CERIT-SC, MetaCentrum

Rozvrhový plánovač v CERIT-SC

Václav Chlumský, Dalibor Klusáček

CESNET, z. s. p. o.

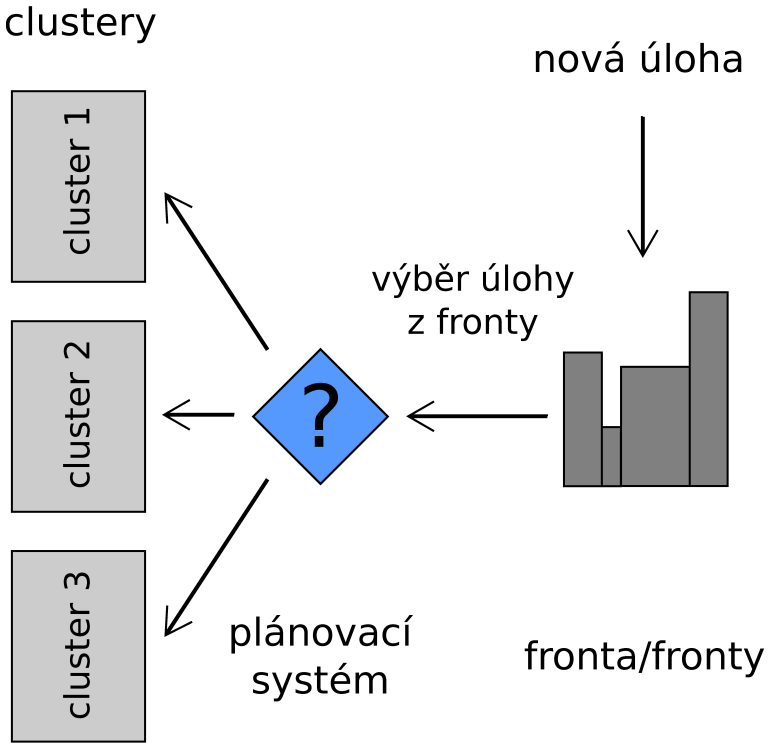
2. 12. 2014

Úvod

- MetaCentrum a CERIT-SC
  - mj. poskytují rozsáhlé výpočetní zdroje
  - efektivní spouštění úloh na těchto zdrojích
    - vysoké využití zdrojů
    - férovost vůči uživatelům
  - zdroje jsou více homogení v CERIT-SC
- nový plánovač v CERIT-SC
  - výzkum a vývoj okolo plánovače několik let
  - vychází z dizertace D. Klusáčka (2011)
    - praktický vývoj v TORQUE součástí diplomové práce (2012)
    - další vývoj pod CESNETem a GAČR projektem P202/12/0306
  - vytváří plán budoucího spuštění úloh
    - optimalizace plánu
  - nasazen v červenci 2014

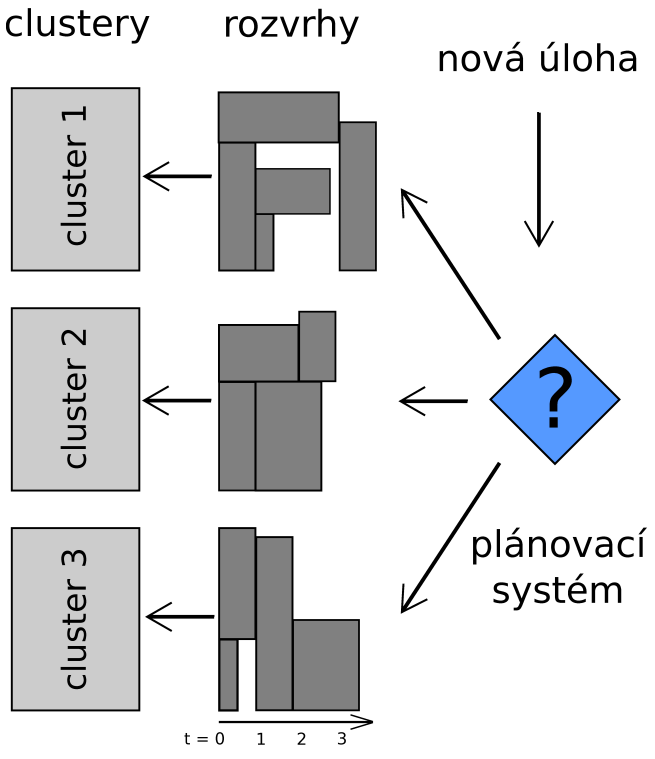
Předchozí přístup k plánování

- úlohy jsou uloženy ve frontách
- rozhodnutí o spuštění úlohy až v poslední okamžik před spuštěním
- obtížně předvídatelné a těžko plánovatelné
- je těžké optimalizovat několik kritérií současně



