

Párhuzamos és Grid rendszerek (3. ea)

cluster rendszerek, hosszútávú ütemezők

Szeberényi Imre
BME IIT

<szebi@iit.bme.hu>



Hol tartunk ?

- Megismerkedtünk az alapfogalmakkal, architektúrákkal.
- Egyszerű absztrakciós modellt alkottunk a párhuzamos gépek leírására.
- Megismertük a párhuzamos programok tervezésének egy módszerét (PCAM).

Klaszter

- Párhuzamos rendszerek fejlődésének egyik fontos állomása, amit ma több gyártó ismét elővett.
 - közös állományrendszer
 - laza → szoros csatolás
 - batch feldolgozás
 - hosszútávú ütemezés

Klaszterek története

- Kezdetek: szg. hálózatok megjelenése – 60-as évek vége 70-es évek eleje.
- Igazi fejlődés a 70-es évek vége, 80-as évek eleje. (DEC, VAXcluster)
 - elosztott, párhuzamos számítás
 - megosztott fájlrendszer
 - megosztott perifériák

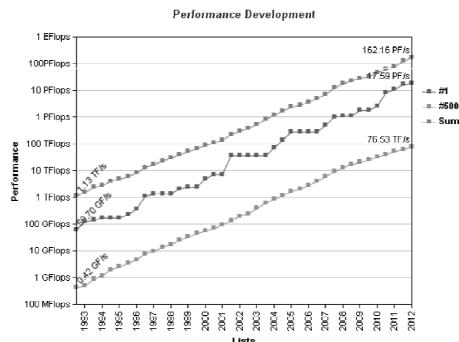
Klaszterek ma

- Nagy rendelkezésre állást biztosító klaszter
- Terheléskiegyenlítő klaszter
- Számítási klaszter
 - házi: beowulf
 - gyártóktól: TOP500
- Grid klaszter
 - grid site-ok
- Elosztott + redundáns adattárolás, big data
 - Hadoop

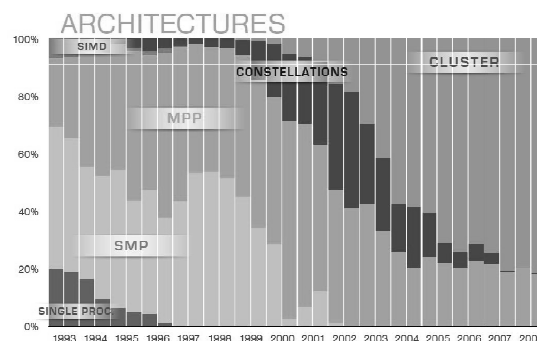
TOP 500 2012 november

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	ORNL OLC Edge National Laboratory United States	Titan - Cray XK7, 1.15peta, Gemini Interconnect, InfiniBand, Cray iW	261192	1.75811	2.1112	11.294
2	U.S. NNSA LLNL United States	Sierra - BlueGene/Q, PowerPC, 1.63peta, Custom IBM	1572681	1.63218	2.0132	7.69
3	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VII-X 2.0peta, Tofu Interconnect, Fujitsu	206074	1.05110	1.1704	12.620
4	CCES/SCS National Laboratory United States	Mira - BlueGene/Q, PowerPC, 1.60peta, Custom IBM	786432	8.1624	10.0663	3943
5	Leibniz Rechenzentrum Jülich (LRZ) Germany	JUQUEVA - IBM x86_64, 1.38peta, PowerPC, 1.000peta, Custom IBM	147436	2.0710	3.1851	3423

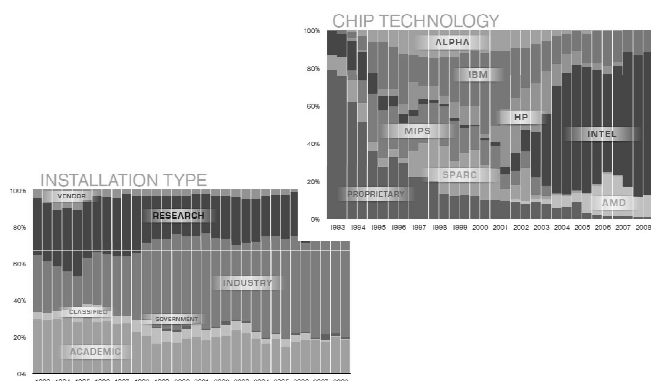
Csak 5.
az első európai
A 8. pedig kínai



