



HPC v praxi

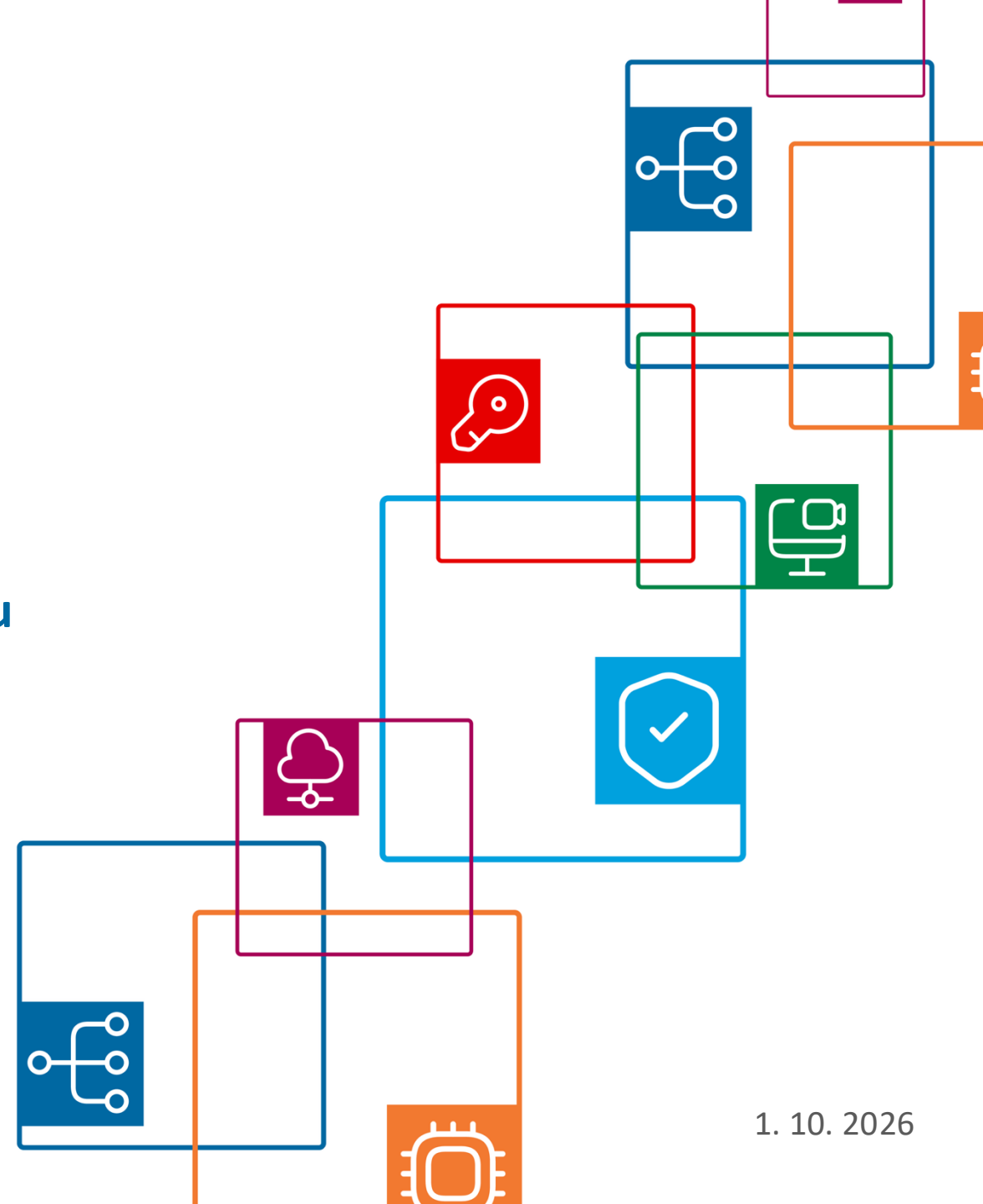
- Jak a proč plánujeme výpočty v MetaCentru
- Jak pomoci vlastním výpočtům