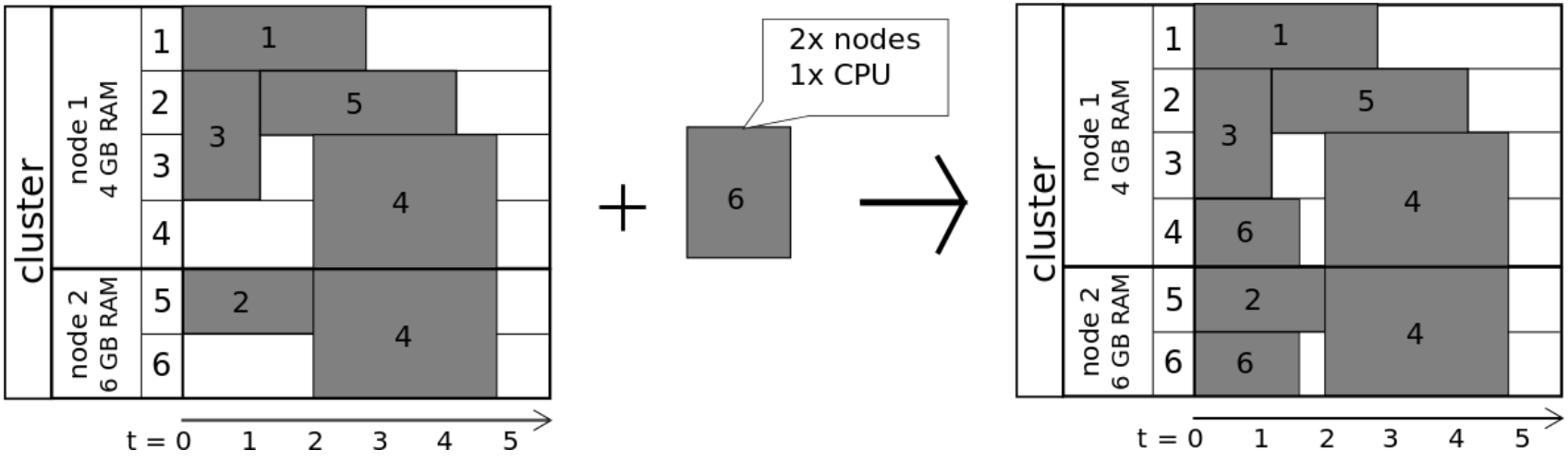
Nový přístup k plánování

- fronty se ignorují (s vyjimkou prioritních front)
- vytváří se budoucí plán spuštění úloh
- umožňuje předvídat kde a kdy bude úloha spuštěna
- plán spuštění je možné ohodnotit a vylepšit jeho kvalitu
- každý cluster má svůj plán
- plán se neustále aktualizuje a *komprimuje*