Architektúra alakulása



NOVEMBER 2008



NOVEMBER 2008

Operating system Family	Count	Share %	Rmax Sum (GF)	Rpeak Sum (GF)	Processor Sum
Linux	439	87.80 %	13309834	20775171	2099535
Windows	5	1.00 %	328114	429555	54144
Unix	23	4.60 %	881289	1198012	85376
BSD Based	1	0.20 %	35960	40960	5120
Mixed	31	6.20 %	2356048	2933610	869676
Mac OS	1	0.20 %	16180	24576	3072
Totals	500	100%	16927325.79	25401883.80	3116923

Interconnect Family	Count	Share %	Rmax Sum (GF)	Rpeak Sum (GF)	Processor Sum
Myrinet	10	2.00 %	350290	488934	56576
Quadrics	4	0.80 %	122220	147507	21040
Gigabit Ethernet	282	56.40 %	4948233	9795163	941748
Infiniband	141	28.20 %	6349813	8721697	841730
Crossbar	1	0.20 %	35860	40960	5120
Mixed	1	0.20 %	66567	82044	13824
NUMalink	3	0.60 %	122664	137625	21504
SP Switch	10	2.00 %	229541	273754	34208
Proprietary	42	8.40 %	4143049	5243830	1108169
Cray Interconnect	6	1.20 %	359197	466470	73004
Totals	500	100%	16927325.79	25401883.80	3116923

TOP 500 2012 november

- 84.4% legalább 6 magos, 46% pedig legalább 8 magos
- 100. helyen 243.9 Tflop/s az 500. helyen 76.5 Tflop/s
- 75.8% INTEL
- 12% AMD Opteron
- 10% IBM Power
- IBM 193 ↔ HP: 146 Cray: 31

TOP 500 2012 november

- 45% InfiniBand (2x nagyobb telj. adnak)
- 37% Gigabit Ethernet
- Power eff.: 2450Mflops/watt- 90Mflops/watt
- Kínában 72 rendszer, Japánban 31
- Angliában, Franciaországban, Németországban közel azonos: 24, 21, 19
- Linux: 469, UNIX: 20, Windows: 3

Összeköttetések

- Myrinet
 - 10G, réz v. üveg
- Gigabit Ethernet
 - 1G, réz v. üveg
- Infiniband
 - 10-300 Gbit/s, réz
- NUMalink
 - 7.5G, réz

Fájlrendszerek

- NFS (NFS 1,2,3,4) (1985, Sun)
 - V4-et kivéve állapotmentes
- AFS (CMU)
 - Kerberos,
 - nagy cache, nagy cellaszám
 - jól skálázható
- SFS (Lustre, Sun)
 - objektum orientált
 - jól skálázható

Ütemezők

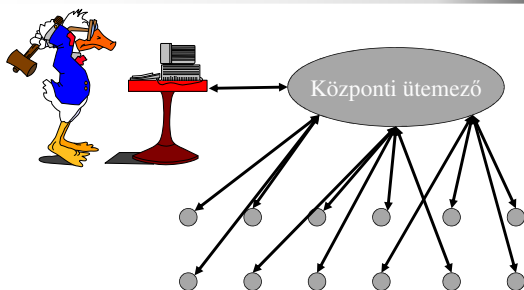
- Condor (Uni. of Wisconsin)
- DQS (Florida State Uni)
- LoadLeveler (IBM)
- Maui, Moab (Cluster Resources)
- LSF (Platform)
- PBS, OpenPBS (Alatair)
- Sun Grid Engne (SUN)
- Torque (Cluster Resources)

A Condor rendszer jellemzői

Speciális ütemező (batch) rendszer

- Elosztott, heterogén rendszerben működik.
- Alapvetően a szabad CPU ciklusok kihasználására tervezték.
- Képes egy működő feladatot áthelyezni az egyik gépről a másikra (migráció).
- Az ún. ClassAds mechanizmussal képes a rendszerben levő változó erőforrásokat az igényeknek megfelelően elosztani.
- Opportunista környezet.