Dalibor Klusáček, Václav Chlumský

Konference e-infrastruktury CESNET 2026

Praha

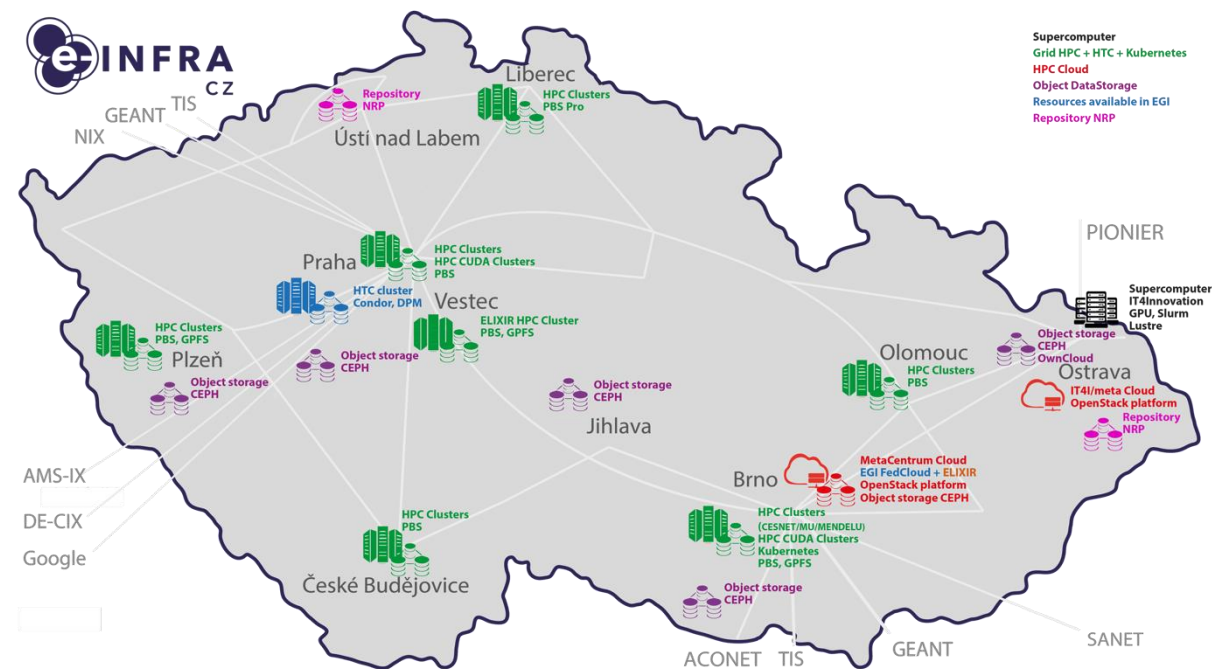
#PROPOJUJEME VĚDU



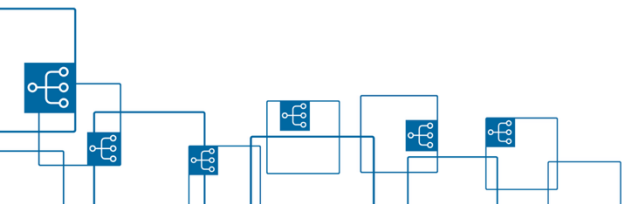
1. 10. 2026

Základní pojmy – letem světem

- MetaCentrum má heterogenní a distribuované výpočetní zdroje
 - 10 organizací, 50 clusterů, 581 uzlů, 42 380 CPU, 342 GPU
 - Různí vlastníci mají různé politiky přístupu na jimi vlastněné zdroje
 - Centralizované plánování pomocí OpenPBS

