MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

Návrh a implementace rozhraní pro monitorování komponenty Engine systému Perun

BAKALÁŘSKÁ PRÁCE

Jana Čecháčková

Brno, jaro 2014

Zadání:

Engine je komponenta systému Perun, která slouží pro generování konfiguračních souborů a jejich následné rozesílání koncovým službám. Engine funguje masivně paralelně, špičkovou zátěž rozkládá v čase pomocí interních front a dalších vnitřních stavů.

Úkolem práce bude seznámit se s Engine a následně navrhnout a implementovat rozhraní pro monitorování aktuálního vytížení a vnitřních procesů Engine. Rozhraní musí poskytovat nejen statistické údaje, ale musí umožňovat přístup i k detailům konkrétních úloh, které jsou právě zpracovávány nebo čekají ve frontách. Dále toto rozhraní musí umožňovat sběr statistických dat o průběhu zpracovávaných úloh, například délky běhů a počty dokončených úloh v čase.

Osnova

Úvod

Představení systému Perun

- Virtuální organizace
- Správa identit uživatelů
- Správa služeb
    - Facility
    - Resource
- Hlavní složky systému Perun

Představení Perun Engine

Model Perun Engine

Task Executor

Návrh řešení

Implementace

Úvod

Představení systému Perun

Systém Perun vznikl jako projekt sdružení CESNET[1] a jedná se o systém spravující identity uživatelů a přístup ke službám.

Původní motivací pro vývoj systému Perun bylo vytvořit systém, který bude schopný řídit uživatele a služby aktivity MetaCentrum[2]. Metacentrum je česká národní gridová infrastruktura, která má výpočetní a úložné zdroje rozprostřeny na místech, kde jsou uživatelé, administrátoři a uživatelská podpora z různých organizací. Metacentrum poskytuje ve svých službách téměř 10 000 CPU jader a 3.5 PB místa k uložení dat. Tyto služby využívá okolo 700 uživatelů z akademické půdy. MetaCentrum potřebovalo ke své činnosti zjednodušení řízení uživatelů a služeb. Bylo potřeba zajistit automatické vytvoření uživatelských účtů na všech strojích a také jejích pozdější automatickou expiraci - tyto úkony nebylo možné z důvodu velkého počtu uživatelů i služeb provádět manuálně. Za tímto účelem vznikl systém Perun.

Perun podporuje správu uživatelů, delegování práv přístupu, řízení skupin a zápisu členů za účelem zjednodušit správu uživatelů. Perun je nyní vyvíjen už ve své třetí verzi a jeho funkcionalita již vzrostla nad rámec aktivity MetaCentrum. V nynější podobě je spravován sdružením CESNET a oproti předchozím verzím nabízí správu virtuálních organizací.

Virtuální organizace

Virtuální organizace je jednoduchá skupina skládající se z uživatelů, definovaného správce a souborem pravidel, které definují, kdo se může stát členem této virtuální organizace. Výhodou virtuální organizace je, že pokud chtějí její členové používat určitě služby, správce vyjedná přístup ke službám s poskytovatelem pouze jednou tzn. správce udělá práci za všechny členy jeho virtuální organizace (bez virtuální organizace by si každý uživatel musel vyjednat přístup ke službě sám).

Perun může spravovat neomezené množství virtuálních organizací, které jsou složeny z tisíců členů a služeb. Uživatelé jsou zapisováni do virtuálních organizací, kde mohou být dodatečně organizováni do skupin a podskupin. Každá skupina má definovaného svého správce, který spravuje členství ve skupině.

1

2

Správa identit uživatelů

Zvyšující se počet služeb, které jsou využívány výzkumnými pracovníky, vyžaduje nějaký typ autentizace a autorizace. Důvodů může být několik např. služby mohou být zpoplatněné, mohou reprezentovat jedinečné a důležité zařízení nebo nemohou být využívány bez přístupu a poskytovatelé těchto služeb musí vědět, kdo k těmto službám přistupuje i v případě, že služba není zpoplatněna.

Obecně je k identifikaci uživatelů používán Identity Management System[3] a ten může být realizován v domovské instituci uživatele nebo poskytován třetí stranou (např. Sociální sítě). Tento typ identifikace ale nemusí být vždy dostatečný, protože poskytovatelé služeb musí v tomto případě vyslovit určitou důvěru k třetím stranám, že tyto identity dostatečně prověřily. To v některých případech z důvodu „peer to peer“[4] důvěry a otázky ochrany soukromí možné. Systém správy identit a přístupu - Perun, řeší tento problém řízení přístupových práv a identit.

