoding	700	30	18
Re-encoding	1 000	30	21.6
Re-encoding	10 000	30	129.6
Encoded	10	30	8.724
Encoded	30	30	8.964
Encoded	50	30	9.204
Encoded	100	30	9.804
Encoded	300	30	12.204
Encoded	500	30	14.604
Encoded	700	30	17.004
Encoded	1 000	30	20.604
Encoded	10 000	30	128.604

Tabuľka 15: Náklady pre 1 stream

Server Pipeline 2 cores 756 MB RAM 5 GB HDD	12 E/mes	10 re-encoding (=60 encoded)
Server Auth 1 core 256 MB RAM 5 GB Traffic 48 Mbit/s	6 E/mes	750 poslucháčov 64 kbps (=375 p. 128 kbps)
Server Web 1 core 512 MB RAM 5 GB HDD	8 E/mes	1 processing 1 média v čase
Server MongoDB 1 core 512 MB RAM 305 GB HDD	44 E/mes	10 rádií po 30 GB
Server MongoDB 2 1 core 512 MB RAM 35 GB HDD	17 E/mes	10 rádií po 3 GB
Farma re-encoding 1	70 E/mes	10 re-encoding rádií po 37 poslucháčoch 128 kbps po 30 GB
Farma re-encoding 2	154 E/mes	10 re-encoding rádií po 296 poslucháčoch 128 kbps po 60 GB
Farma re-encoding 3	43 E/mes	10 re-encoding rádií po 37 poslucháčoch 128 kbps po 3 GB
Farma encoded	70 E/mes	60 encoded rádií po 6 poslucháčoch 128 kbps po 5 GB

Tabuľka 16: Zostavy a ich variácie a kapacity

A MongoDB servers


```
{
   "_id":ObjectId("54ac34c5b928fe01884e64f6"),
   "down":false,
   "local_ip":"192.168.0.42",
   "level":0,
   "public_ip":"94.229.33.134",
   "type":"pipeline",
   "port":9999
}{
   "_id":ObjectId("54ac34d5b928fe01884e64f7"),
   "local_ip":"192.168.0.42",
   "port":8000,
   "streaming":{
      "mount":"/silvia.ogg",
      "streaming":false,
      "password":"n71dy238",
      "listeners":0,
      "max_listeners":8,
      "stream":"54b00875ed317725b1ed9e31",
      "quality":0
   },
   "type":"streaming",
   "public_ip":"94.229.33.134",
   "down":false,
   "level":0,
   "user_id":"54b00865ed317725b1ed9e2f"
}
```


B Konfigurácia VPS

```
 # df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/soundslash--production-root 9.2G 2.4G 6.4G 27% /
/dev/mapper/soundslash--production-home 29G 3.8G 24G 14% /www
# grep MemTotal /proc/meminfo
MemTotal: 508832 kB
# free -m
             total used free shared buffers cached
Mem: 496 252 244 0 17 127
-/+ buffers/cache: 107 389
```

```
Swap: 1019 138 881
# grep "model name" /proc/cpuinfo
model name : Common KVM processor
model name : Common KVM processor
# lscpu
Architecture: x86_64
CPU(s): 2
Vendor ID: GenuineIntel
CPU family: 15
Model: 6
Stepping: 1
CPU MHz: 2266.746
BogoMIPS: 4533.49
Hypervisor vendor: KVM
Virtualization type: full
L1d cache: 32K
L1i cache: 32K
L2 cache: 4096K
```

C Skript na Meranie CPU, RAM a I/O

log.sh:

```
for i in ‘seq 1 300‘;
do
       iotop -n 1 -k| head -1|awk ’{print ($4+$10)}’ 1>> io‘echo $1‘
       eval $(awk ’/^cpu /{print "previdle=" $5 "; prevtotal=" $2+$3+$4+$5 }’ \
              /proc/stat); sleep 0.4; eval $(awk ’/^cpu /{print "idle=" $5 "; \
              total=" $2+$3+$4+$5 }’ /proc/stat); intervaltotal=$((total-${prevtotal:-0}));
              echo "$((100*( (intervaltotal) - ($idle-${previdle:-0}) ) / (intervaltotal)\
              ))" 1>> cpu‘echo $1‘
       free -m|head -2|tail -1|awk ’{print ($3*100/$2)}’ 1>> ram‘echo $1‘
       sleep 1
done
```

avg.sh:

```
cat $1|tr ’\n’ ’ ’|awk ’{v=0;for(i=0;i<NF;i++) v=v+$i;; print v/NF }’
```

D MongoDB Používanie Aplikácie

Používanie rozhrania, posúvanie piesní, spúšťanie, aktualizácia playlistov:

```
> db.system.profile.find({user: "pipeline@pipeline", ts: {$gte: ISODate("2015-01-28T14:12:54
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "update", "ns" : "pipeline.servers", "nscanned" : 2, "nscannedObjects" : 2, "millis
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.files", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "update", "ns" : "pipeline.servers", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "update", "ns" : "pipeline.servers", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "update", "ns" : "pipeline.servers", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "update", "ns" : "pipeline.streams", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "update", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.streams", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.groups", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.streams", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "command", "ns" : "pipeline.$cmd", "millis" : 0 }
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.groups", "nscanned" : 56, "nscannedObjects" : 56, "millis
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.groups", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.media", "nscanned" : 25, "nscannedObjects" : 25, "millis"
{ "op" : "query", "ns" : "pipeline.programs", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.streams", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.programs", "nscanned" : 26, "nscannedObjects" : 26, "mil
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.streams", "nscanned" : 15, "nscannedObjects" : 15, "milli
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.groups", "nscanned" : 56, "nscannedObjects" : 56, "millis
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.groups", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.media", "nscanned" : 25, "nscannedObjects" : 25, "millis"
{ "op" : "query", "ns" : "pipeline.streams", "nscanned" : 1, "nscannedObjects" : 1, "millis"
```

```
{ "op" : "query", "ns" : "pipeline.programs", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.programs", "nscanned" : 26, "nscannedObjects" : 26, "mill
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "command", "ns" : "pipeline.$cmd", "millis" : 0 }
{ "op" : "query", "ns" : "pipeline.users", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.sessions", "nscanned" : 1, "nscannedObjects" : 1, "millis
```

Najdlhšia query 75 ms.

```
db.system.profile.find({user: "pipeline@pipeline", ts: {$gte: ISODate("2015-01-28T14:12:54.4
{ "op" : "command", "ns" : "pipeline.$cmd", "millis" : 3 }
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "query", "ns" : "pipeline.fs.chunks", "nscanned" : 1, "nscannedObjects" : 1, "milli
{ "op" : "update", "ns" : "pipeline.media", "nscanned" : 1, "nscannedObjects" : 1, "millis"
{ "op" : "query", "ns" : "pipeline.media", "nscanned" : 5, "nscannedObjects" : 3, "millis"
{ "op" : "query", "ns" : "pipeline.media", "nscanned" : 5, "nscannedObjects" : 4, "millis"
```