



Superpočítání a gridové počítání

Martin Petřek,^{1,2} Petr Kulhánek,^{1,2}

Jan Kmuníček^{1,3}

petrek@chemi.muni.cz, kulhanek@chemi.muni.cz, kmunicek@ics.muni.cz

1) CESNET z. s. p. o., Žitkova 4, CZ-16000 Praha, Česká republika

2) Národní centrum pro výzkum biomolekul, Přírodovědecká Fakulta, Masarykova univerzita,
Kotlářská 2, 61137 Brno, Česká republika

3) Ústav výpočetní techniky, Masarykova univerzita, Botanická 68a,
60200 Brno, Česká republika





Obsah

1. Náročné výpočty a aplikace
 - Výpočetní chemie, částicová fyzika, zpracování dat
2. Gridové systémy a práce v nich
 - METACentrum, EGEE2
3. Software pro řazení a správu úloh
 - PBS, gLite/LCG
 - ukázka spouštění jobů
4. Systém CHARON
 - Koncepce systému
 - Použití na klastru a v gridu
 - Správa aplikací



Náročné výpočty a aplikace

Co jsou náročné výpočty?

- relativní pojem vzhledem k prudkému vývoji výpočetní techniky
- jednotka výkonu – FLOPS (Floating Point Operations Per Second)
- jednotka dat – BYTE
- dnešní Pentium 4, 1GB RAM, 2GHz – výkon několik GFLOPS
- *do náročných výpočtů řadíme aplikace vyžadující*
 - výkon v řádech \geq TFLOPS vyšší
 - práce s daty v řádech \geq GB
- *„aplikace běžící na superpočítačích nebo rozsáhlých výpočetních systémech (gridy)“*
- doba běhu na domácím PC by trvala týdny, měsíce, roky, ...

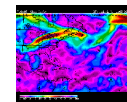


Náročné výpočty a aplikace

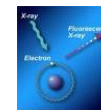
Typy aplikací:

- *Matematicko-Fyzikální aplikace:*

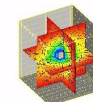
- modely předpovědi počasí (systém Aladin)



- simulace experimentů z oblasti částicové fyziky (HEP)



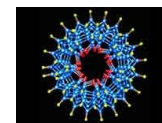
- úlohy z oblasti pružnosti-pevnosti, termo-elasticita (FEM)



- simulace proudění kapalin (CFD)

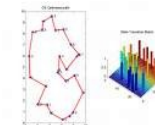


- materiálové inženýrství, nanotechnologie

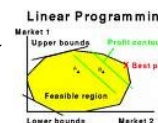


- simulace zemetřesení

- NP-těžké úlohy (TSP), optimalizační úlohy



- úlohy z lineárního resp. matematického programování



- lámání šifer (DES, Enigma [M4 Project])



- hledání prvočísel (GIMPS), ...

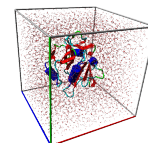


Náročné výpočty a aplikace

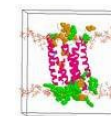
Typy aplikací:

- *Chemické-biologické aplikace:*

- simulace chování biologických systémů (Molekulová dynamika)
- návrhy léčiv (studium interakce enzym X léčivo)



- molekulové dokování a konformační analýza molekul



- zkoumání reakčních mechanismů (tranzitní stavy, odhady energetických rozdílů pro reakční cestu, výpočty 'volné energie')



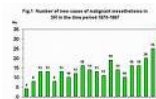
- protein folding



- simulace chování organismů v prostředí



- šíření epidemií v prostředí





Náročné výpočty a aplikace

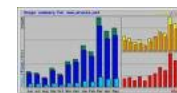
Typy aplikací:

- *Zpracování dat:*

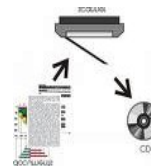
- lékařství (CT-snímky, NMR, příznakové rozpoznávání)



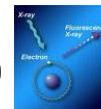
- zpracování rozsáhlých statistik



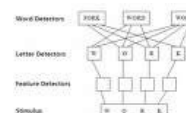
- analýza a rozpoznávání obrazu



- HEP - částicové experimenty (ATLAS, CMS, Alice, LHCb)



- tvorba expertních systémů (AI)



- *Visualizace dat*

- *renderování náročných scén*



- *Ostatní*

- simulace sociálních a ekonomických jevů

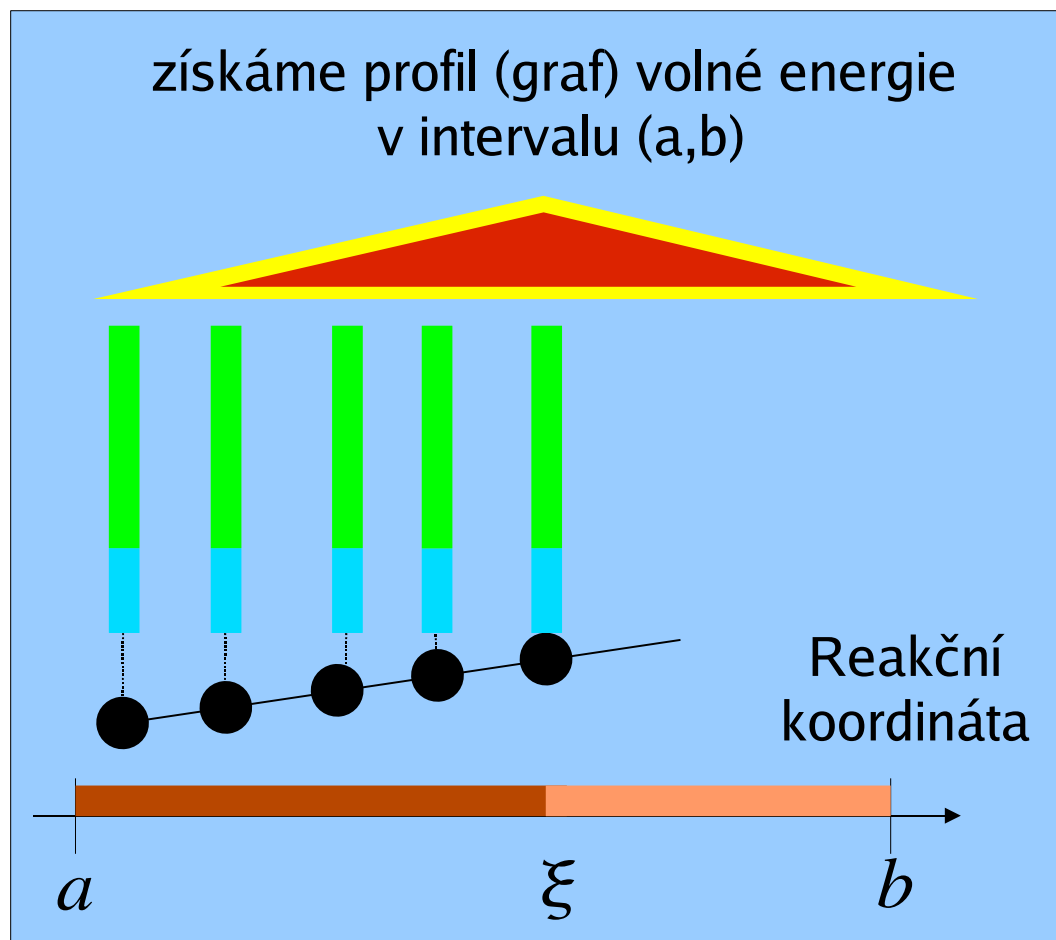
- ...a spousta dalších...