Systém Perun v sobě zahrnuje celý cyklus uživatele, od jeho zápisu, přes správu přístupových práv až k expiraci uživatelského účtu. Perun nepracuje pouze s uživatelskými identitami, je schopný uchovat další informace o uživatelích, organizovat uživatele do skupin a virtuálních organizací. V neposlední řadě mohou být tyto skupiny přiřazeny ke službám tzn. že členové této skupiny mají nastaveno právo pro používání služby. To je hlavní výhoda ve srovnání s klasickými IdM systémy.

Ve většině institucí nebo výzkumných skupin již nějaká správa uživatelů existuje, stejně tak i správa služeb. Tato správa ale většinou nesplňuje všechny požadavky, které jsou kladeny. Není robustní, neposkytuje programovatelné prostředí a postrádá přívětivé prostředí pro uživatele.
Perun je vytvořen i k nasazení do existujících infrastruktur, kde přináší robustní a škálovatelné řešení.

V porovnání s běžnými systémy spravující identity, Perun nabízí také správu služeb a přístupu. Perun je komplexní nástroj, který zjednodušuje správu výzkumných spolků nebo uživatelů a služeb napříč organizacemi. Systém Perun je používán na národní i mezinárodní úrovni, je dnes reálně využíván několika organizacemi, ke kterým se řadí MetaCentrum, C4E nebo Fedcloud. Budoucí potencionál systému Perun by mohl spočívat např. i v nasazení do sítě Eduroam.

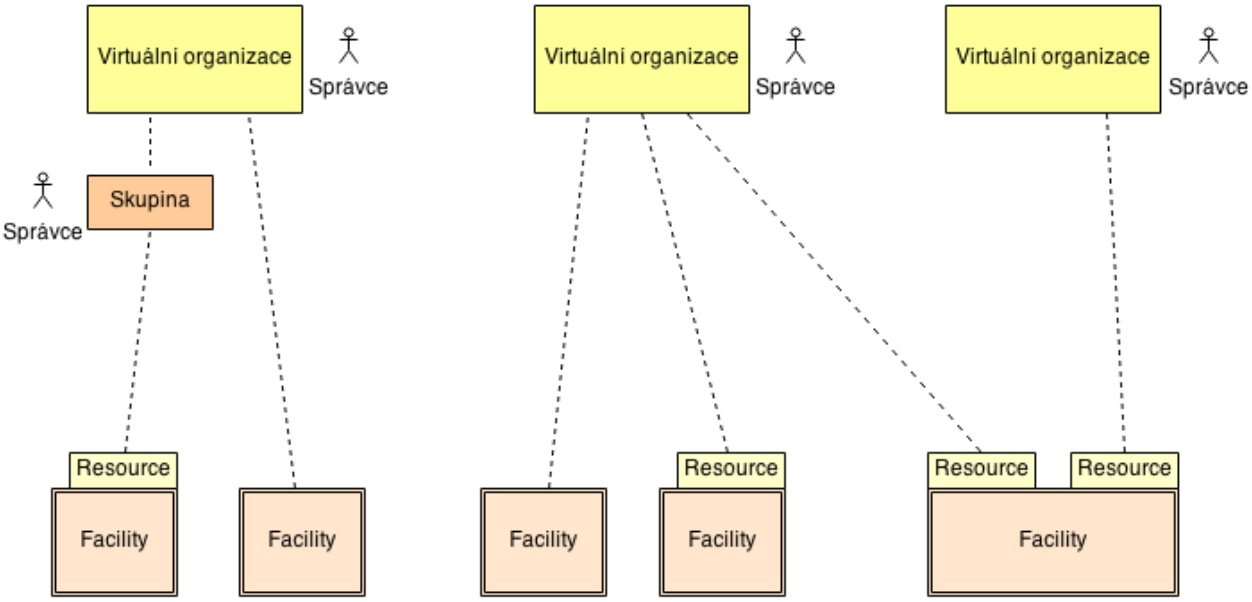
3

4

Perun je dobře uzpůsoben také pro organizace nebo výzkumné skupiny, které chtějí řídit přístup ke svým službám a nemají žádný systém správy identit nebo jich mají několik odlišných systémů a chtějí z nich získat a propojit uživatelské identity. Perun podporuje i složitější nasazení, jako je sdílení služeb mezi několika institucemi.