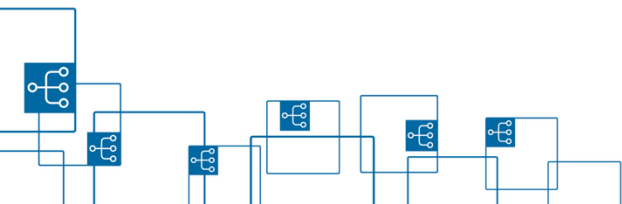


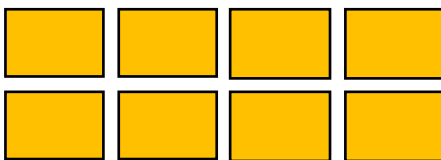
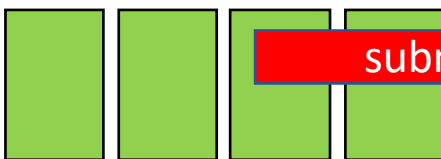
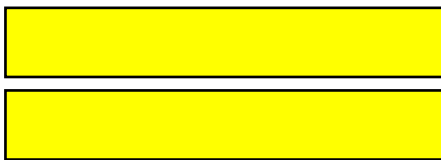
- Naplnit očekávání **uživatelů** ale i **vlastníků** zdrojů
- Pár příkladů:
 - Uživatelé typicky „nechtějí čekat“
 - Když už musí čekat, tak chtějí férové zacházení
 - Vlastníci chtějí dobré využití
 - Někteří pak požadují prioritní přístup pro své „domácí“ uživatele
 - Kladen důraz na využití drahých GPU
 - Krátké úlohy jsou „lepší“ než dlouhé (kvůli hrozbě blokování zdrojů)



■ OpenPBS

- „Operační systém“ infrastruktury
- Rozhoduje kdy, kde, co a komu se spustí
 - Primárně očekává dávkové (batch) úlohy, běží bez zásahu uživatele
 - Interaktivní práce je možná, ale není stěžejní
- Plánovací mechanismus
 - Prioritní fronty („normální“ vs. „důležité“)
 - Plánovací algoritmus (efektivní zaplňování uzlů)
 - Dynamická priorita řazení úloh (férovost)



 User A User B User C

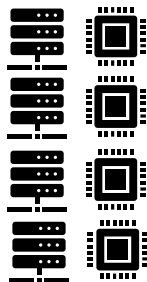
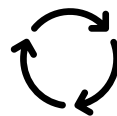
submit úloh