Náročné výpočty a aplikace

Příklad z výpočetní chemie – výpočet volné energie

- hlavní úloha (doba běhu ~ 15h) - generuje mnoho podúloh (stovky)
- podúloha (doba běhu ~ 25 h) (2 CPU)



- na domácím PC (1CPU)
by úloha trvala ~ 7 měsíců
(24 h denně)
- v METACentru ~ za 3 dny
máme výsledky



Náročné výpočty a aplikace

Příklad z výpočetní chemie – výpočet vibračních modů molekuly

- výpočet matice 2. derivací energie podle souřadnic (tzv. Hessian)
- $3 \cdot N \cdot 2$ nezávislých výpočtů gradientu energie (Quant. Mech.)

$$\begin{pmatrix} \frac{\partial^2 E}{\partial x_1 \partial x_1} & \frac{\partial^2 E}{\partial x_1 \partial y_1} & \frac{\partial^2 E}{\partial x_1 \partial z_1} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_n} & \frac{\partial^2 E}{\partial x_1 \partial y_n} & \frac{\partial^2 E}{\partial x_1 \partial z_n} \\ \frac{\partial^2 E}{\partial y_1 \partial x_1} & \frac{\partial^2 E}{\partial y_1 \partial y_1} & & \cdots & & \frac{\partial^2 E}{\partial y_1 \partial y_n} & \frac{\partial^2 E}{\partial y_1 \partial z_n} \\ \frac{\partial^2 E}{\partial z_1 \partial x_1} & & & & & & \frac{\partial^2 E}{\partial z_1 \partial z_n} \\ \vdots & & & \ddots & & & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & & & & & & \frac{\partial^2 E}{\partial x_n \partial z_n} \\ \frac{\partial^2 E}{\partial y_n \partial x_1} & \frac{\partial^2 E}{\partial y_n \partial y_1} & & \cdots & \frac{\partial^2 E}{\partial y_n \partial y_n} & \frac{\partial^2 E}{\partial y_n \partial z_n} \\ \frac{\partial^2 E}{\partial z_n \partial x_1} & \frac{\partial^2 E}{\partial z_n \partial y_1} & \frac{\partial^2 E}{\partial z_n \partial z_1} & \cdots & \frac{\partial^2 E}{\partial z_n \partial x_n} & \frac{\partial^2 E}{\partial z_n \partial y_n} & \frac{\partial^2 E}{\partial z_n \partial z_n} \end{pmatrix}$$

- $N \sim 100$ atomů \Rightarrow 600 úloh
- 1 úloha \sim 1 hodina
- **na domácím PC (25 dní)**
- **v METACentru \sim 1hodina**



Gridové systémy a práce v nich

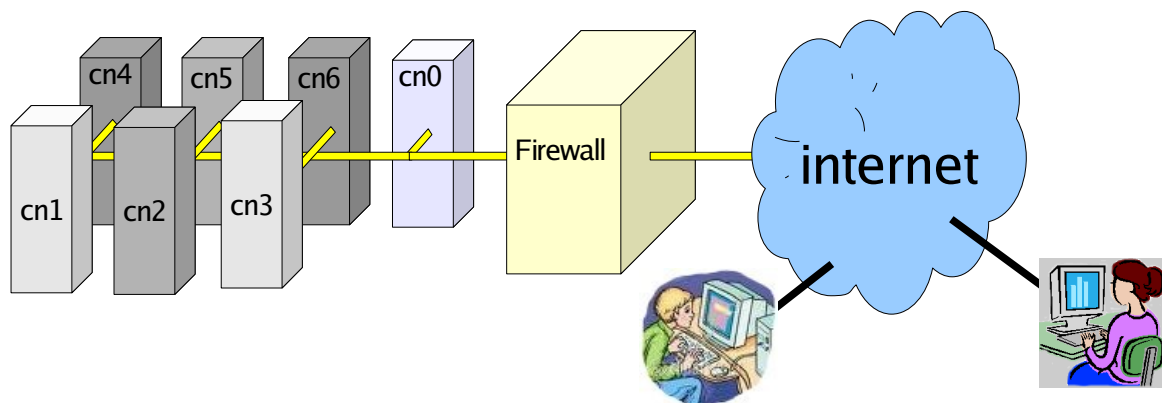
- *Computer cluster*
 - několik počítačů spojených pomocí sítě (LAN)
 - lze s nimi pracovat odděleně nebo se můžou navenek (při vzdáleném připojení) jevit jako jeden počítač
 - uvnitř sítě se lze svobodně pohybovat (jednotlivé počítače si navzájem „věří“)
 - lze poměrně levně postavit z běžně dostupných PC a síťových komponent
 - většinou stejné typy strojů (homogenní cluster X heterogenní cluster)
- *Gridový systém*
 - rozsáhlý co do počtu výpočetních strojů, ukládacích kapacit, ...
 - chápán spíš jako výpočetní nástroj než jako jeden počítač
 - spojení několika „clusterů“, různé architektury, heterogenní stroje
 - velký důraz na bezpečnost (dílčí clustery mohou být různě po světě)



Gridové systémy a práce v nich

Společné znaky většiny klastrů:

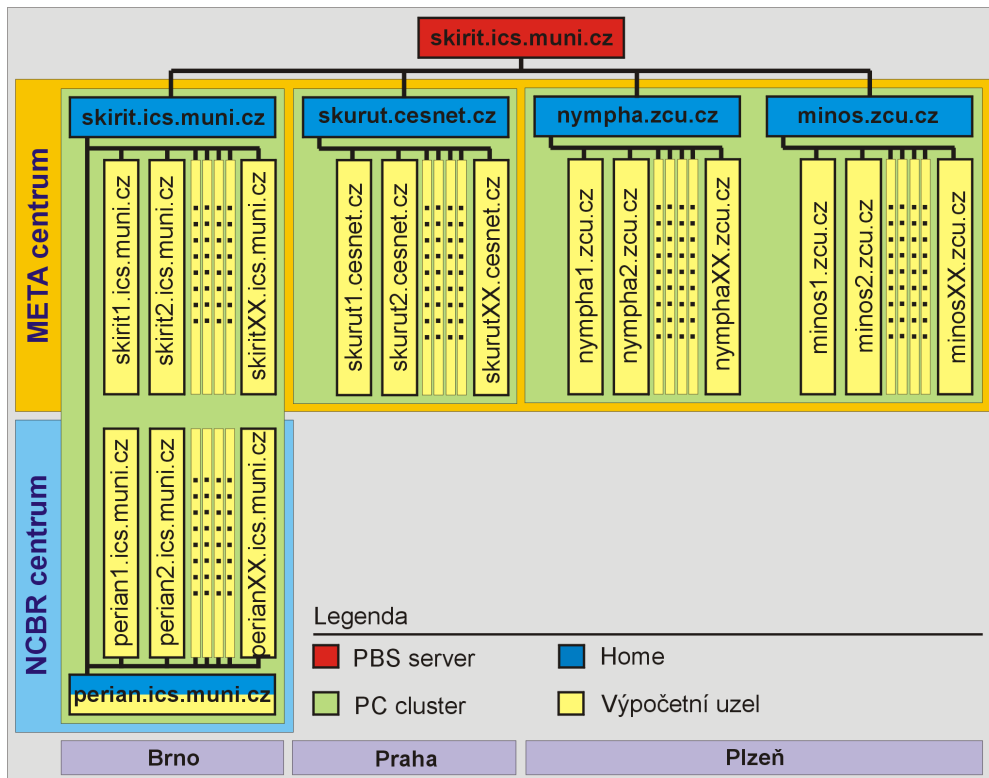
- operační systémy typu UNIX (unix, linux, freebsd, netbsd, ...)
- sdílení souborů v klustru (souborové systémy NFS, AFS, ...)
- systém správy aplikačního softwaru (systém tzv. modulů)
- autentizační systém v rámci klastru (Kerberos)
- aspoň jeden centrální uzel pro přístup zvenčí (SSH, certifikáty,...)
- software pro řazení úloh do fronty (PBS+varianty, NQE, LSF, ...)
- uživatel má účet, domovský adresář přímo v systému
- z centrálního uzlu se lze logovat na jednotlivé stroje bez hesla



Gridové systémy a práce v nich

příklad HPC (high-performance cluster):

METACENTRUM



<http://meta.cesnet.cz> (sdružení CESNET)

- Distribuovaný výpočetní systém
 - Superpočítačové centrum Brno MU (<http://scb.ics.muni.cz/static>)
 - Superpočítačové centrum UK (<http://supercomp.cuni.cz>)
 - Superpočítačové centrum ZČU (<http://zsc.zcu.cz>)

Techinfo: 218 uzlů, 463 CPU

SMP stroje (shared memory), klastry (1-2 procesorové PC)

1Gb/s (GE, Gigabit Ethernet) nebo 2.5Gb/s (Myrinet)

Gridové systémy a práce v nich

příklad HPC (high-performance cluster):

METACENTRUM

Programové prostředky:

- distribuovaný souborový **system AFS**
- autentizační **system Kerberos** (kinit, kauth, SSH protokol)
- systém správy aplikačního software **(meta)moduly**
- přístup na centrální uzel pomocí **SSH**
- přístup pomocí hardwarových klíčů (**Token s certifikátem**)



Software pro řazení úloh (dávkové systémy):

- **PBSPro** – Portable Batch System, dávkový systém pro PC klastr



Gridové systémy a práce v nich

Specifika Gridů:

- výpočetní zdroje nejsou spravovány centrálně
- administrativní rozdělení gridu na „virtuální organizace (VO)“
 - speciální uzly pro ukládání dat – stroje, které zajišťují služby pro práci se soubory (Storage Elements)
 - služby pro monitorování stavu gridu
 - služby pro plánování úloh (Computing Elements)
 - vlastní výpočetní kapacity (Worker Nodes)
- k propojení VO slouží „grid-middle-ware“ - otevřené standardy
- k přihlášení do gridu slouží několik počítačů (User-Interface)
- autentizace pomocí certifikátů (silné elektronické šifrování)
- uživatel patří do VO, nemá přímý přístup ke zdrojům, ale ke službám



Gridové systémy a práce v nich

příklad gridu: Enabling Grid for E-science (EGEE2)



- <http://egee.cesnet.cz> (informace o projektu)
- mezinárodní projekt Evropské Unie (CESNET za ČR - VOCE)
- celoevropská gridová infrastruktura pro vědeckou komunitu i průmysl (>30 zemí, 100 organizací)

pilotní aplikace:

- HEP (High Energy Physics) – zpracování a analýza dat z experimentů částicové fyziky (Atlas, CMS, Alice, LHCb, ...)
- výpočetně-chemické simulace biologických systémů
- biomedicínské gridy
- zpracování bioinformatikých a lékařských dat





Gridové systémy a práce v nich

příklad gridu: Enabling Grid for E-science (EGEE2)



Techinfo:

- přes **20 000 CPU** (7x24h), **5 PB** (5 miliónů GB), **1.5 GB/s**

Programové prostředky:

- grid-middle-ware: gLite/LCG, EDG, Genius
- bezpečnost:
 - GSI (Grid Security Infrastructure)
 - X.509 certifikáty vydávané národními certifikačními autoritami (CA)
- monitorování stavu gridu: LCG2 Real Time Monitor
- databázové služby (MySQL, Oracle, ...)
- webové služby





Gridové systémy a práce v nich

Ukázka typické práce na klastru:

- 1) připojení z domácího stroje na centrální uzel klastru
- 2) příprava úlohy a spouštěcího skriptu
- 3) odeslání úlohy do fronty
- 4) monitorování úlohy
- 5) obdržení výsledků

- i) zastavení resp. restart úlohy
- ii) přeplánování úlohy, zrušení naplánované úlohy
- iii) specifikace zdrojů, kde má úloha běžet
- iv) monitoring stavu klastru (volné stroje, výpadky klastru)

- více uživatelů generuje spoustu úloh, kapacita zdrojů omezená
=> **system pro plánování, řazení a správu úloh**



Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

- *úlohy se řadí do tzv. front*

fronty:

<u>Jméno fronty</u>	<u>Max. doba běhu</u>	<u>Maximum úloh</u>	<u>Maximum/Uživatel</u>
• short	2 hodiny	12	8
• normal	24 hodin	24	12
• long	720 hodin	96	32
• ncbr	720 hodin	120	32
• cpmd	720 hodin	120	16

- *strojům lze přiřadit tzv. vlastnosti (využití v heterogenních clusterech)*

Vlastnosti (meta):

- linux
- praha
- brno
- plzen
- iti

Vlastnosti (ncbr):

- lcc
- ibp
- cpmd

Vlastnosti (obecné):

- p3
- xeon
- athlon



Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

Příkazy pro základní práci s úlohami:

- *zaslání úlohy do fronty (qsub)*
- *vymazání ještě nespouštěné úlohy z fronty (qdel)*
- *informace o běžících úlohách (qstat)*
- *Informace o uzlech (pbsnodes, xpbs)*

v praxi to vypadá přibližně takto:

- *odeslání úlohy do fronty:*

```
[petrek@skirit test]$ qsub -r n -m abe -j oe -o test.out \
-e test.err -N "Test cislo 1" \
-q normal -l "node=1:brno:xeon" \
-v "BACKUPDIR" test
142606.skirit.ics.muni.cz
[petrek@skirit test]$
```

standardní a
chybový
výstup

Identifikátor úlohy

proměnné
prostředí

fronta a vlastnosti

vlastní skript



Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

- *vlastní skript*

```
[petrek@skirit petrek]$ cat test
```

```
#!/bin/bash
```

```
#PBS -W stagein=/scratch/petrek/xxx.com@skirit:test/xxx.com
```

```
#PBS -W stageout=/scratch/petrek/xxx.log@skirit:test/xxx.log
```

```
# Inicializace modulu a pridani modulu g98:
```

```
. /packages/run/modules-2.0/init/sh
```

```
module add g98
```

```
# zmena pracovniho adresare
```

```
cd /scratch/petrek
```

```
# Spusteni ulohy:
```

```
g98 xxx.com
```

Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

- Informace o úlohách

```
[petrek@skirit petrek]$ qstat
Job id          Name          User          Time Use  S  Queue
-----
138034.skirit-f tri_2fsm      zeleny        68:49:00 Q  long
138035.skirit-f tri_3fsm      zeleny       188:01:0  Q  long
138036.skirit-f tri_4fsm      zeleny       99:39:18  Q  long
138195.skirit-f  opt1         jsebera     107:21:3  Q  long
139206.skirit-f  jedu        sponer      621:11:3  R  ncbr
139731.skirit-f  a2:=24      hornak      531:31:2  R  iti
140366.skirit-f  24t5p.run   vrbka      1109:53:  R  parallel
142457.skirit-f  S011       petrek       05:22:49  C  cpmd
142562.skirit-f  m2sr       soliman     28:24:05  R  cpmd
142606.skirit-f  test       petrek              0  Q  normal
```

režimy úlohy: Q (naplánovaná) => R (running) => E (end) => C (completed)

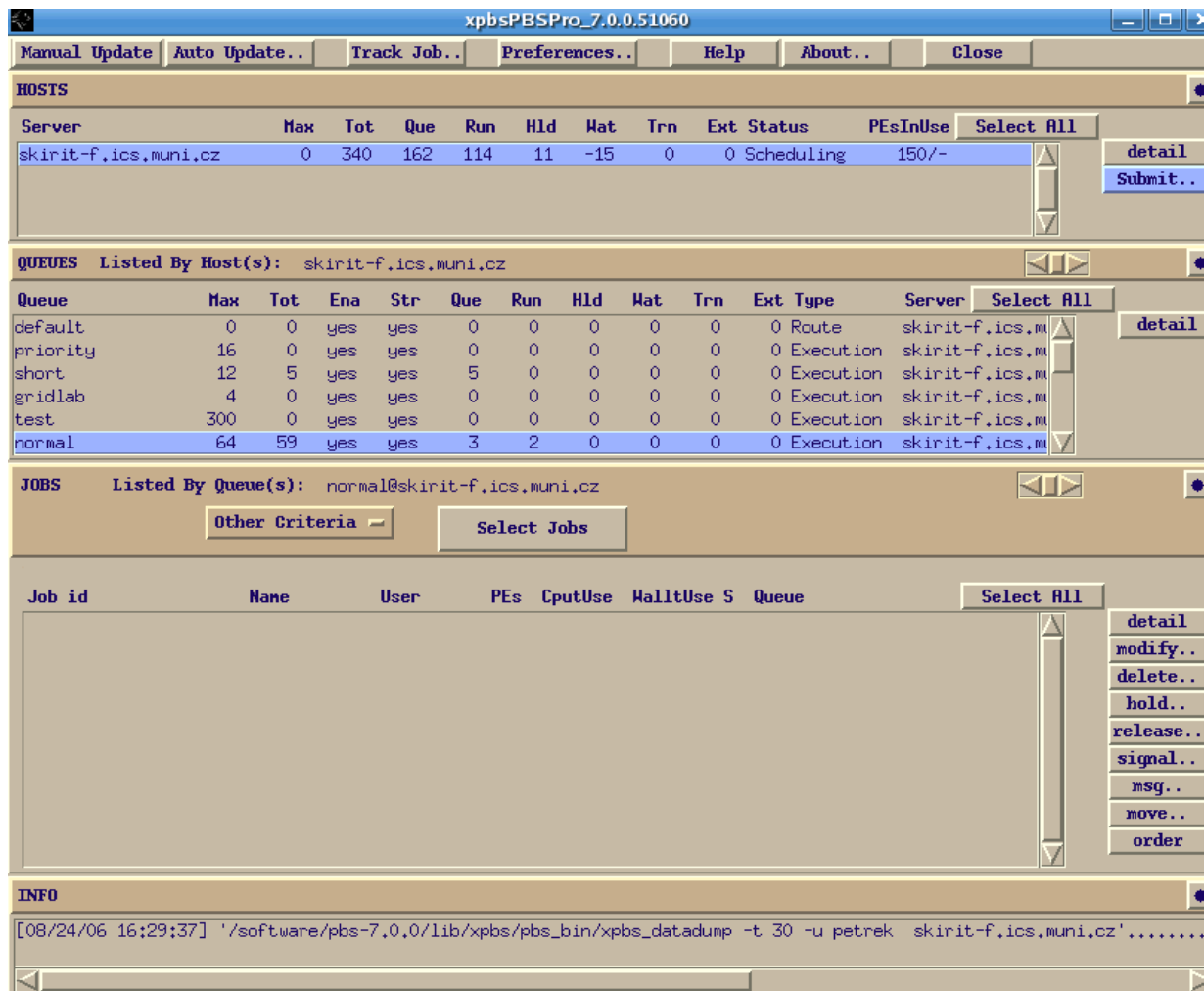
- smazání z fronty ve stavu Q:

```
[petrek@skirit petrek]$ qdel 142606
```

Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

- *Monitoring úloh:* [xpbs](#)



The screenshot displays the xpbsPBSPro_7.0.0.51060 application interface. It features a menu bar with options: Manual Update, Auto Update..., Track Job..., Preferences..., Help, About..., and Close. The main content is divided into three sections: HOSTS, QUEUES, and JOBS.