Správa služeb

Vytvořit řízení služeb, které bude efektivní, vyžaduje účast jak poskytovatelů služeb, tak i správců virtuálních organizací. Poskytovatelé služeb vyžadují jednoduchý způsob, jakým udělat své služby dostupné pro virtuální organizace. Navíc chtějí, aby byla provedena minimální nebo žádná změna na jejich službách a také požadují plnou kontrolu nad celým konfiguračním procesem. Na druhou stranu správci virtuálních organizací potřebují poskytnout uživatelům využití zdrojů. Proto byla vytvořena základní jednotka pro management zdrojů, která se nazývá facility.



Facility

Facility je homogenní entita, která poskytuje služby. Může představovat libovolnou službu, např. datové úložiště, tiskárnu, učebnu atd. Jediná podmínka pro facility je, že nastavení zůstane pro celou tuto jednotku neměnné. Poskytovatel služeb provádí konfiguraci této facility a po dokončení k ní poskytuje přístup virtuálním organizacím. Pokud spolu uzavřou poskytovatel služeb a správce virtuální organizace dohodu o používání služeb, měla by tato dohoda zahrnovat podmínky, pod kterými budou členové virtuální organizace tuto službu využívat. Poskytovatel služeb může pro virtuální organizace vytvořit tzv. Resource.

Resource

Resource definuje technické podmínky a omezení používání facility virtuálními organizacemi. Správce virtuální organizace se poté může rozhodnout, kteří členové z virtuální organizace mohou používat resource nebo mohou tímto právem pověřit některého ze správců skupin.

Základní komponenty systému Perun

Perun je skládá z několika důležitých komponent, které mají přesně definovanou funkcionalitu. Mezi ně patří jádro, RCP, Registrar a Dispatcher s Engine. Jádro Perunu má na starost data a operace s uživateli, virtuálními organizacemi, službami a zdroji. Komponenta RPC je hlavní programovatelné prostředí systému Perun. RPC zprostředkovává komunikaci ostatním komponentám nebo i externím systémům se systémem Perun. Registrar je komponenta určená k zápisu uživatele a správě registračních formulářů. Dispatcher a Engine jsou zodpovědni za distribuování seznamů přístupů a konfigurací dále konečným službám.

Představení Perun Engine

Perun Engine je součást systému Perun, která zpracovává přijaté události a propaguje nový stav do vybraných destinací. Pod událostmi si můžeme představit např. přidání nového uživatele do virtuální organizace, zahájení využívání služby nějakou virtuální organizací, atd. Perun Engine zodpovídá za to, aby byly všechny změny propagovány na samotné služby. Události do Perun Engine zasílá komponenta systému Perun s názvem Perun Dispatcher.

Perun Dispatcher

Tato komponenta je důležitou součástí systému Perun, která zpracovává databázová data a vytváří z nich události. Pokud je rozpoznána událost, která je spojena s nějakou ze služeb, je tato událost zaslána do Engine.

Událostí může být v jednom čase i velké množství a mohou způsobit velké vytížení Engine, což není žádoucí. Proto je další funkcí Perun Dispatcher vhodně události rozdělovat mezi několik instancí Engine. Rozdělení může být provedeno podle různých kriterií, např. geografické vzdálenosti strojů nebo podle toho, který Engine momentálně žádné události nezpracovává.

V současné době je pro zpracování událostí využíván Engine pouze jeden, v budoucnu se předpokládá rozšíření systému Perun a nasazení více instancí Engine.

Zpracování událostí

Perun Engine zodpovídá za zpracování událostí, které získá. Událost má při příchodu do Engine podobu textového řetězce a obsahuje strukturované informace o změnách, které se mají provést. Engine pomocí parsování této zprávy získává informace o změnách, které má za úkol propagovat.