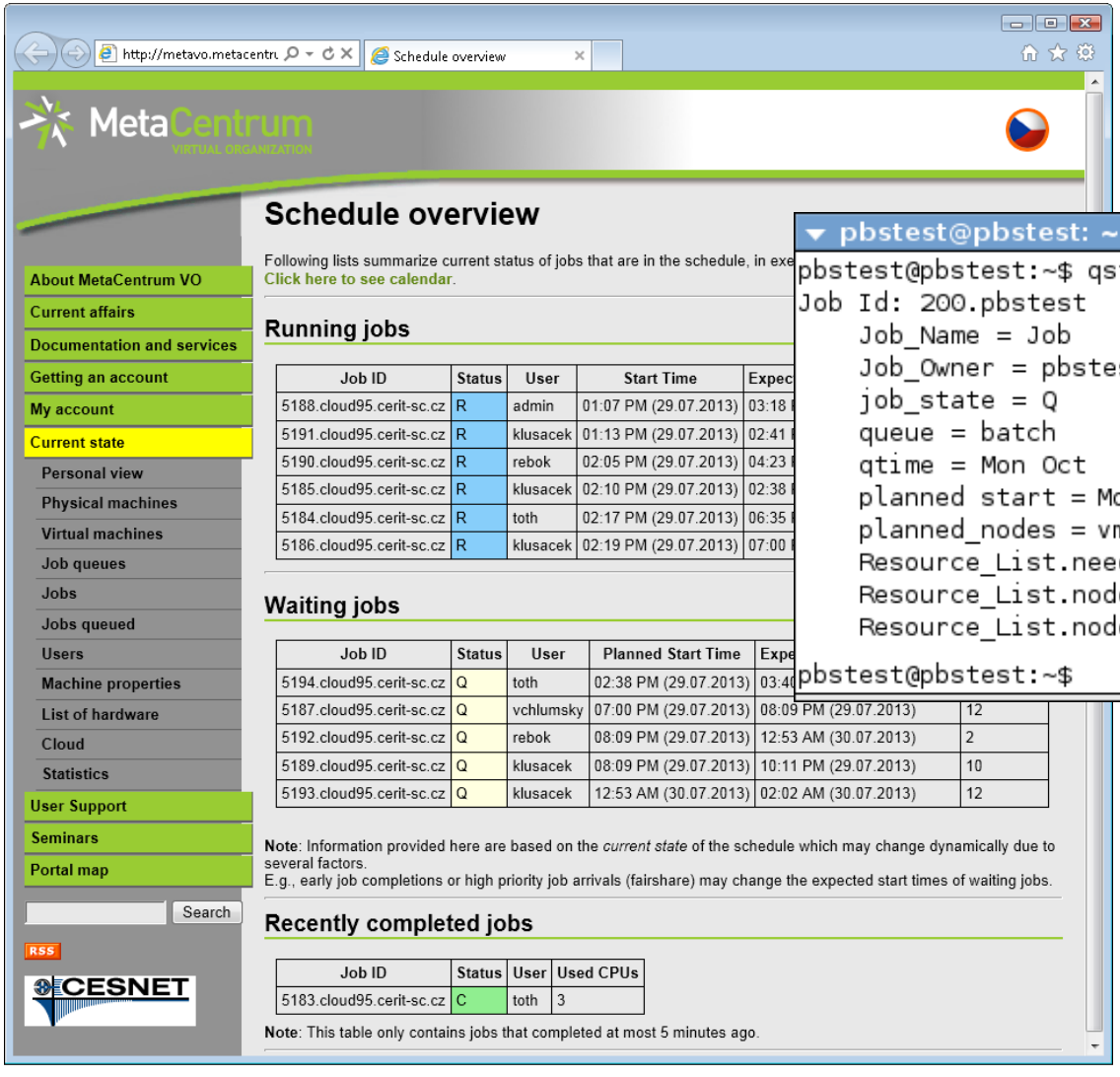


Novinky: backfilling

- plánovač ví, kde je kolik nevyužitých zdrojů, tzv. *díry*
- umožňuje zaplnit tyto *díry* aniž by byl ohrožen plán spuštění dříve naplanované úlohy
- pro novou úlohu plánovač hledá nejdřívější vhodnou *díru*



Novinky: kdy a kde



qstat

```
pbstest@pbstest: ~
pbstest@pbstest:~$ qstat -f 200
Job Id: 200.pbstest
    Job_Name = Job
    Job_Owner = pbstest@pbstest
    job_state = Q
    queue = batch
    qtime = Mon Oct  1 14:44:59 2012
    planned start = Mon Oct  1 15:24:40 2012
    planned_nodes = vm2-82, vm2-78, vm2-79
    Resource_List.neednodes = 3:ppn=1:mem=1kb
    Resource_List.nodect = 3
    Resource_List.nodes = 3:ppn=1:mem=1kb

pbstest@pbstest:~$
```

http://metavo.metacentrum.cz/schedule-overview/

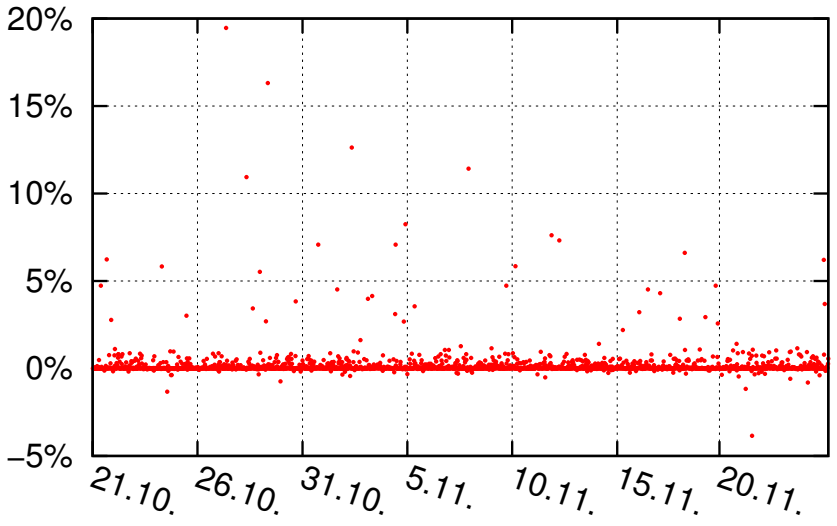
Novinky: optimalizace

- existující plán je možné ohodnotit
- z dvou různých plánu lze rozhodnout, který je podle zvolených kriterií lepší
- optimalizovaná kritéria
  - průměrná doba čekání
  - průměrné zpomalení úlohy
    - zpomalení úlohy $= \frac{\text{jak dlouho je úloha v systému}}{\text{doba výpočtu}}$
  - férovost vůči uživateli v rozvrhu
    - má desetkát větší váhu
- sledované kritérium: průměrný čas odezvy (jak dlouho je úloha v systému)
- změny v rozvrhu jsou náhodné
  - na základě předcházejících experimentů
- přijímají se pouze zlepšující změny