fronta



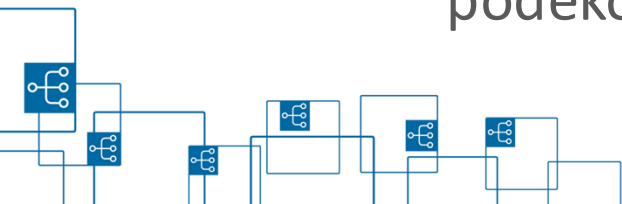
scheduling

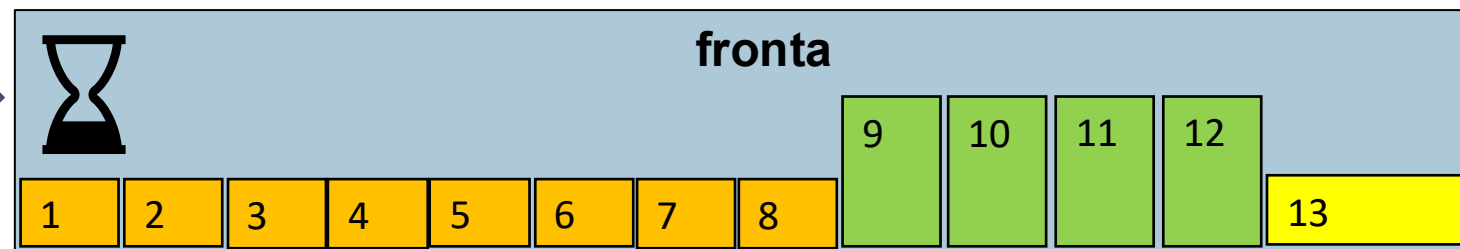
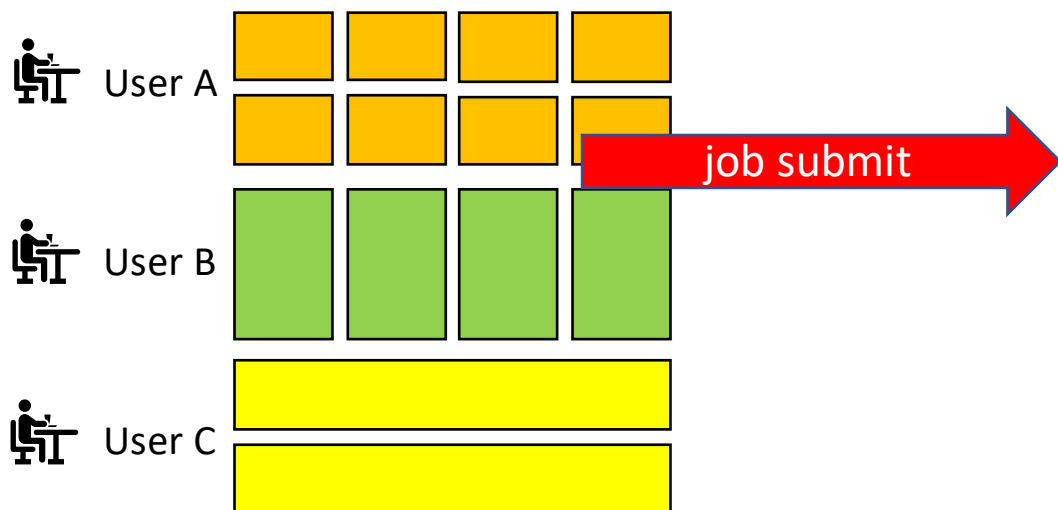


stroje

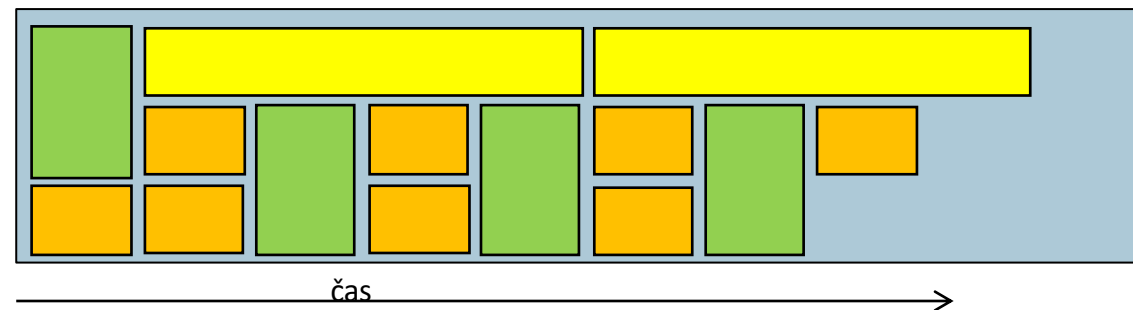
čas

- Každému vždy stejně?
 - Dedikovaný zlomek kapacity (absolutně fér, ale neefektivní)
- Každému všechno, ale jen někdy?
 - Princip kalendáře: ráno počítá Jana, po obědě Andrea, večer Jiří... (dtto)
- Každému přidělit „budget“, který čerpá?
 - Zkušenosti ukazují, že všichni chtějí čerpat budget v jeden moment
- V MetaCentru používáme tzv. **fairsharing**
 - Štědré limity, **vyvažování spotřeby v čase** (budget se obnovuje), bonusy za poděkování





Férové plánování (uživatelé A, B a C mají podobnou spotřebu)

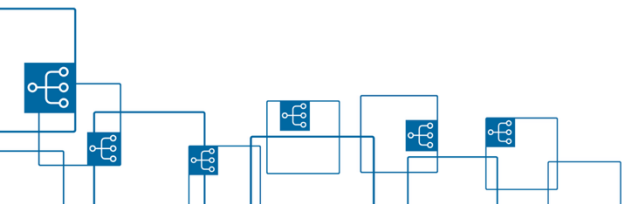
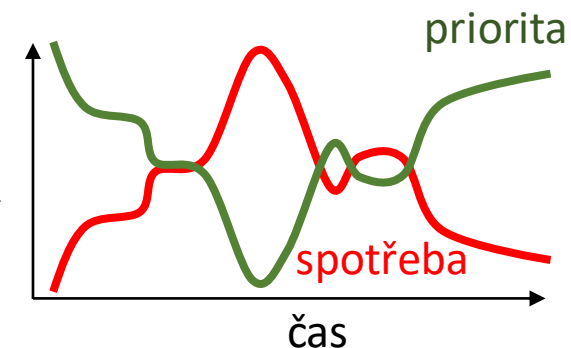


