

KAPITOLA 11

Zálohy a zotavení

Snadno se stane, že se člověk tak soustředí na "skutečnou práci, která se má udělat", že zanedbává zálohování a nemyslí na zotavení. Ovšem to, na co se spěchá, nebývá často důležité, a to, co je opravdu důležité, to se zdá, že až tak nespěchá. Zálohy jsou důležité pro vysoký výkon i pro zotavení po nějaké katastrofě. Zálohování byste měli naplánovat a navrhnout pořádně už od samého začátku, aby nezpůsobovaly prostoje a nesnižovaly výkon.

Pokud nemáte žádný plán pro zálohy, takže je vyrábíte průběžně, až zbude nějaký čas, obvykle vytvoříte řešení, které vypadá jako ohyzná přístavba. V okamžiku, kdy je hotové, zřejmě zjistíte, že jste dříve učinili rozhodnutí, která vylučují použít ten nejlepší způsob vysoce výkonného zálohování. Například jste už přichystali nějaký server a následně zjistili, že vlastně potřebujete LVM, abyste mohli pořizovat momentky (snapshots) souborového systému – na to už je ale příliš pozdě. Také můžete přehlédnout některé důležité dopady na výkon, pokud nenakonfigurujete své systémy i s ohledem na zálohování. A pokud nebudete mít nějaký plán pro zotavovací operace, věřte nám, že určitě neproběhnou hladce, až dojde na lámání chleba a budete muset zotavovat ve stresu.

Se zálohovacími systémy je to jako s monitorovacími a signalizačními systémy: většina systémových administrátorů u nich dříve nebo později znovu vynalézá kolo. Je to ostuda, protože často je po ruce solidní, dobře podporovaný a flexibilní zálohovací software – a některý je dokonce zdarma. Rozhodně doporučujeme, abyste používali ty části těchto systémů, které můžete rozumně využít.

V této kapitole nehodláme probírat veškeré části dobře navrženého zálohovacího a zotavovacího řešení. Tato problematika je totiž natolik obsáhlá, že by snadno vystačila na celou knihu. A vskutku – několik takových knih existuje¹. Některá témata zde přeskočíme a soustředíme se na řešení pro vysoce výkonný MySQL. Na rozdíl od prvního vydání knihy zde předpokládáme, že čtenáři této knihy používají úložný engine InnoDB, buď místo MyISAM, nebo společně s ním. Tím se ovšem poněkud komplikují některé zálohovací scénáře.

¹ Myslíme si, že dobrou volbou je kniha Backup & Recovery, kterou napsal W. Curtis Preston (O'Reilly).

Přehled

Kapitolu začneme revizí několika termínů a prodiskutujeme různé záležitosti, které byste měli vést v patrnosti, až budete plánovat svá řešení pro zálohování a zotavování, včetně potenciálních nároků na to či ono. Pak předložíme přehled různých technologií a metod, jimiž se vytvářejí zálohy, a prozkoumáme techniky pro obnovu dat a zotavování po haváriích. Nakonec probereme několik vybraných dostupných zálohovacích nástrojů, a tuto kapitolu uzavřeme několika příklady, jak budovat vlastní zálohovací utility.

Terminologie

Než začneme, ujasněme si několik termínů. Zaprvé, zálohy se často v angličtině blíže specifikují přívlastky hot, warm a cold. Obvykle se jimi vyznačuje, jaký dopad bude mít zálohování. Například, při zálohování za chodu (hot backup) se nepředpokládá, že by to vyžadovalo nějaké prostoje serveru. Problém je v tom, že tyto termíny jsou vágní a pro každého znamenají něco trochu jiného. Některé nástroje mají dokonce slovo hot ve svých názvech, přičemž rozhodně nedělají to, co my považujeme za zálohování za chodu. (Občas se používá i termín dynamické zálohy; termínem cold backup se někdy chápou offline zálohy). Proto se snažíme těmto termínům vyhýbat a místo nich uvádíme, do jaké míry nějaká konkrétní technika nebo nástroj narušuje chod serveru.

Další dvě slova, která se často pletou, jsou restore (obnovit) a recover (zotavit). V této kapitole je používáme ve specifickém smyslu. Obnovit znamená získat data z nějaké zálohy – buď je ručně načíst do MySQL, nebo je umístit tam, kde MySQL očekává, že mají být. Zotavit obvykle vyjadřuje celý proces záchrany systému (nebo části nějakého systému) poté, co se přihodilo něco katastrofálního. Do tohoto procesu patří nejenom obnova dat ze záloh, ale také všechny další nezbytné kroky, které je potřeba učinit, aby byl server znovu plně funkční, například restart MySQL, změna konfigurace, zahřátí (naplnění) cache serveru atd.

