MetaCentrum - Virtualizace a její použití

Miroslav Ruda, . . .

Cesnet

Brno, 2009

Obsah

- Motivace
  - co je virtualizace
  - kde ji lze využít
- Stávající využití na výpočetních uzlech
  - přepínání různých instalací na jednom stroji
  - preempce (pozastavení výpočtu)
  - služební domény
- Nová služba - virtuální clustery
  - úloha ve svém obraze OS
  - cluster z virtuálních strojů

Co je virtualizace

- Iluze celého počítače
  - pomocí softwarové implementace
  - s pomocí moderního hardware (Intel/AMD rozšíření)
- Posun izolace úloh od procesů k celým OS
- Režie nasazení se minimalizuje
  - OK na malých výpočetních uzlech
  - problémy na větších NUMA strojích (manwe,eru)
  - potenciální problémy s I/O
    - u nás OK Infiniband, rozumně ethernet
    - OK zvláštní diskové oddíly
- Více virtuálních strojů na jednom fyzickém
  - sdílení/rozdělení paměti, procesorů

Virtualizace – využití

Virtualizace výpočetních uzlů:

- možnost provozovat aplikace s různými požadavky na OS na stejném fyzickém stroji
- přidělování zdrojů (CPU, paměť) virtuálním uzlům
- pozastavení, checkpointing, migrace virtuálního stroje
  - = "svatý grál superpočítání a gridů"
- izolace jednotlivých výpočtů
  - požadavek aplikace
  - požadavek správců

Virtualizace služebních strojů, webhosting

- není obsahem této prezentace

Technická vsuvka č. 1 – virtualizační nástroje

Používáme virtualizační nástroje

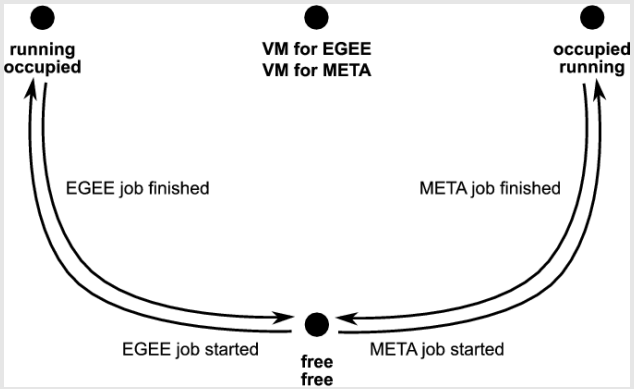
- Xen – iluze celého počítače, paravirtualizace
- Vserver – jediný systém, různá aplikační prostředí
- zvažujeme i podporu KVM a LXC

Na správu virtuálních strojů jsme vyvinuli nástroj Magrathea:

- umožňuje přepínání aktivních domén, jejich správu
- dynamická správa zdrojů (CPU, paměť)
- zjednodušený pohled pro zbytek světa (PBS, uživatelé)
  - stav virtuálního stroje

Přepínání stavu v nejjednodušším případě (dvě alternativní domény):

- omezení na pouze dvě domény jen dočasné



Současné využití na výpočetních uzlech I.

Dva staticky instalované virtuální stroje

- celý stroj přidělený aktivnímu virtuálnímu stroji
- v každém okamžiku nanejvýše jeden virtuální stroj s aplikací
- v každém instalovaná jiná distribuce Linuxu
- dynamické přepínání podle aktuální potřeby
  - různé fronty pro různé typy virtuálních strojů
  - plánování úloh v PBS rozhoduje který virtuální stroj dostane zdroje
- přepínání MetaCentrum / EGEE
- použitelné i pro dvě nezávislé instalace PBS

První prostor na dotazy

Současné využití na výpočetních uzlech I.

Dva staticky instalované virtuální stroje

- celý stroj přidělený aktivnímu virtuálnímu stroji
- v každém okamžiku nanejvýše jeden virtuální stroj s aplikací
- v každém instalovaná jiná distribuce Linuxu
- dynamické přepínání podle aktuální potřeby
  - různé fronty pro různé typy virtuálních strojů
  - plánování úloh v PBS rozhoduje který virtuální stroj dostane zdroje
- přepínání MetaCentrum / EGEE
- použitelné i pro dvě nezávislé instalace PBS

První prostor na dotazy

Současné využití na výpočetních uzlech II.