HOSTS

Server	Max	Tot	Que	Run	Hld	Mat	Trn	Ext	Status	PEsInUse	Select All
skirit-f.ics.muni.cz	0	340	162	114	11	-15	0	0	Scheduling	150/-	detail Submit..

QUEUES Listed By Host(s): skirit-f.ics.muni.cz

Queue	Max	Tot	Ena	Str	Que	Run	Hld	Mat	Trn	Ext	Type	Server	Select All
default	0	0	yes	yes	0	0	0	0	0	0	Route	skirit-f.ics.m	detail
priority	16	0	yes	yes	0	0	0	0	0	0	Execution	skirit-f.ics.m	
short	12	5	yes	yes	5	0	0	0	0	0	Execution	skirit-f.ics.m	
gridlab	4	0	yes	yes	0	0	0	0	0	0	Execution	skirit-f.ics.m	
test	300	0	yes	yes	0	0	0	0	0	0	Execution	skirit-f.ics.m	
normal	64	59	yes	yes	3	2	0	0	0	0	Execution	skirit-f.ics.m	

JOBS Listed By Queue(s): normal@skirit-f.ics.muni.cz

Other Criteria [v] Select Jobs

Job id	Name	User	PEs	CputUse	WalltUse	S	Queue	Select All
--------	------	------	-----	---------	----------	---	-------	------------

detail
modify..
delete..
hold..
release..
signal..
msg..
move..
order

INFO

```
[08/24/06 16:29:37] '/software/pbs-7.0.0/lib/xpbs/pbs_bin/xpbs_datadump -t 30 -u petrek skirit-f.ics.muni.cz'.....
```

METACENTRUM

Software pro řazení a správu úloh

PBS – Portable Batch System (dávkový systém pro klastry)

- přehled vytížení strojů:

<http://meta.cesnet.cz/pbsmon/nodes.do>

Stroje

Podrobnější výpis z Ganglie

pracující	částečně volné	volné	nepřipojené	vypnuté	neznámý stav	celkem
114	9	87	1	3	4	218

acharon ajax aule glamdring konos minos narsil nympha

konos11

minos1	minos2	minos3	minos4	minos5
minos9	minos10	minos11	minos12	minos13
nympha1	nympha2	nympha3	nympha4	nympha5
nympha9	nympha10	nympha11	nympha12	nympha13
perian1	perian4	perian5	perian6	perian7
perian11	perian12	perian13	perian14	perian15
perian19	perian20	perian21	perian22	perian23
perian27	perian28	perian29	perian30	perian31
perian35	perian36	perian37	perian38	perian39
perian43	perian44	perian45	perian46	perian47
perian51	perian52	perian53	perian54	perian55
perian59	perian60	perian61	perian62	perian63
perian67	perian68	perian69	perian70	perian71
perian75	perian76			
puu1	puu2	puu3	puu4	puu5
puu9	puu10			
quark1	quark2	quark3	quark4	quark5
skirit1	skirit2	skirit3	skirit4	skirit5
skirit9	skirit10	skirit11	skirit12	skirit13
skirit17	skirit18	skirit19	skirit20	skirit21
skirit25	skirit26	skirit27	skirit28	skirit29
skirit33	skirit34	skirit35	skirit36	skirit37
skirit41	skirit42	skirit43	skirit44	skirit45
skurut33	skurut34	skurut35	skurut36	skurut37
skurut41	skurut42	skurut43	skurut44	skurut45
skurut49	skurut50	skurut51	skurut52	skurut53
skurut57	skurut58	skurut59	skurut60	skurut61
skurut65	skurut66	skurut67		

perian37.ics.muni.cz

Informace z PBSPro:

jmeno	perian37.ics.muni.cz
stav	job-busy
architektura	linux
typ uzlu	cluster
vlastnosti	onlycpmd,normal,cpmd,per,xeon,i386,debian
využití	dostupné
CPU	2
paměť	0kb
	1031436kb

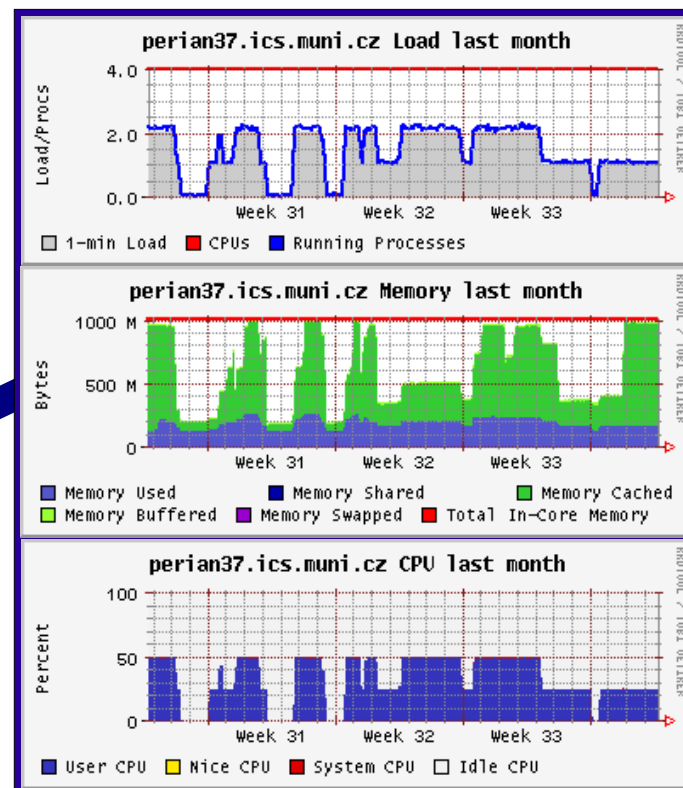
číslo CPU	úloha	uživatel	jmeno	čas vytvoření
0	142553.skirit-f.ics.muni.cz	petrek	S041	23.8.06 11.07
1	142553.skirit-f.ics.muni.cz	petrek	S041	23.8.06 11.07