Pro mnoho lidí znamená termín zotavení (recovery) pouze opravu porušených tabulek po nějaké havárii. Není to totéž co zotavení celého serveru. Zotavení úložného enginu musí dát do souladu jeho data a soubory logu. Také musí zajistit, že datové soubory budou obsahovat pouze ty modifikace, které učinily potvrzené transakce a přehraje ze souborů logu ty transakce, jež ještě nebyly aplikovány na datové soubory. Používáte-li nějaký transakční úložný engine, možná je to součást obecného zotavovacího procesu, nebo dokonce součást zálohování. Není to ovšem stejné zotavení, jaké potřebujete udělat například poté, co omylem vydáte příkaz `DROP TABLE`.

Všechno se točí kolem zotavení

Pokud všechno běží jako na drátkách, nikdy nemusíte myslet na zotavení. Ovšem v okamžiku, až budete potřebovat zotavovat, nepomůže vám ani ten nejlepší zálohovací systém na světě. Potřebujete totiž výtečný zotavovací systém.

Hlavní problém spočívá v tom, že je podstatně snadnější docílit, aby hladce fungovaly zálohovací systémy, než vybudovat dobré zotavovací procesy a nástroje. Proč? To objasňuje následující výčet:

- Především musíte mít zálohy. Nebudete schopni vůbec nic zotavit, pokud jste si předtím neudělali zálohy, takže když budujete nějaký systém, soustřeďte svou pozornost logicky v prvé řadě na zálohy. Je velmi důležité s těmito situacemi počítat, takže si nejprve naplánujte zotavení. Věřte nám – opravdu nemůžete vybudovat zálohovací systémy, dokud nebudete vědět, jaké máte požadavky na zotavení.
- Zálohování je rutinní záležitost. Tato skutečnost často zaměřuje vaši pozornost na automatizaci a doлаđování zálohovacího procesu. Ačkoliv vám pět minut přemýšlení o tom, jak přizpůsobit zálohovací proces, nepřipadá moc podstatné, zeptejte se sami sebe – věnujete každodenně stejnou pozornost i úvahám o zotavení? Zotavovací postup byste si měli procvičovat tak dlouho, dokud nebude zcela hladký a bezchybný, stejně jako zálohovací proces.
- Zálohy se obvykle nedělají pod extrémními tlaky, ovšem zotavení je typicky krizová situace. Je velmi důležité si tohle uvědomit.
- Do hry často vstupuje bezpečnost. Pokud děláte zálohy mimo web, pravděpodobně data záloh šifrujete nebo podnikáte nějaká jiná opatření, abyste je chránili. Ovšem až příliš často se stává, že se soustředíte pouze na to, jaké škody by nadělalo, kdyby se někdo k těmto datům dostal a zneužil je, přičemž úplně zapomenete na to, jaké škody nastanou, až nebude nikdo schopen odemknout zašifrovaný svazek za účelem zotavení dat nebo až budete potřebovat extrahovat jediný soubor z obrovského monolitického zašifrovaného souboru.
- Zálohy může klidně napláňovat, navrhnout a implementovat jeden člověk, zejména tehdy, pokud má k dispozici vynikající nástroje. Ovšem až dojde ke katastrofě, je možné, že tento člověk již nebude k dosažení. Je potřeba vytrénovat několik lidí a napláňovat pokrytí, aby data nemusela zotavovat nějaká nekvalifikovaná osoba.

Uvedme jeden příklad ze skutečného světa: jistý zákazník nás informoval o tom, jak se mu zálohy nebetýčně zrychlily, když s `mysql dump` uvedl volbu `-d`, a chtěl vědět, proč se nikde nepíše o tom, jak hodně může tato volba urychlit zálohovací proces. Kdyby se ovšem tento zákazník alespoň jednou pokusil o obnovu ze svých záloh, nemohl by si nevšimnout, proč je tomu tak: volba `-d` znamená, že se vůbec nedělá `dump` dat řádků! Zákazník byl zcela pohlcen zálohováním, nikoliv zotavením, takže se o této věci vůbec nedozvěděl. Když začnete myslet na zotavení, tak ještě předtím, než vůbec začnete cokoli dělat, specifikujte všechny požadavky. Například byste měli vzít v úvahu toto:

- O kolik dat můžete přijít, aniž by to mělo nějaké závažné důsledky? Budete potřebovat zotavovat ke konkrétnímu časovému bodu, nebo lze akceptovat, že přijdete o všechno, co se událo od chvíle, kdy jste pořídili poslední zálohy? Jsou nějaké požadavky z hlediska zákonů?
- Jak rychlé musí být zotavení? Jaký druh prostojů je akceptovatelný? Jaké dopady (například částečnou nedostupnost) budou akceptovat uživatelé a aplikace? Jakým způsobem zabudujete funkcionalitu, která vám umožní fungovat i ve chvíli, kdy nastanou tyto scénáře?
- Co všechno potřebujete zotavovat? Mezi běžné požadavky patří celý server, jediná databáze, jediná tabulka, nebo pouze specifické transakce nebo příkazy.