Preempce (pozastavení běžící úlohy)

- standardní úlohy v první doméně
- úloha s vyšší prioritou spuštěna ve druhé doméně
  - první doména pozastavena
- v současnosti pozastavujeme jen sekvenční úlohy
- vyšší priorita vlastníkům
  - cluster Orca
- velké paralelní úlohy
  - jinak by dlouho blokovaly zdroje při čekání na dostatečný počet uzlů

Domény dedikované pro "služby"

- provozované dlouhodobě
- pozastavené v době kdy nejsou aktivní
- znovu aktivovány podle požadavků uživatele

Novinka - virtuální clustery

Nové požadavky motivované "cloud přístupem"

- jednotlivé úlohy spuštěny každá ve "svém" prostředí
  - obrazy OS podporované MetaCentrem nebo uživatelské
- postavení semi-permanentního clusteru z virtuálních strojů
  - uživatelé si mohou sami spravovat uzly, úlohy...

Nadále plně zapojené do standardního prostředí, úlohy i clustery

- zadávané přes stejné rozhraní
- normálně soupeří o zdroje
- stejné plánovaní, stejná pravidla, stejná omezení

Součástí může být i privátní síť (VLAN)

- přidělovaná jako každý jiný zdroj plánovačem

Technická vsuvka č. 2 - obrazy virtuálních strojů

Obraz celé instalace operačního systému:

- instaluje se pomocí překopírování z "repository"
  - databáze obrazů, spolu s popisem (metadata)
    - metadata používá plánovač pro svoje rozhodování
    - kdo je vlastník, kdo může používat
    - jaké vlastnosti obraz poskytuje (debian, suse, ncbr)
    - jaké požadavky obraz má (hardware, prostor na disku)
- podporované MetaCentrem nebo uživatelské
  - MetaCentrový debian, vývojový obraz, SLC5 pro EGEE
  - uživatelské
    - prostředí vycházející z MetaCentrového obrazu
    - úplně vlastní
    - RedHat/Suse pro komerční aplikace
    - teoreticky i MS Windows

Úloha ve vlastním prostředí

Úloha s požadavkem na vlastní prostředí

- uzly nejsou předinstalované, instalují se online podle požadavků
- po skončení úlohy jsou z obrazu uloženy logy a scratch
- jinak se chovají a plánují jako normální úlohy
- ideálně `qsub -l nodes=1:muj_debian`
  - teď jen ve spojení s postavením clusteru
  - jak zajímavé jako samostatná služba?

Obrazy OS dodané uživatelem

- podporujeme i obrazy "do kterých nevidíme"
  - žádné změny do instalace
- bezpečnostní implikace
  - uživatel má rootovské oprávnění
  - nemáme záruku že jsou aplikovány bezpečnostní patche
- zavřené do privátní sítě

Úloha ve vlastním prostředí

Úloha s požadavkem na vlastní prostředí

- uzly nejsou předinstalované, instalují se online podle požadavků
- po skončení úlohy jsou z obrazu uloženy logy a scratch
- jinak se chovají a plánují jako normální úlohy
- ideálně `qsub -l nodes=1:muj_debian`
  - teď jen ve spojení s postavením clusteru
  - jak zajímavé jako samostatná služba?

Obrazy OS dodané uživatelem

- podporujeme i obrazy "do kterých nevidíme"
  - žádné změny do instalace
- bezpečnostní implikace
  - uživatel má rootovské oprávnění
  - nemáme záruku že jsou aplikovány bezpečnostní patche
- zavřené do privátní sítě

Virtuální clustery

Cluster z virtuálních strojů

- `qsub -l cluster=JMENO -l nodes=2:debian+4:slc5`
- plánovaní analogické paralelním úlohám
- uživatel může po spuštění
  - použít ssh pro přímý přístup na stroje
  - provozovat uvnitř cluster svoje nástroje pro správu
  - použít centrální PBS pro spouštění úloh do clusteru (jen náš obraz)
- použitelné s naším nebo uživatelským obrazem

Autorizace (plán do konce roku)

- možnost rebootovat vlastní virtuální stroj
- u obrazu definovat skupinu, která smí obraz také používat
- u běžícího clusteru definovat skupinu, která do clusteru také smí zadávat úlohy