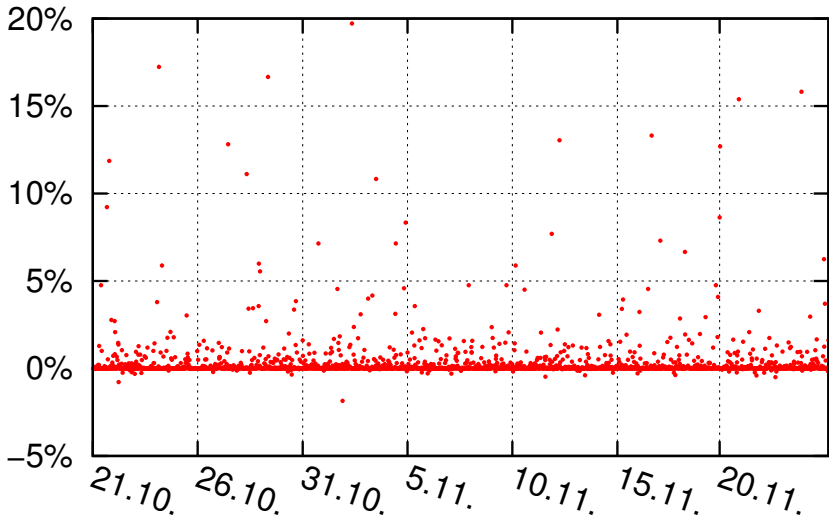
MetaCentrum

Novinky: optimalizace

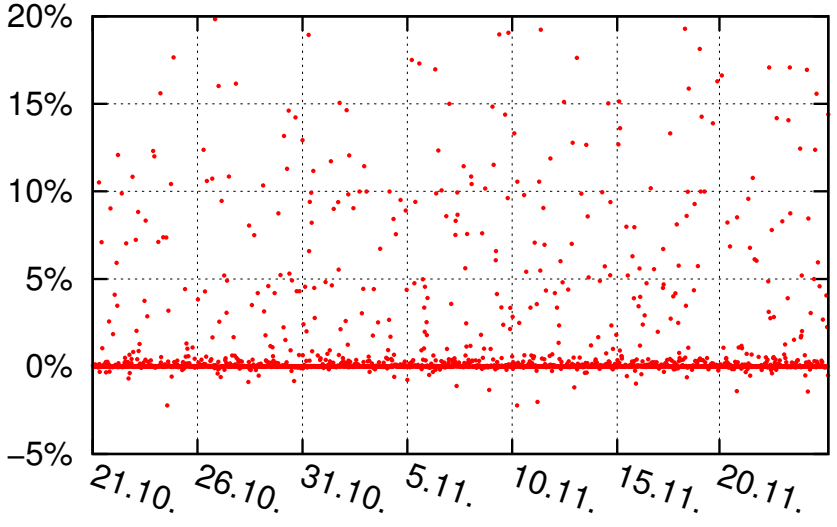
doba čekání



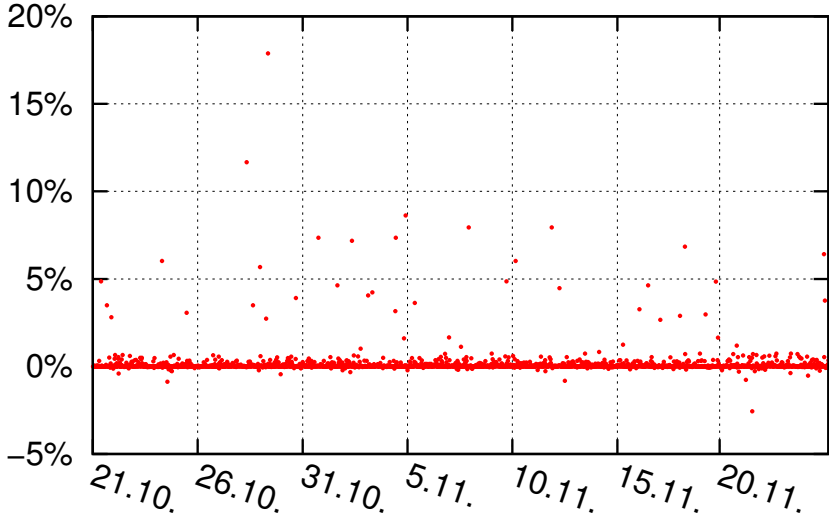
zpomalení



férovost



doba odezvy (nulová váha)



Hezké vlastnosti plánovače

- víme kdy a kde bude úloha spuštěna, uživatel může lépe organizovat svoji práci
- iniciálně naplánovaný čas spuštění úlohy má tendenci se snižovat
- lze vylepšovat plán podle různých kriterií současně, s různou váhou na jednotlivá kriteria
- správce systému může v mnoha případech detekovat potíž pouhým pohledem na plán