■ Max-Min přístup

- Kdo hodně počítal (nedávno), má nízkou prioritu a naopak

■ V praxi složitější

- Pro každého uživatele započítáváme **přidělené zdroje**
 - Uvažujeme nejen CPU čas ale i RAM, GPU a scratch
 - Zohledňujeme, zda byl použit pomalý nebo rychlý stroj
 - Rychlý (drahý) stroj = kratší výpočet
- Periodicky „umazáváme“ 25% celkové spotřeby, abychom zohlednili stárnutí
- Dáváme bonusové zvýhodnění za publikace s poděkováním MetaCentru
 - Celočíselný faktor, kterým se „boostuje“ váš relativní přiděl zdrojů



■ Princip výpočtu:

1) Spočtení spotřeby zdrojů

2) Výpočet priority (Fairshare Factor)

$$PE(r, j) = \left(\frac{used_resc_r(j)}{total_resc_r} \right) \cdot total_cpus$$

$$usage(job_j) = \left(\sum_{r \in R} PE(r, j) \right) \cdot runtime(job_j)$$

$$usage(User_i) = \sum_{j \in [jobs\ of\ User_i]} usage(job_j)$$

$$usage_{effective}(User_i) = \frac{usage(User_i)}{usage(total)}$$

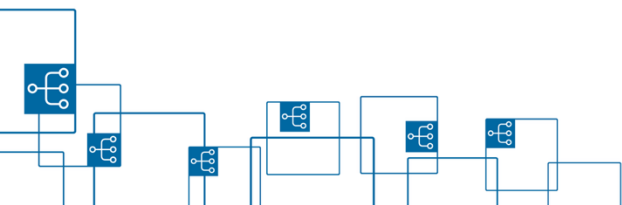
$$usage_{target}(User_i) = \frac{shares(User_i)}{shares(total)}$$

$$FF(User_i) = 2^{-\left(\frac{usage_{effective}(User_i)}{usage_{target}(User_i)} \right)}$$