Sepište si odpovědi na tyto otázky, přidejte je do dokumentace vašeho systému, a mějte je na paměti při četbě zbytku této kapitoly. Jakmile tento úkol splníte, budete se lépe soustředit na samotné zotavení, až začnete plánovat zálohy. A když se tyto odpovědi stanou nedílnou součástí vaší dokumentace, budete je mít pohotově po ruce, až se k nim budete potřebovat později vracet.

Mýtus číslo 1 o zálohách: "Jako zálohy používám replikaci."

Chyba, se kterou se setkáváme poměrně často. Server repliky není záloha. Ani pole RAID není záloha. Abyste zjistili proč, uvažte toto: pomohou vám dostat zpět všechna data, když někdo omylem vydá příkaz DROP DATABASE na ostrou databázi? RAID ani replika vůbec neprojdou tímto primitivním testem. Nejenže to nejsou zálohy, nejsou to ani vhodné náhrady za zálohy. Potřeby ohledně záloh nesplní nic jiného než právě samotné zálohy.

Témata, která neprobíráme

Zálohování MySQL je v mnoha ohledech pouze specializovanější případ obecnějšího problému zálohování a zotavení. Ačkoliv se chceme soustředit na vysoce výkonný MySQL, bylo pro nás dost obtížné sem nezařadit materiály o spoustě dalších témat, zejména proto, že se setkáváme až s příliš velkým počtem lidí, kteří se stále potýkají se stejnými zálohovacími a zotavovacími problémy.

Tady máte seznam toho, co jsme se rozhodli sem nezařadit:

- Bezpečnost (kdo má přístup k zálohám, kdo má oprávnění obnovit data, šifrování záloh).
- Kam se mají zálohy ukládat, včetně toho, jak mají být daleko od zdroje (na jiném disku, na jiném serveru, nebo někde úplně jinde), a jak se budou data přesouvat ze zdroje na cíl.
- Retenční zásady, audit, požadavky vyplývající ze zákonů a související témata.
- Řešení úložišť a médií, komprimace a inkrementální zálohování.
- Ukládací formáty (řekneme jen tohle: vyhýbejte se proprietárním zálohovacím formátům).
- Monitorování záloh a informování o nich.
- Zálohovací schopnosti, které jsou zabudovány do úložných vrstev nebo konkrétních zařízení, jako jsou prefabrikované souborové servery.

Pokud vám tato témata nejsou důvěrně známá, měli byste si přečíst nějakou knihu o zálohování.

Celkový obraz

Než se pustíme velmi podrobně do všech možností, které jsou k dispozici, přečtěte si náš názor na to, co pravděpodobně potřebuje většina lidí pro řešení zálohovacího a zotavovacího procesu. Tato doporučení můžete chápat jako startovní čáru nebo jako směr, kterým se můžete ubírat:

- Zálohování souborů (raw backups) je u velkých databázích nezbytností. Je docela rychlé, což je velmi důležité. Ačkoliv našim favoritem jsou zálohy založené na momentkách databáze

(snapshots), jako dobrá alternativa může posloužit i nástroj ibbackup InnoDB pro zálohování za chodu, za předpokladu, že používáte pouze tabulky InnoDB.

- Zálohujte binární logy pro potřeby zotavení k danému časovému bodu.
- Udržujte několik generací záloh a udržujte binární logy dostatečně dlouho, abyste z nich mohli obnovovat.
- Testujte pravidelně zálohovací a zotavovací proces tak, že kompletně si vyzkoušíte celý proces zotavení.
- Vytvářejte pravidelně logické zálohy (kvůli efektivitě to patrně budete dělat ze zálohovaných souborů). Přesvědčte se, že máte uchováno dostatek binárních logů, abyste mohli zotavit z poslední logické zálohy.
- Pokud to jde, otestujte zálohované soubory, abyste se ujistili, že z nich bude možné zotavovat. Pokud můžete, testujte je v průběhu zálohovacího procesu, než je zkopírujete na cíl.
- Intenzivně myslte na bezpečnost. Co se stane, když někdo napadne server – dostal by se útočník i k serveru, kde jsou zálohy, nebo naopak?
- Monitorujte zálohy a zálohovací proces nezávisle na samotných zálohovacích nástrojích. Je potřeba externě ověřit, že jsou v pořádku.
- Zvolte nějaký chytrý způsob, jak se budou soubory kopírovat mezi stroji. Existují totiž efektivnější způsoby kopírování než scp nebo rsync. Více o tom si můžete přečíst v příloze A.

Proč zálohovat?