Informace z Ganglie:

jmeno	perian37.ics.muni.cz
operační systém	Linux 2.4.29-smp
typ CPU	x86
processory	4x 2399 MHz
zátěž procesorů	25.0% user, 0.0% system, 0.0% nice, 75.0% idle
průměrný počet úloh	1.00 (1 min), 0.97 (5 min), 0.91 (15 min)
čas startu	Čtvrtek, 27. červenec 2006 10:13:19 CEST
systémový čas	Čtvrtek, 27. červenec 2006 10:13:45 CEST
okamžik hlášení	Čtvrtek, 24. srpen 2006 16:34:46 CEST
volné místo na disku	19.619 GB / 34.238 GB
volná paměť	24884 KB / 1031436 KB
volný swap	2097136 KB / 2097136 KB

perian37.ics.muni.cz Load last week

perian37.ics.muni.cz Load last month



METACENTRUM

Software pro řazení a správu úloh

Nevýhody přímého použití dávkových systémů:

- nutná znalost front, vlastností
- uživatel musí znát poměrně dost informací o systému
- **kopírování vstupních dat na výpočetní uzel a stažení výsledku musí zajistit váš skript :-)**
- paralelní úlohy - speciální volby ve spouštěcím skriptu ohledně architektury (shmem, p4, mpich-gm)
- **nastavení cest k software – uživatel musí opět znát, co je kde nainstalováno, jakou architekturu použít**
=> různé skripty pro různé architektury :-)
- informace o úloze svázané s identifikačním číslem jobu => při velkém množství úloh neúnosné





Software pro řazení a správu úloh

Služby v gridových systémech: middleware gLite/LCG

Certifikáty: (bez nich nelze na gridu existovat)

- soubor s informacemi o vaší identitě; má omezenou platnost, údaje šifrované

příkazy pro práci s cert.:

- pro dlouhodobější úlohy => MyProxyCertifikát

příkazy pro operaci se soubory:

- lcg-cp, ...

příkazy pro práci s úlohou:

- edg-job-submit, ...

příkazy pro službu s VOMS (Virtual Organization Membership Service):

- edg-voms-proxy-info



Software pro řazení a správu úloh

Služby v gridových systémech: middleware gLite/LCG

práce na gridu:

- 1) připojení z domácího stroje na User Interface (gsssh)
- 2) Inicializace certifikátů (myproxy-init-sc, myproxy-get-delegation)
- 3) Nahrání vstupních dat na storage element (lcg-cp)
 - služba vrátí identifikátor souboru na SE
- 4) Sestavení popisovacího skriptu pro úlohu (*.JDL)
- 5) Vlastní odeslání úlohy do gridu (edg-job-submit)
 - služba vrátí identifikátor jobu
- 6) Sledování stavu úlohy (edg-job-status)
- 7) Stáhnutí výsledku ze storage elementu (lcg-cr)

Software pro řazení a správu úloh

Služby v gridových systémech: middleware gLite/LCG

4) Sestavení popisovacího skriptu pro úlohu (*.JDL)

```
# JDL Test.jdl
Type = "Job";
JobType = "Normal";
Executable = "Test";
StdOutput = "Test.stdout";
StdError = "Test.stderr";
InputSandbox = {"in1.xml", "in2.xml"};
OutputSandbox = {"out1.xml", "out2.xml"};
Environment = {
  "AMBERPATH=/var/amber",
  "BIGFILE1=guid:645c2af0-498e-4657-8154-8295380b349e"
};
Arguments = "";
RetryCount = 1;
```

předává se s spolu s úlohou

identifikátor souboru na SE

5) Vlastní odeslání úlohy do gridu (edg-job-submit)

```
$ export VOCONFIG=edg_wl_ui.conf
$ edg-job-submit --config-vo $VOCONFIG -o JID test.jdl
```



Software pro řazení a správu úloh

Nevýhody přímého použití API gridu:

- JDL jazyk
- Správa identifikátoru pro soubor
- **kopírování vstupních dat z SE na výpočetní uzel (WN) a nahrání výsledků na SE musí zajistit váš skript :-)**
- speciální volby v popisovacím JDL skriptu ohledně par. architektury, délky jobu
- **software – je třeba kopírovat s úlohou nebo předávat informace, odkud lze spouštět (není známé obecně na gridu)**
- informace o úloze svázané s identifikačním číslem jobu, místo souborů identifikátory na SE
=> při velkém množství úloh opět neúnosné

Software pro řazení a správu úloh

„Kdo si tohle objednal?“

I.I. Rabi (1946-1947)

*„Naštěstí existují nádstavby
nad přímým použitím dávkových systémů“*



=> **system CHARON**

(další možnosti: UNICORE, GENIUS portál, ...)



System CHARON

Co je CHARON?

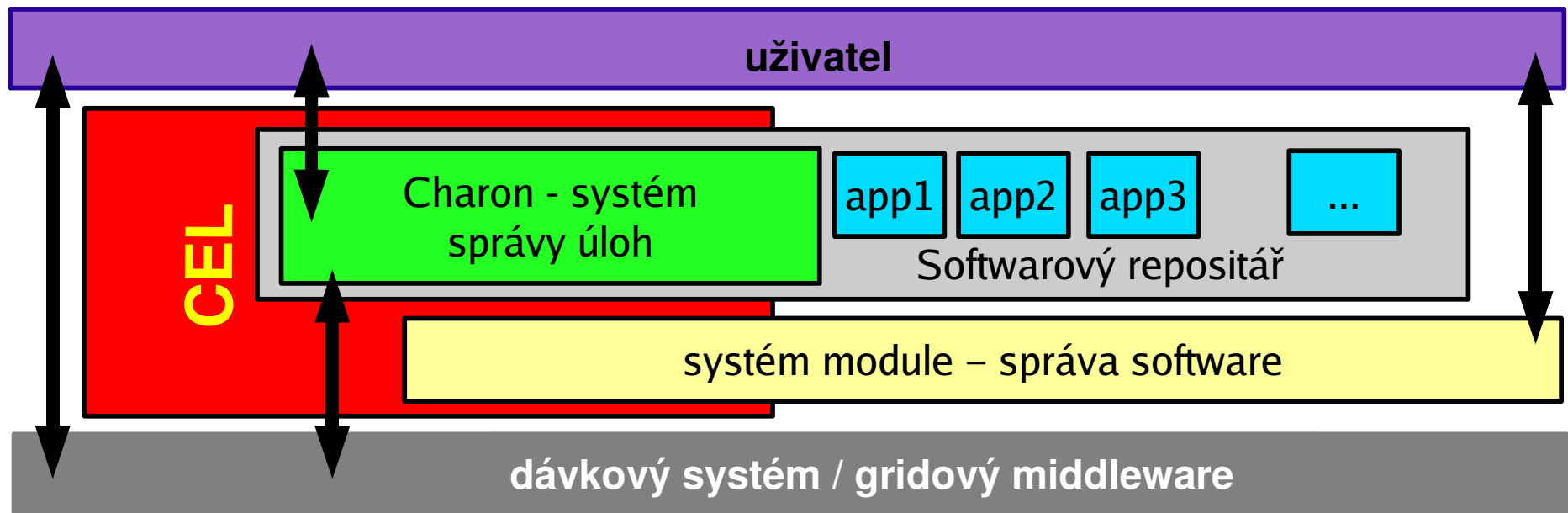
- komplexní nadstavba na dávkovými/gridovými systémy, zajišťující sjednocený přístup k využívání výpočetních zdrojů
- nástroj pro správu a údržbu aplikací v těchto systémech
- nástroj pro sjednocené odesílání a sledování úloh

Proč CHARON?