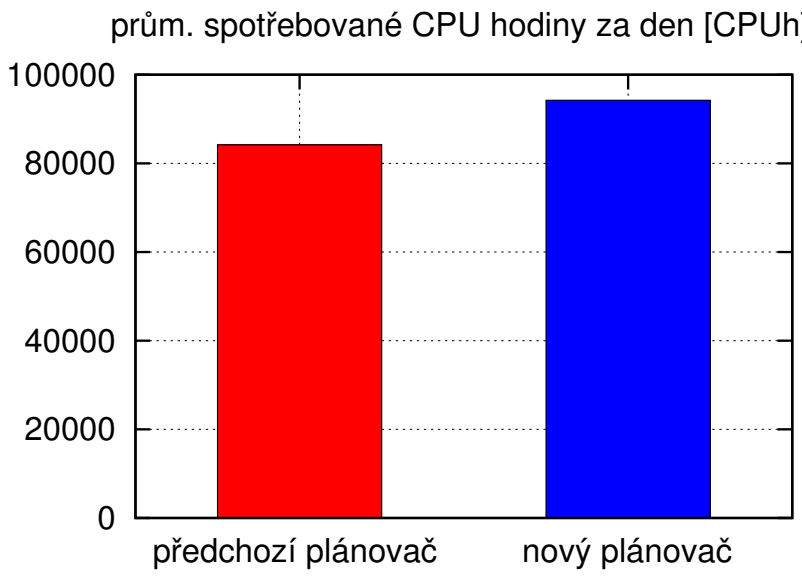
Nepodporované vlastnosti

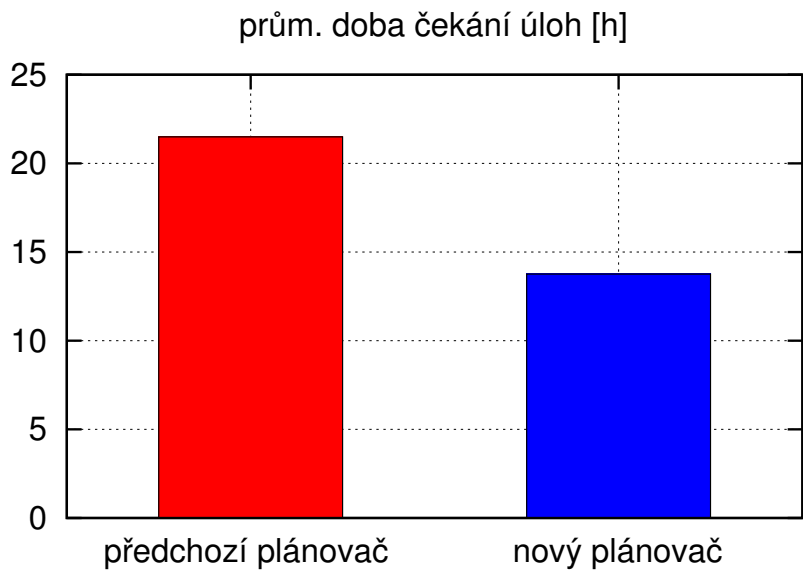
- složité požadavky na uzly, např: *nodes=1:ppn=1+2:ppn=4*
- konkrétní uzel
- zavislá úloha je přidána do plánu až když je závislot splněna
- synchronizované úlohy

Nový vs. předchozí plánovač

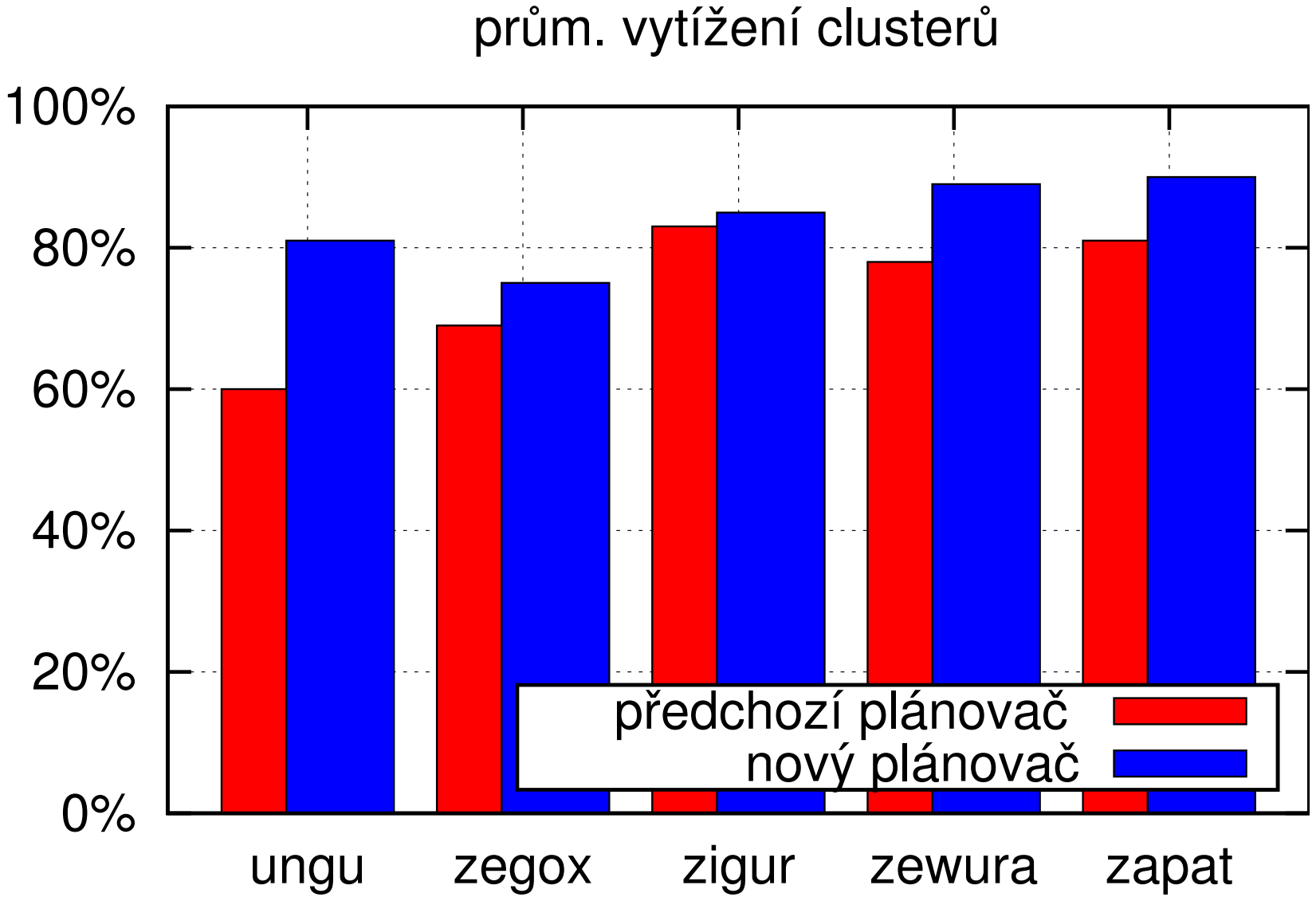
- sledujeme a analyzujeme co se po nasazení děje
- průběžne ladíme plánovač a jeho konfiguraci
- data v následujících grafech jsou za tato období
  - předchozí plánovač: leden – červen 2014
  - nový plánovač: červenec – listopad 2014