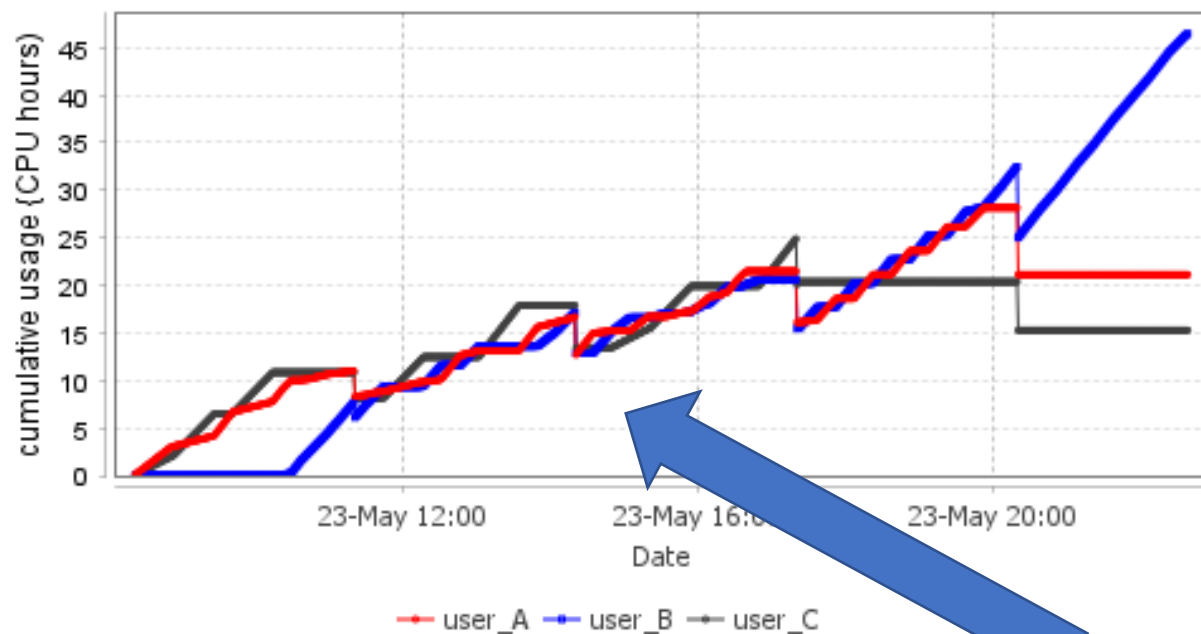


Výsledná dynamická priorita

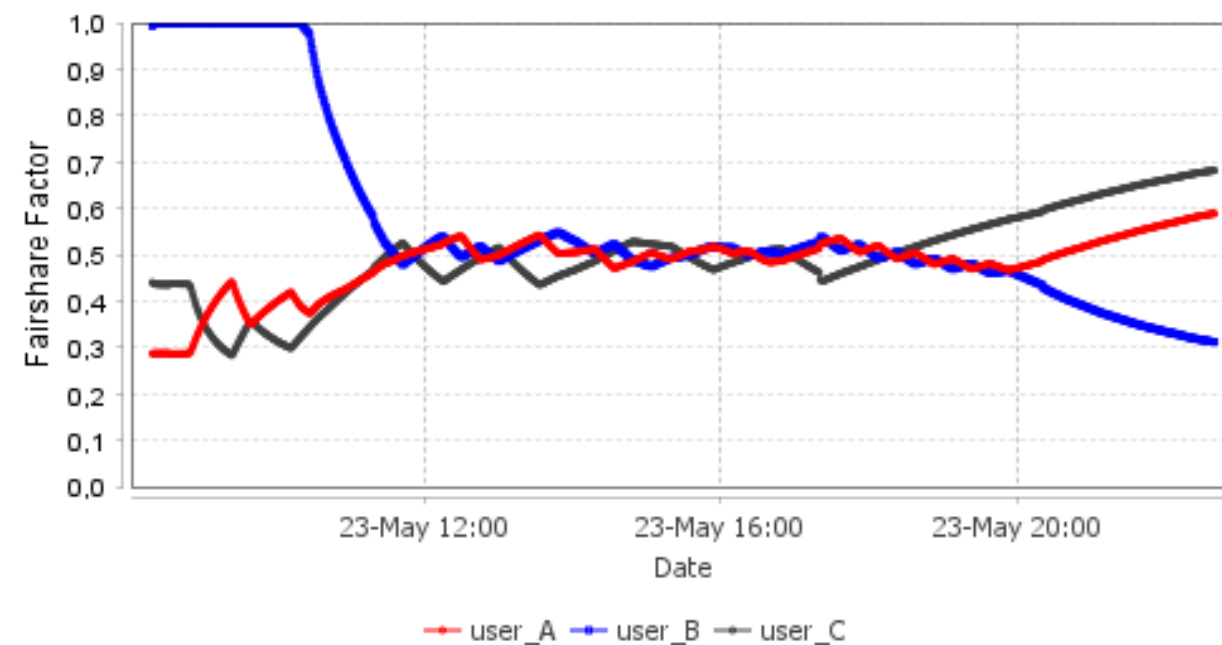
- **Dynamická priorita ve frontě**
 - S každým plánovacím cyklem se **přepočítává priorita všech uživatelů**
 - Fronta se dynamicky seřadí dle priorit uživatelů (sestupné pořadí)
 - A poté se zkouší spustit úlohy v tomto pořadí
- **Neúspěšné (velké) úlohy z čela fronty dostávají rezervaci**
 - **Volná místa se zaplňují vhodnými úlohami** (tzv. backfilling)
 - Přičemž **výplňové úlohy nesmí kolidovat s rezervacemi** čelních úloh
- **Kompromis mezi striktním požadavkem férovosti a využitím systému**