Condor pool



ClassAds lényege

- A rendszerben levő erőforrások különböző jellemzőkkel (teljesítmény, architektúra, op. rendszer, stb.) rendelkeznek.
- A job összeállításánál ezekre a jellemzőkre igényeket lehet előírni, amit a Condor rendszer megpróbál kielégíteni. (Párosítja az igényt az erőforrással)
- A job összeállításánál lehetőség van preferenciák megadására, ami alapján a Condor rangsorolni fog és kiválasztja az igénynek leginkább megfelelő gépet.

ClassAds lényege (2)

- Így nincs szükség a batch rendszerekben megszokott sorokra. (Úgyis a rosszat választanánk)

Követelmény és rangsor

- Követelmény:

Requirements = Arch=="SUN4u"

Pontosan kell illeszkednie.

- Rangsor:

Rank = Memory + Mips

Ha választhat, akkor a nagyobbat fogja választani

A dolgok két oldala (1)

A kifejezések a két hirdetés adatterében értékelődnek ki (adA, adB).

Felhasználó (igénylő) oldala:

Requirements = Arch == "INTEL" &&

OpSys == "LINUX"

Rank = TARGET.Memory * 10 +

TARGET.Disk + Mips

A dolgok két oldala (2)

Erőforrás oldal:

Friend = Owner == "haver"

Trusted = Owner != "judas"

Mygroup = Owner == "zoli" || Owner == "jani"

Requirements = Trusted && (Mygroup ||

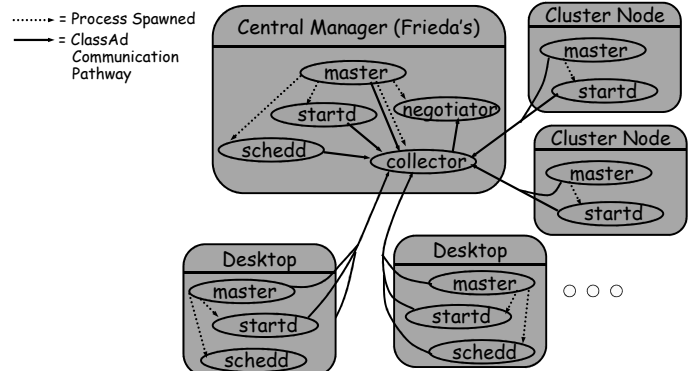
LoadAvg < 0.5 && KeyboardIdle > 10*60)