Nový vs. předchozí plánovač

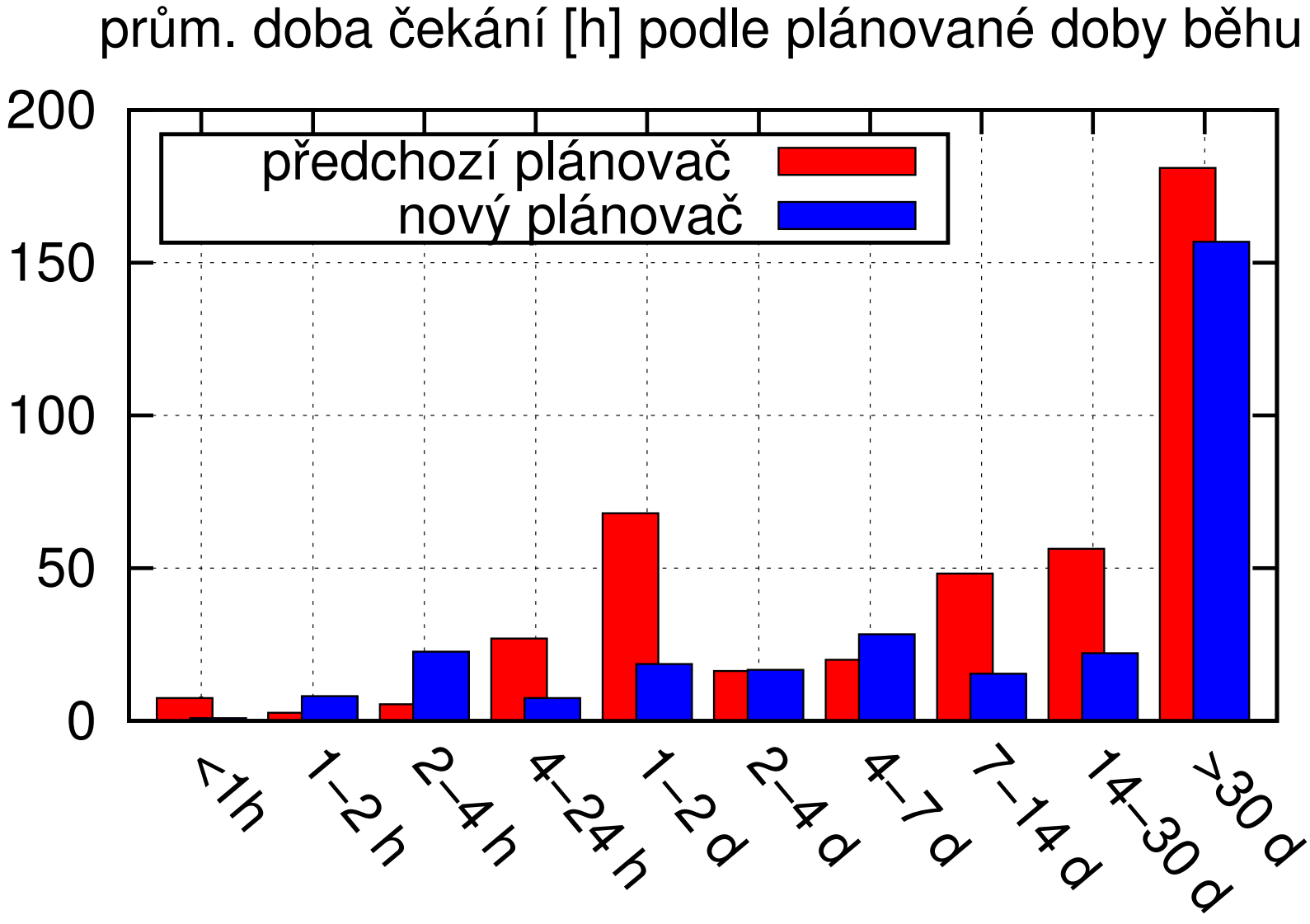




Nový vs. předchozí plánovač

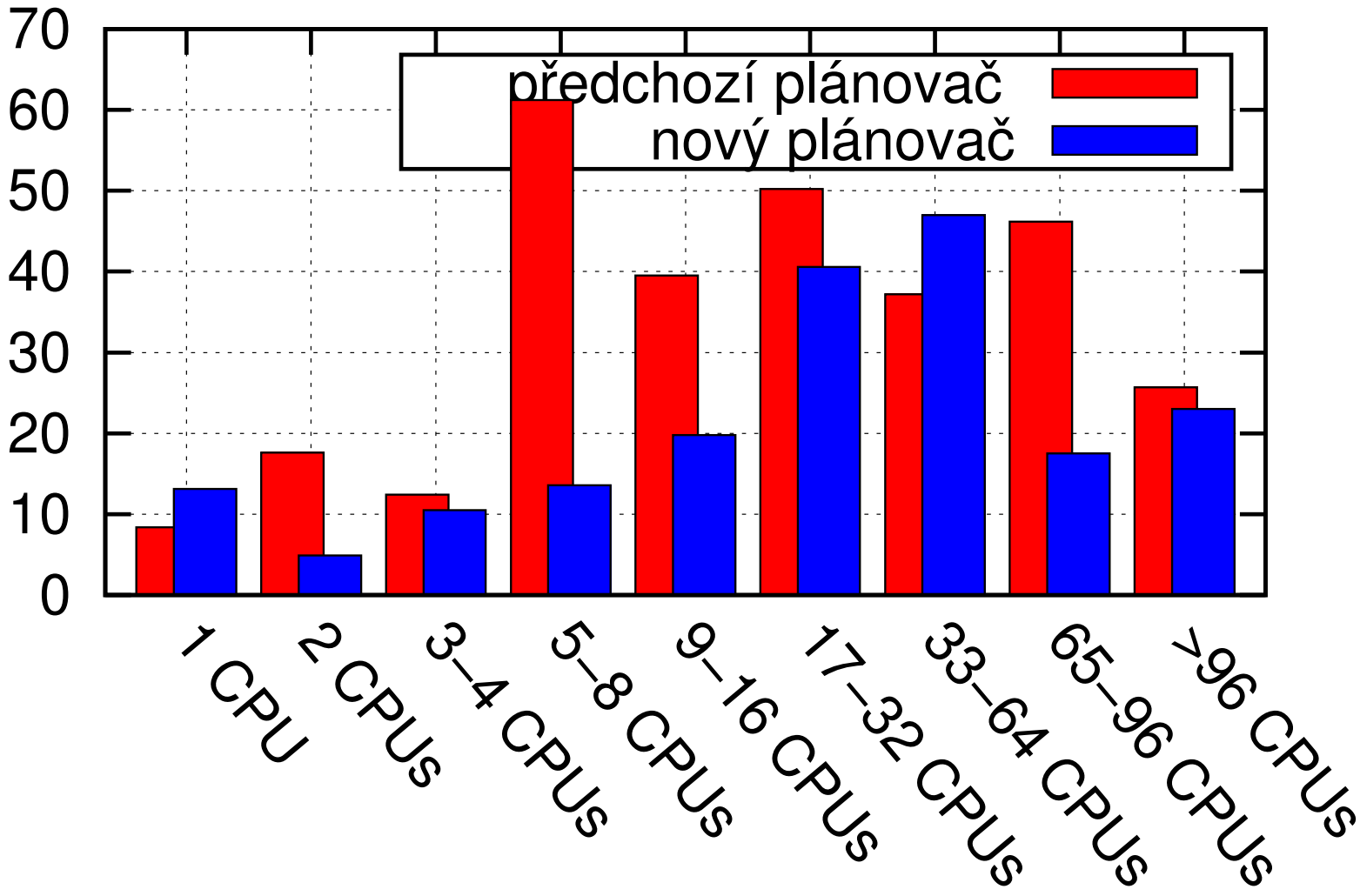


Nový vs. předchozí plánovač



Nový vs. předchozí plánovač

prům. doba čekání [h] podle požadovaných počtu CPU



MetaCentrum

CERIT-SC, MetaCentrum

Tipy pro zadávání úloh

Tipy pro zadávání úloh

- pro časté a pohodlné přihlašování na čelní uzel je vhodné nainstalovat na svoje PC Kerberos
  - řeší opakované zadávání hesla
  - Linux i Windows
- příkaz pro zadávání úloh
  - qsub [-q @server] -l resource_string skript

Výběr serveru

- specifikace serveru (paramer -q)
  - @arien.ics.muni.cz (vychozí pro: skirit, tarkil, nympha, hermes, minos, perian)
  - @wagap.cerit-sc.cz (vychozí pro: zuphux)
- NEspecifikovat frontu v MetaCentru ani v CERIT-SC
  - s výjimkou prioritních front