Pokud budujete nějaký vysoce výkonný systém, který se spoléhá na MySQL, je velmi důležité zálohovat. A to hned z několika následujících důvodů:

- **Zotavení po katastrofě.** Zotavení po katastrofě je to, co budete muset udělat, až vypadne hardware, až nějaká ošklivá chyba poruší data, nebo až se stanou server a jeho data nedostupnými nebo nepoužitelnými z nějakého jiného důvodu (potenciálních možných příčin je celá řada s mnoha variantami – necháme to na vaší představivosti). Šance, že dojde ke konkrétní pohromě, je velmi malá, ale pravděpodobnost, že dojde k nějaké pohromě, je už podstatně větší. Měli byste být připraveni na všechno možné, počínaje tím, že se někdo připojí omylem k nesprávnému serveru a vydá nějaký příkaz ALTER TABLE¹, přes možnost, že vyhoří budova, až k útočníkovi se zlými úmysly nebo k chybě v samotném MySQL.
- **Lidé mění své názory.** Byli byste až překvapeni, jak často se setkáváme s tím, že je potřeba zotavit alespoň některá data do přesně takového stavu, v jakém byla jistého konkrétního dne v jistý konkrétní okamžik. U některých aplikací k tomu může docházet i častěji než

¹ Baron si stále pamatuje, jak se mu tohle stalo, když pracoval jako vývojář pro jeden web elektronického obchodování. Napsal prostě příkaz do špatného okna. Na vině byl samozřejmě tým DBA – neměli dát vývojářům taková oprávnění k ostrým serverům. Rozhodně ne!

k pohromám (například tehdy, když si nějaký významný zákazník omylem smaže nějaká data a chce je nazpátek).

- **Audity.** Někdy se potřebujete dozvědět, jak data nebo schéma vypadaly v nějakém konkrétním časovém okamžiku v minulosti. Můžete se například stát účastníkem nějakého soudního řízení, nebo jste odhalili ve své aplikaci nějakou chybu a potřebujete zjistit, co všechno tehdy kód dělal (to, že máte v kódu v nějakém nástroji pro řízení verzí, nemusí stačit).
- **Testování.** Jedním z nejjednodušších způsobů, jak testovat na realistických datech, je periodicky kopírovat na testovací server nejnovější ostrá data. Jestliže si děláte zálohy, jednoduše to dělejte pomocí záloh.

Kontrolujte své předpoklady. Předpokládáte například, že poskytovatel sdíleného hostingu zálohuje MySQL server poskytovaný k vašemu účtu? Přestože sdílený hosting ve skutečnosti není relevantní pro vysoký výkon, chceme zdůraznit, že takové předpoklady se mohou pěkně vymstít. (Mnozí webhosteři totiž MySQL servery vůbec nezalohují, a jiní dělají pouze kopie souborů, zatímco server běží, takže pravděpodobně vznikne porušená záloha, která bude nepoužitelná.)

Úvahy a kompromisy

Zálohování MySQL je těžší, než jak vypadá. Na té nejzákladnější úrovni je sice záloha pouze kopie dat, nicméně pořízení této kopie mohou ztěžovat potřeby aplikace, architektura úložného enginu MySQL i systémová konfigurace.

O co si můžete dovolit přijít?

Při vytváření strategie pro zálohování byste měli vycházet z toho, o kolik dat si můžete dovolit přijít. Budete potřebovat takové vybavení, abyste byli schopni zotavit k danému časovému bodu, nebo bude stačit, když zotavíte ze záloh pořízených minulou noc, takže přijdete o veškerou práci, která se od té doby udělala? Pokud potřebujete zotavovat k danému časovému bodu, postačí, když budete zálohovat pravidelně a ujistíte se, že je zapnuto zaznamenávání do binárního logu. Pak budete schopni obnovit ze zálohy a zotavit k danému časovému bodu tak, že přehrajete binární log.

Všeobecně řečeno: čím více dat můžete po případné havárii postrádat, tím snadnější bude zálohování. Pokud ovšem máte velmi striktní nároky, je podstatně obtížnější zajistit, aby se dalo zotavit opravdu všechno. Existují dokonce i různé varianty zotavení k danému časovému bodu. "Mírný" požadavek na zotavení k danému časovému bodu znamená, že s největší pravděpodobností budete schopni opětovně vytvořit data tak, že budou "dostatečně blízko" k místu, kde se vyskytl problém. Striktní požadavek na zotavení znamená, že nikdy nebudete moci tolerovat ztrátu potvrzené transakce, i kdyby se stalo něco opravdu hrozného (například to, že shoří celý server). Uvědomte si ovšem, že něco takového vyžaduje použití speciálních technik, například udržování binárního logu na samostatném svazku SAN, nebo používání diskové replikace DRBD. O těchto přístupech si můžete více přečíst v kapitole 9.