MOSIX

Jan Outrata

24. února 2002

1 Co je to cluster

Pro provoz aplikace náročné na výpočetní výkon (např. SETI@Home, lámání hesel) bývá oproti pořízení drahého multiprocesorového superpočítače výhodnější sestavit z obyčejných počítačů **výpočetní cluster**. Cluster je taková konfigurace několika počítačů, která se navenek tváří a chová jako jediný stroj. Clustery lze rozdělit na dva typy:

- high-availability (vysoce dostupné) - používají se pro nepřetržitý provoz určité služby bez ohledu na možné HW či SW problémy, které mohou potkat jednotlivé stroje
- computational (výpočetní) - naznačeny v úvodu; spojením slabších levných strojů lze získat slušný výpočetní výkon

Na clusterech se provozují úlohy, které se dají dobře paralelizovat (tj. rozdělit na spoustu menších podobných pod-úloh a tyto zpracovávat samostatně a nezávisle jednu na druhé). Dílčí pod-úlohy se zpracovávají na uzlech clusteru (nodes). Všechny se mohou zadat jen jednomu uzlu (master nebo home node), tento je pak rozdistribuuje (migruje) ostatním uzlům. V praxi se můžeme setkat s problémy rozdílné rychlosti uzlů (různě výkonné stroje), rozdílné zátěže uzlů (více úloh než uzlů), rozdílné rychlosti komunikace mezi uzly (vyvstává potřeba analyzovat na který uzel jakou úlohu) a výpadku uzlu (úloha se musí předat jinému uzlu). Výpočetní clustery se dělí na dva typy:

- na úrovni aplikace - aplikace pomocí speciálních funkcí (např. knihovna PVM - Parallel Virtual Machine nebo MPI - Message Passing Interface) bude sama řídit distribuci úloh a komunikaci uzlů
- na úrovni operačního systému - aplikace se nebude o nic starat (jen bude paralelizovaná do více procesů), nejvhodnější rozmístění procesů zajistí OS, např. Linux s podporou MOSIXu.

2 Co je to MOSIX

MOSIX (Multicomputer Operating System for UnIX, `http://www.mosix.org`) je softwarový projekt Prof. Amnona Baraka a studentů Institutu informatiky Hebrejské univerzity v Jeruzalémě z roku 1977 (1988 MOSIX) pro vytvoření škálovatelných výpočetních clusterů. Existují dvě verze: **K-MOSIX** - patch do linuxového jádra a **U-MOSIX** - uživatelský balík programů pro různé UNIXové platformy. Obě verze mají za cíl zajistit rovnoměrnou distribuci práce a vyváženou zátěž uzlů v clusteru pro dosažení maximálního výkonu clusteru při co největší jednoduchosti použití. Pro nově vytvořený proces MOSIX určí nejvhodnější uzel pro běh tohoto procesu. Algoritmy MOSIXu jsou decentralizované, tzn. každý uzel může vykonávat lokální procesy stejně jako i procesy, které mu byly přiděleny migrací z jiného uzlu.

K-MOSIX vyvažuje zátěž uzlů pomocí transparentní migrace již běžících procesů, existuje zde ale nemalá režie migrace a těsná vazba na jádro systému.

U-MOSIX distribuuje procesy ihned po jejich vytvoření a je proto vhodnější pro aplikace s velkým počtem krátkých procesů a velkými daty v heterogenních clusterech s různými verzemi UNIXů.

MOSIX podporuje různě velké konfigurace clusterů, od pár PC v Ethernetu až po desítky multiprocesorových serverů ve vysokorychlostní síti.

3 Instalace a konfigurace clusteru

Protože zatím existují jen implementace K-MOSIXu pro linuxové jádro, podíváme se zde na použití této verze MOSIXu.

Pro vytvoření výpočetního clusteru potřebujeme síť počítačů s Linuxem. Na každém uzlu musí být stejně nakonfigurované jádro s patřičnou verzí MOSIXu pro toto jádro. Při konfiguraci jádra se konfiguruje též MOSIX, např. jestli vůbec chceme umožnit migraci mezi uzly, zda máme složitější síť (více switchů, různý HW), možnost debugování, podpora informací o uzlech, společný sdílený souborový systém MFS - MOSIX File System (jinak se musí IO operace přesměrovávat na home node, což odrazuje od migrací). Pak ještě nainstalujeme utility a dokumentaci MOSIXu a tím je instalace dokončena.

Konfigurace se provede editací konfiguračního souboru, ve kterém specifikujeme skupiny počítačů a adresy a počet uzlů v každé skupině. Soubor musí být na všech uzlech stejný.

4 Spuštění clusteru a aplikací

Celý cluster se spouští a ukončuje jediným programem `setpe`. Po spuštění můžeme přimountovat MFS a máme běžící cluster.

Úlohy se spouštějí jako obvykle, při velké zátěži jednoho uzlu se úlohy transparentně přesunou na jiný volný nebo méně zatížený uzel. Zátěž jednotlivých uzlů (procesor, paměť) můžeme monitorovat pomocí programu `mon`. Pomocí programu `mosrun` můžeme úlohu spustit tak, že specifikujeme např. že má úloha zůstat na home (nebo i jiném) node a nikam nemigrovat, že může migrovat jen mezi určitými uzly, jaký je to druh procesu (výpočetní nebo IO orientovaný).

5 Administrace clusteru

Cluster lze samozřejmě také administrovat a to pomocí programu `mosctl`. Např. lze uzlům zakázat či povolit přijímání nebo odmigrování úloh, nastavovat parametry rozhodování o migraci úlohy, zjišťovat informace o uzlech (statistiky). Administraci lze učinit uživatelsky přívětivější pomocí programu `MosixView`. K ladění clusteru na co nejvyšší výkon slouží utilitka `tune`, pomocí které lze měřit různé parametry uzlu (procesor, paměť, připojení k síti) i celé sítě (rychlost, propustnost) a určit tak ideální parametry rozhodování o migrování úloh.

6 Příklad použití

Příkladem použití a zároveň demonstrace výkonu MOSIXu (a clusterů vůbec) je např. projekt kvantové simulace velkých biologických molekul (např. proteinů). Algoritmy počítají multidimenzionální kvantové vlnové funkce pomocí Schrödingerových rovnic a jsou velice náročné na výpočetní výkon. Následující tabulka udává čas, který algoritmus spotřebuje, a zrychlení výpočtu v závislosti na počtu uzlů Pentium-Pro 200MHz v clusteru na 100Mb/s síti.

počet uzlů	1	2	4	8	16
čas [s]	760	400	230	117	70
zrychlení	1	1,9	3,3	6,5	10,8