Doba běhu úlohy

- požadovaná doba běhu (parametr -l)
  - -l walltime=[[dny:[hodiny:]]minuty:]sekundy[.milisekundy]
    - -l walltime=10:00 – úloha bude trvat nejvýše 10 minut
    - -l walltime=3:00:00 – úloha bude trvat nejvýše 3 hodiny
  - -l walltime=[1w][1d][1h][1m][1s]
    - -l walltime=1d1h – úloha bude trvat nejvýše 25 hodin
    - -l walltime=4w – úloha bude trvat nejvýše 4 týdny, tj. 28 dnů
  - lze doporučit nadhodnocení 20%–30%
  - čím přesnější jsou odhady doby běhu úloh, tím přesnější je plán v CERIT-SC

Uzly a procesory

- požadavky na uzly a počty procesorů (parametr -l)
  - -l nodes=1:ppn=1 – jeden procesor na jednom stroji
  - -l nodes=1:ppn=16 – jeden stroj s 16 procesory
  - -l nodes=20:ppn=2 – dvacet dvouprocesorových strojů
  - -l nodes=1:ppn=4#excl – exkluzivní přiřazení uzlu s minimálně čtyřmi procesory
    - exkluzivitu si lze představit jako "nafouknutí" úlohy na celý uzel
    - -l nodes=1:ppn=3:cl_zapat#excl
    - -l nodes=1:ppn=16:cl_zapat
  - konkrétní uzel/y (nelze v CERIT-SC)
    - -l nodes=doom2.metacentrum.cz+doom9.metacentrum.cz

Vlastnosti a vyhraditelné zdroje

- vlastnosti
  - uzly mají různé vlastnosti
    - brno, praha, cl_doom, infiniband, ...
    - vyloučení uzlů s vlastností: ^cl_doom
- vyhraditelné zdroje
  - -l zdroj=hodnota
    - příklady zdrojů: city, cluster, home, infiniband, room, ...
    - např.: -l city=plzen
  - -l place=zdroj
    - stejný zdroj, nezáleží na hodnotě
    - např.: -l place=cluster
    - použití vlastnosti infiniband vynutí -l place=infiniband

Scratch

- lokální úložiště dočasných dat na výpočetních uzlech
  - -l scratch=1 – úloha vyžaduje 1 KiB místa
  - -l scratch=1gb – úloha vyžaduje 1 GiB místa
  - -l scratch=10gb:ssd – úloha vyžaduje 10 GiB místa na SSD disku
  - -l scratch=20gb:local – úloha vyžaduje 20 GiB místa na lokálním HDD
  - -l scratch=100gb:shared – úloha vyžaduje 100 GiB sdíleného místa na síťovém disku
  - -l scratch=500gb:first – úloha vyžaduje 500 GiB místa na hlavním výpočetním uzlu (na ostatních nebude scratch alokován)

Scratch a jeho úklid

```
#!/bin/bash
#PBS -N poradkumilovnyjob
#PBS -l nodes=1:ppn=1
#PBS -l mem=500mb
#PBS -l scratch=1gb
trap 'clean_scratch' TERM EXIT
DATADIR="/storage/brno2/home/$LOGNAME/"

cp $DATADIR/vstup.txt $SCRATCHDIR || exit 1
cd $SCRATCHDIR || exit 2

#... vlastní výpočet ...

cp vystup.txt $DATADIR || export CLEAN_SCRATCH=false
```

Další tipy

- notifikace o stavu úlohy e-mailem (parametr -m)
  - n – neposílat žádný e-mail
  - a – notifikace o zrušení úlohy systémem
  - b – notifikace o spuštění úlohy
  - e – notifikace o dokončení úlohy
  - qsub [-q @server] -l resource_string -m abe skript
  - -M e-mail1,e-mail2
- předání proměnných úloze (parametr -v)
  - qsub [-q @server] -l resource_string -v a=1,i=$j skript

Sledování stavu úlohy

- během výpočtu je možné úlohu sledovat
  - když se něco nedaří, nečekat
  - je možné logovat, že se něco (ne)podařilo
- na konkrétním uzlu: */var/spool/torque/spool/*
  - *1234.arien.ics.muni.cz.OU* – standardní výstup
  - *1234.arien.ics.muni.cz.ER* – standardní chybový výstup
- volba -j pro qsub
  - *n* – implicitní nastavení
  - *oe* – chybový výstup do standardního
  - *eo* – standardní výstup do chybového
- omezeno na 1GB

Co když se úloha nespouští?

- překlep v požadovaných vlastnostech
  - qsub nekontroluje, že požadovaná vlastnost existuje
  - např.: cl_capat
- kontrola správného množství požadované paměti
  - překlep v řádech/jednotkách
- plánovaný výpadek
  - např. nelze naplánovat měsíční úlohu, pokud má být požadovaný cluster za 14 dní odstaven kvůli udržbě

Jak urychlit spuštění úlohy?

- nejsnadněji se spustí úzká a krátká úloha
- zbytečně nenadhodnocovat požadavky
- přesnější odhad doby běhu
- požadovat pouze skutečně potřebné vlastnosti
- pokud je to možné, použít -l place=zdroj
- v CERIT-SC je snažší spustit nodes=2:ppn=1 než nodes=1:ppn=2
  - ale pozor