- složitost dávkových a gridových middle-ware přesahuje uživatelsky únosnou mez
- v této oblasti Charon nabízí maximální zjednodušení práce
- způsob práce na gridu, resp. na klastrech je pro uživatele jednotný (přenositelnost úloh)



System CHARON – koncepce systému



- správa aplikací
 - spouštění úloh pro více procesorů (paralelní úlohy) bez nutnosti modifikace hlavního skriptu úlohy
- správa úloh
 - Jednoduché odeslání jobu, monitoring a obdržení výsledků
- příkazový řádek (Command Line Interface)



System Charon – použití na klastru

1) připojení na centrální uzel klastru (SSH, Putty)

```
[petrek@joplin ~]$ hostname
joplin.chemi.muni.cz
[petrek@joplin ~]$ ssh test1@wolf.chemi.muni.cz
test1@wolf.chemi.muni.cz's password:
Last login: Wed Aug 23 14:30:31 2006 from joplin.chemi.muni.cz

      *** Welcome to WOLF cluster ***
=====
You are logged on host : wolf.chemi.muni.cz
Its architecture is   : i786
-----
      Charon Extension Layer (CEL) is enabled.
      If you have any problem with this system you should
      contact authors. (More information at WWW portal.)
                ***
      http://troll.chemi.muni.cz/whitezone/development/charon
-----

[test1@wolf ~]$
```



System Charon – použití na klastru

2) vytvoření adresáře s úlohou (nakopírování přes SCP)

```
[test1@wolf ~]$ scp -r petrek@wolf:job1 .  
petrek@wolf's password: *****  
job1.run          100%    49      0.1KB/s    00:00  
input1.pov       100%  3347    3.3KB/s    00:00
```

3) vytvoření spouštěcího skriptu pro úlohu

```
[test1@wolf ~]$ cd GridComputing/01.simple/job1  
[test1@wolf job1]$ ls  
input1.pov  job1.run*  
[test@wolf job1]$ cat job1.run
```

```
#!/bin/bash  
  
# activate povray package  
module add povray  
  
# render scene  
povray -W800 -H600 input1.pov
```




System Charon – použití na klastru

4) odeslání úlohy

```
[test1@wolf ~]$ psubmit long job1.run
Job name      : job1.run
Job title     : job1.run (Job type: generic)
Job directory :
wolf.chemi.muni.cz:/home/test1/GridComputing/01.simple/job1
Job project   : -none-
Cluster name  : WOLF (Driver: pbs)
=====
Alias         : -none-
Queue        : long
Profile      : wolf
-----
NCPU         : 1
Resources    : nodes=1:ppn=1:node
Sync mode    : sync
-----
Start after  : -not defined-
=====
Do you want to submit job with pbs driver (YES/NO)?
> YES

Job was successfully submitted to PBS queue system.
```



System Charon – použití na klastru

5) vzniklé kontrolní soubory

```
[test1@wolf job1]$ ls
input1.pov  job1.run*  job1.run.ces*  job1.run.info
```

6) stav úlohy

```
[test1@wolf job1]$ pgstat1
```

```
wolf.chemi.muni.cz:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
-----	-----	-----	-----	-----	---	---	-----	-----	--	----
700.wol	test1	long	job1.run	9873	1	--	--	168:0	R	0:0

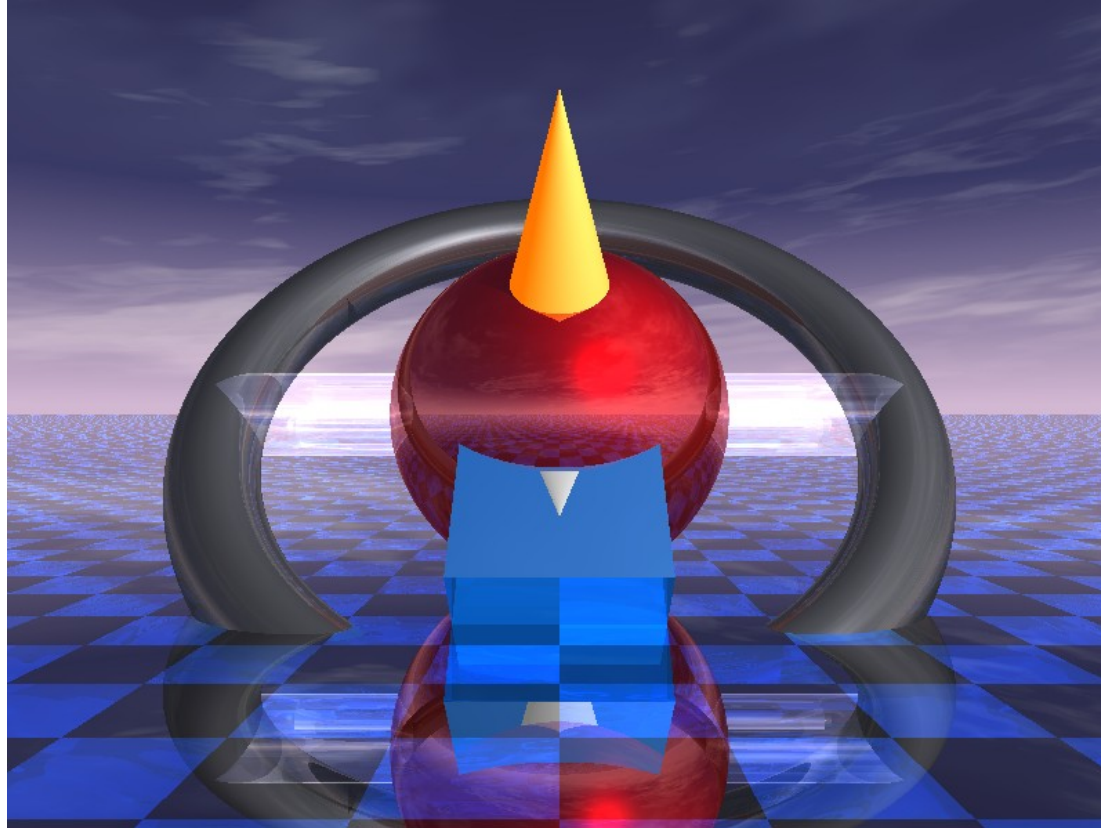
7) výsledné soubory