Clustery v privátní síti

Nutné pro privátní obrazy, možno i u dalších obrazů

- `qsub -l cluster=NAME,net=private`
- přidá se jeden servisní uzel, běžící DHCP a VPN servery
  - plán - DHCP konfigurace podle uživatelova nastavení
  - autorizace na VPN server pomocí certifikátu
- standardní openvpn klient, umožňuje
  - klasický NAT
  - virtuální cluster připojený přímo do sítě vlastníka
    - vlastní správa adresního prostoru a síťových politik
    - pro koncového uživatele nerozlišitelné od "katedrálního" clusteru
    - snížení vstupního prahu, zdroje MetaCentra zabalené v plně "místním" pojetí
- díky službám NREN CESNET2 můžeme stavět VLAN přes ČR bez významné režie
- vyvinuli jsme službu SBF pro správu VLAN
  - jednoduché rozhraní, přímo integrováno do PBS

Současný stav

- prototyp nasazen v experimentálním prostředí (arien)
- odladěný a dostupný jen MetaCentrový obraz
  - a jeho modifikovaná varianta pro privátní síť
- teď je čas přijít s vlastním obrazem
- autorizace není implementována, musí být do konce roku
- privátní clustery mají staticky konfigurované DHCP
- omezená množina služeb dostupných v privátní síti (AFS, NFS)
- hledáme odvážné testery, sbíráme požadavky na rozšíření...

Příklady – stav uzlů před spuštěním

```
ruda@vilya:~$ pbsnodes -a
skirit82.ics.muni.cz
     Host = skirit82.ics.muni.cz
     ntype = cloud
     state = free
...
skirit82-1.ics.muni.cz
     Host = skirit82-1.ics.muni.cz
     ntype = virt
     state = free
     license = u
...
skirit82-2.ics.muni.cz
     Host = skirit82-2.ics.muni.cz
     ntype = virt
     state = down
```

Příklady – spuštění clusteru

```
vilya:~$ cluster_submit -N ruda_cluster -l 2:debianX
485.vilya.ics.muni.cz
vilya:~$ cluster_status ruda_cluster
Cluster records:
  Cluster name: ruda_cluster
  Record ID:    1257935292
  Attributes:   owner=ruda@vilya.ics.muni.cz
  Job ID:       485.vilya
  Owner:        ruda@vilya.ics.muni.cz
  Machines:     skirit82-2.ics.muni.cz skirit83-2.ics.
  Username:     ruda
  Queue:        default
  Elapsed:      720:0
  State:        T
  Time:         -
ruda@vilya:~$
```

Příklady – stav clusteru

```
ruda@vilya:~$ qstat 485.vilya
Job id      Name           User   Time Use S Queue
------- ---------- ------------   ----
485.vilya  ruda_cluster  ruda  00:00:00 R default
ruda@vilya:~$ cluster_status ruda_cluster
Cluster records:
  Cluster name: ruda_cluster
  Record ID:    1257935292
  Attributes:   owner=ruda@vilya.ics.muni.cz
  Job ID:       485.vilya
  Owner:        ruda@vilya.ics.muni.cz
  Machines:     skirit82-2.ics.muni.cz skirit83-2.ics.
  Username:     ruda
  Elapsed:      720:0
  State:        R
  Time:         00:04
```

Příklady – stav uzlu po spuštění

```
ruda@vilya:~$ pbsnodes skirit82-2.ics.muni.cz
skirit82-2.ics.muni.cz
     Host = skirit82-2.ics.muni.cz
     ntype = virt
     state = free,cloud
     license = u
     pcpus = 4
     properties = virtual,brno,vi822
     added_properties = debianX
     resources_available.arch = linux
     resources_available.mem = 3379452kb
     resources_available.ncpus = 4
...
```

Příklady – spuštění úlohy, ukončení clusteru

```
$ qsub -I -l nodes=1:ppn=1 -l  cluster=ruda_cluster
qsub: waiting for job 486.vilya.ics.muni.cz to start
qsub: job 486.vilya.ics.muni.cz ready
...

ruda@vilya:~$ cluster_delete ruda_cluster
Deleting cluster ruda_cluster (Job ID 485.vilya)


ruda@vilya:~$ qstat
Job id     Name          User  Time Use S Queue
------- -------  --------- - ---
485.vilya  ruda_cluster ruda  00:00:00 C default
```