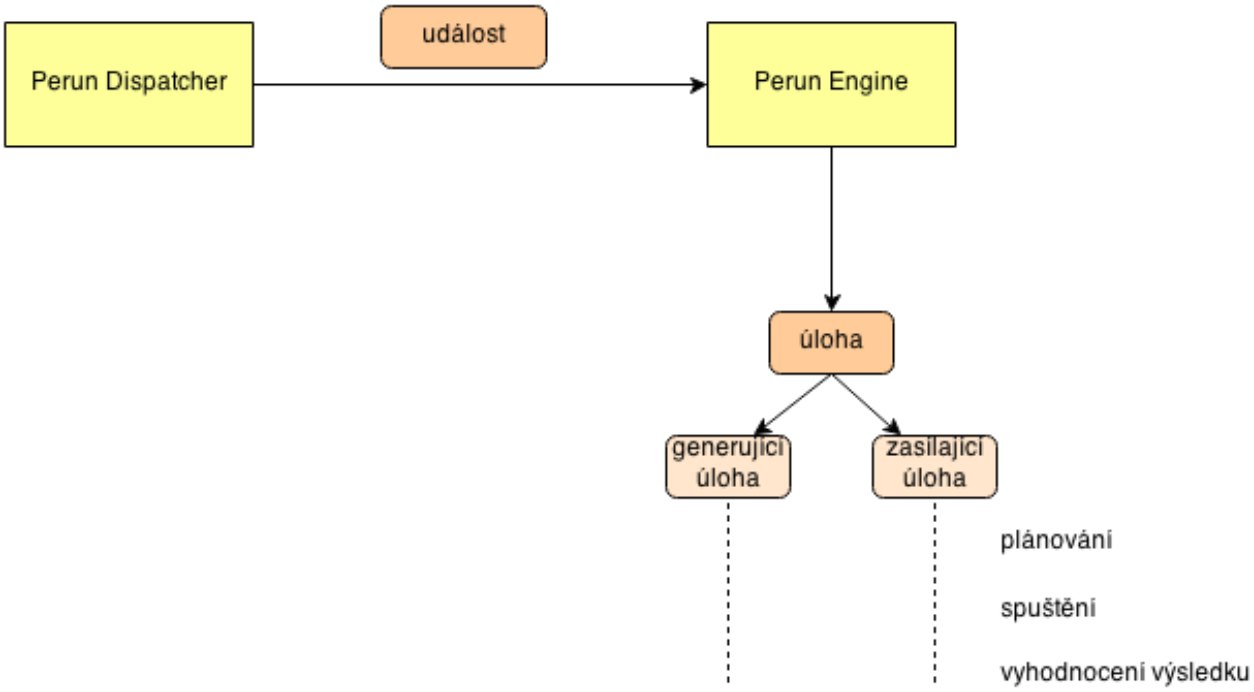
Engine je při tomto zpracování schopen vyhledat v událostech duplicity. Uveďme si příklad vzniku duplicity: Do virtuální organizace, která využívá nějakou službu, se přidají dva noví uživatelé. Perun Dispatcher zprávy o těchto změnách zašle do Engine a ten zaregistruje, že se jedná o stejný typ změny, který má být propagován na stejnou službu. Dále tedy zpracovává tyto dvě události jako jednu úlohu.

Každá úloha se po přijetí do Perun Engine rozdělí na dvě podúlohy, a to generující úlohu a zasílající úlohu. Generující úloha slouží k vygenerování skriptu, který bude spuštěn na cílovém stroji. Zasílající úloha pak slouží k přenesení tohoto skriptu k samotné službě. Je zřejmé, že zasílající úloha může plnit svou funkci jen za předpokladu, že generující úloha už daný skript vytvořila. Tuto závislost je nutné při plánování úloh ke spuštění vzít v úvahu.

Plánování a spouštění úloh

Všechny získané úlohy se v Engine nejprve naplánují ke spuštění. Při plánování úloh se zjišťuje, zda není daná propagace změny na některé ze služeb zakázána – v tom případě není možné propagaci změny provést. Také je nutné zmapovat strom závislostí – pokud Engine zjistí, že je úloha závislá na některé z jiných úloh, které dosud nebyly spuštěny, musí je naplánovat a spustit přednostně.

Po těchto procesech jsou úlohy připravené k samotnému odeslání a následnému spuštění na jednotlivých strojích. Úlohy jsou spouštěny paralelně, tzn. Engine jich spouští několik najednou. Protože množství naplánovaných úloh může být opravdu velké, omezuje Engine spouštění úloh pouze do určitého limitu – zbývající úlohy čekají na spuštění ve frontě na uvolnění místa. Jakmile jsou všechny úlohy, které bylo potřeba spustit, dokončeny, je provedena kontrola o úspěšnosti úloh – zda jejich spuštění proběhlo bez problémů či nikoliv. Pokud je nalezena úloha, při které vznikly v průběhu jejího spuštění problémy, Engine tuto úlohu naplánuje na opětovné spuštění.



Pokud Engine obslouží události, které mu Dispatcher zaslal, tzn. úlohy naplánuje, spustí a počká na jejich dokončení, komunikuje s další komponentou systému Perun a to Perun Controller. Engine informuje Controllera o dokončení jednotlivých úloh.

Perun Controller

Perun Controller komunikuje s komponentami Perun Dispatcher a Perun Engine skrz databázi a koordinuje jejich správu propagací. Jedná se o knihovnu, která také umožňuje přístup k těmto komponentám. Controller je navržen tak, aby poskytl prostředky pro plnění administrativních úloh a dohlížel na aktuální statistiky.