```
[test1@wolf job1]$ ls
input1.png  input1.pov  job1.run*  job1.run.ces*  job1.run.info
job1.run.stdout
```



System Charon – použití na klastru

8) výsledek (input1.png)





System Charon – použití na klastru

Paralelní úlohy v systému CHARON na klastru

```
[test1@wolf job1]$ cd ~/GridComputing/02.parallel/job1
[test1@wolf job1]$ ls
hello* job1.run*
[test1@wolf job1]$ cat job1.run
```

```
#!/bin/bash
module add mpichrun
mpirun -np $CH_NCPU hello
```

```
[test1@wolf job1]$ psubmit long job1.run 4
[test1@wolf job1]$ pinfo
```

```
:
-----
NCPU           : 4
Resources      : nodes=4:ppn=1:node
Properties     : -none-
Sync mode      : sync
:
```



System Charon – použití na gridu

- stejné příkazy i způsob práce (přenositelnost úloh mezi klastry a gridy)
- potřeba certifikátu
- „2 příkazy navíc“ (inicializace gridového modulu, inicializace certifikátu)

/C=IT/O=GILDA/OU=Personal Certificate/L=Masaryk University/CN=ncbr tester/Email=ncbr@atlas.cz

```
cd ~/GridComputing/04.gilda/job1
module add gilda-wolf
voms-proxy-init --voms gilda
voms-proxy-info --all
psubmit gilda job_script
pinfo
psync
```

- úlohy: ~/GridComputing/04.gilda/job1





System Charon – správa aplikací

Příkazy systému Modulů

- příkaz 'module -h'

module [akce] [modul1 [modul2]]..

- hlavní příkaz systému modulů

akce:

add (nahrátí), remove (odpojení)

avail, list*, active, exported, versions, realizations

disp, isactive

* výchozí akce

modconfig

- konfigurace systému modulů (vizualizace, výchozí moduly, ...)

příklad: **module realizations amber**



System Charon – správa aplikací

- systém modulů – příkaz 'module -h'
- aplikace jsou řazeny do hierarchické struktury

jméno_programu:verze:architektura:paralelní_mod

realizace

- systém automaticky doplňuje možnosti (tabulator)
- nastavení výchozí realizace (default realization)
- **amber** **amber:8.1:auto:auto** **amber:8.1:ipn3:single**

abinit-mp

- * abinit-mp:04.12.14
 - + abinit-mp:04.12.14:i686:node
 - + abinit-mp:04.12.14:i686:p4

amber

- * amber:9.0
 - + amber:9.0:noarch:none
 - + amber:9.0:pn3:single
- * amber:8.1
 - + amber:8.1:noarch:none
 - + amber:8.1:pn3:single

- o konkrétní realizaci se rozhoduje až když je úloha spuštěná na výpočetním uzlu
- systém se snaží vybrat optimální realizaci, pak postupuje hierarchicky

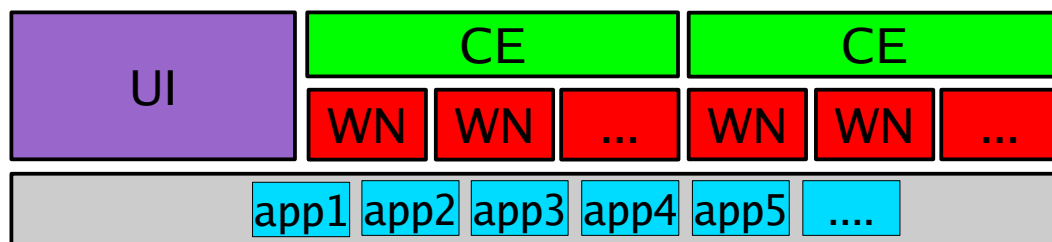


System Charon – správa aplikací

- dva modely „úložiště aplikací“

Model I: METACentrum, většina klastrů

- v klastru (resp. gridu) existuje sdílený disk společný všem výpočetním uzlům

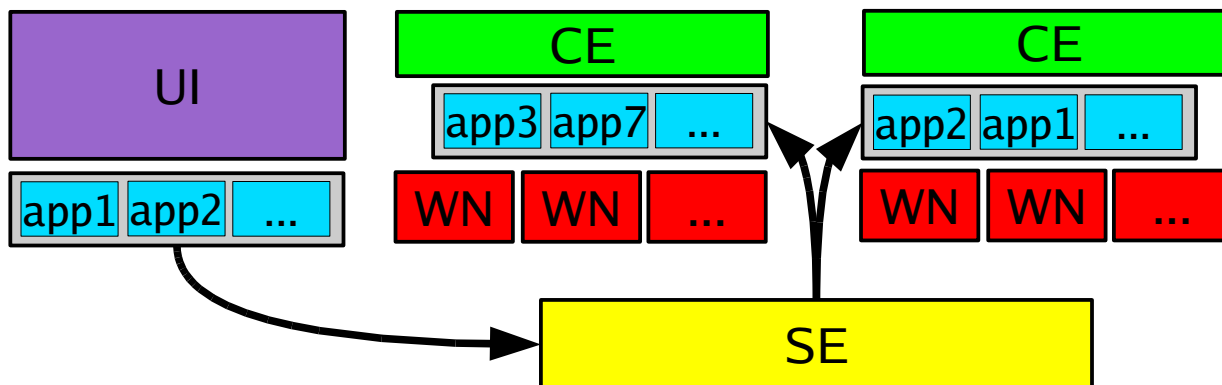


Legenda:

- UI - přístupový počítač
- CE - výpočetní element
- SE - úložiště dat
- WN - výpočetní uzel
- app - aplikace

Model II: EGEE2 GRID

- sdílený disk neexistuje, aplikace se kopírují jednou za čas ze společného SE





Poděkování

- Luděk Matyska (CESNET, ICS)
- Jaroslav Koča (NCBR)
- Evropská komise
 - EGEE II (číslo kontraktu RI-031688)
 - EGEE (číslo kontraktu IST-2003-508833)
- MŠMT (MSM0021622413)
- GAČR (204/03/H016)



Prostor pro dotazy