Rank = Friend + MyGroup*10

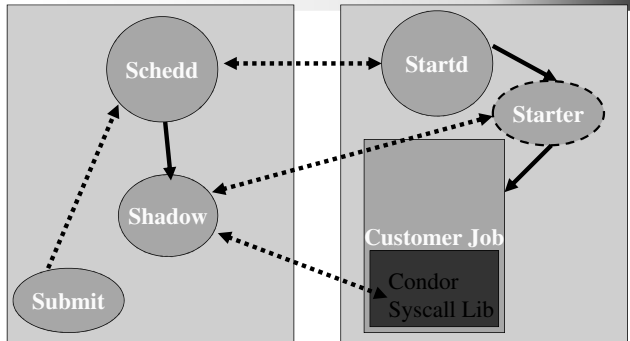
Feladatkörök

- Central Manager
- Execute Machine
- Submit Machine
- Checkpoint Server

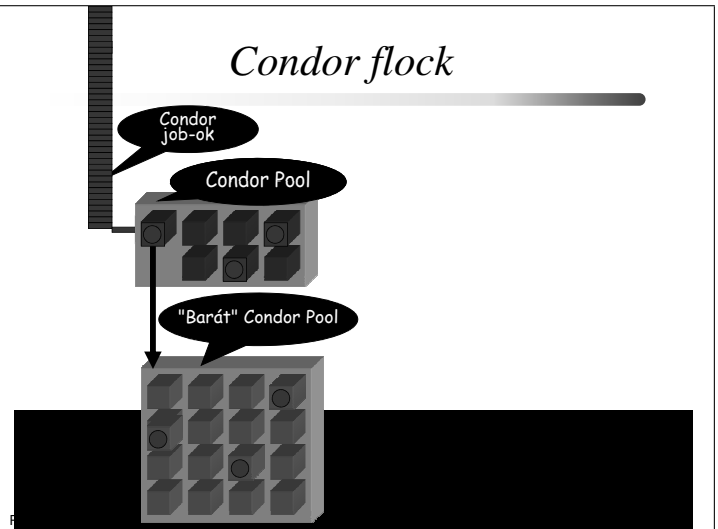
Condor Pool



Job indítás



Condor flock



Milyen feladatok lehetnek ?

- Elsősorban hosszú futási idejű, számításigényes feladatok.
- Különböző univerzumok léteznek
 - Standard
 - Scheduler
 - Vanilla
 - Local
 - MPI
 - Parallel
 - Grid
 - VM
 - Java

Standard univerzum

- checkpointing, automatikus migráció
- meglevő programot újra kell fordítani, esetleg csak linkelni
- az alkalmazás nem használhat bizonyos rendszerhívásokat: pl. fork, socket, alarm, mmap
- („elkapja” a file műveleteket)

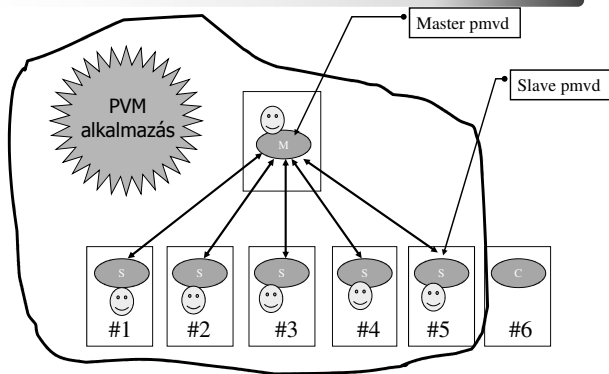
Vanilla univerzum

- nincs checkpointing, nincs migráció
- meglevő futtatható kódot nem kell változtatni
- nincs korlátozás a rendszerhívásokkal szemben.
- NFS, vagy AFS kell !!!!

PVM univerzum

- MW jellegű PVM programok környezete
- Binárisan kompatibilis
- PVM 3.4.2 + taszk kezeléshez kieg.
- Dinamikus VM kialakítás.
- Heterogén környezet támogatása
- Egy user csak egy példányban futathat deamont

Condor felépíti a virtuális gépet



MPI univerzum

- MPICH változtatás nélkül.
- Bináris kompatibilitás
- Csak ch_p4 device
- Dinamikusan nem változhat
- Nem állhat meg.
- NFS vagy AFS kell.

Futtatás lépései

- A job összeállítása
- Job bejelentése a Condor-nak
- Job-ot a Condor futtatja az általa kiválasztott gép(eken), szükség esetén átmozgatja egy másik gépre.
- Job befejeződik, a Condor e-mail-t küld a felhasználónak.

Egy egyszerű jobleíró

universe = vanilla
 executable = mathematica
 input = in\$(Process).dat
 output = out\$(Process).dat
 queue 50

Egy másik jobleíró

universe = vanilla
 executable = /bin/hostname
 output = hostname.out.\$(Process)
 error = hostname.err.\$(Process)
 log = hostname.log
 queue 3

Sun Grid Engine (SGE)

- A Condor-hoz hasonló ütemező.
- Queue-kat definiál.
- Hangsúlyos a terhelés kiegyensúlyozása.
- Backup master ütemező.
- Check-point.
- Migrálási lehetőség.
- Négy szerepkör:
 - master, submit, exec, admin,

SGE komponensei