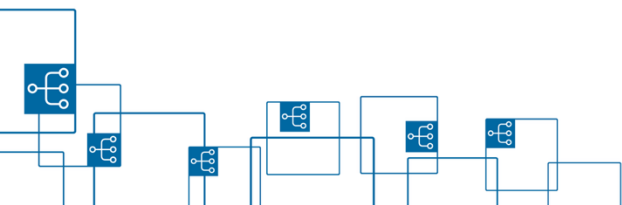
Kumulativní spotřeba (propočítaný čas)



Dynamická priorita uživatelů (Fairshare Factor)



Dokud mají všichni uživatelé úlohy, spotřeba je vyvážená



Jak si mohu jako uživatel pomoci?

Osvědčené „best practices“

- Klíčové jsou **prior**
- Svou **prioritu** nej...
publikacích
 - A reportováním
 - <https://publicati>

cesnet

metacentrum

klusacek

Můj přehled

Sestavovač úloh (qsub)

Stav zdrojů

Stroje MetaCentra

Úložné prostory

Projekty v cloudu

Fronty úloh

Úlohy

Detail uživatele

Základní informace

Uživatelské jméno	Celé jméno
klusacek	RNDr. Dalibor Klusáček Ph.D.
Organizace	Datum vypršení členství
CESNET, z. s. p. o.	02.02.2027
Preferovaný e-mail	Preferovaný jazyk
klusacek@cesnet.cz	CS

Publikace

CERIT-SC: 3

FI MU: 1

MetaCentrum: 14

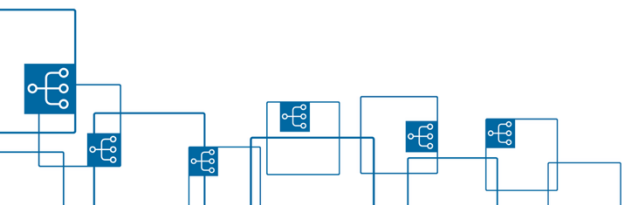
e-INFRA CZ: 8

Skupiny

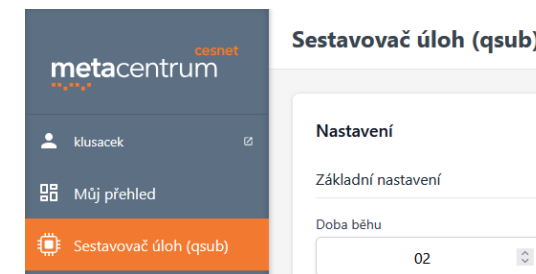
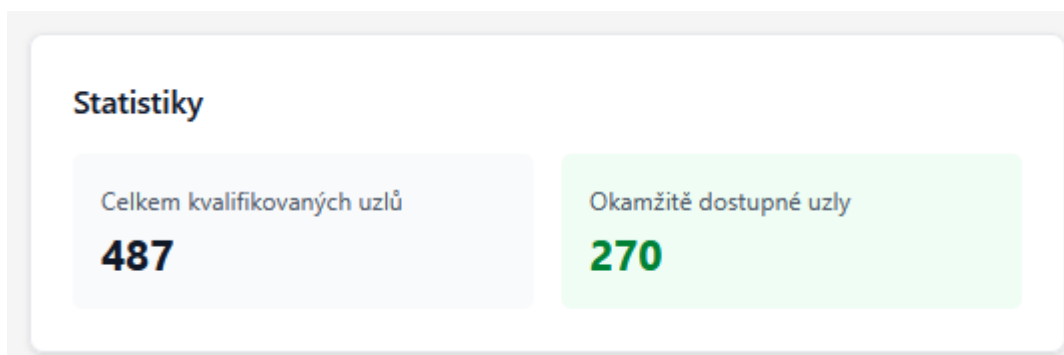
metasw

Fairshare podle serveru

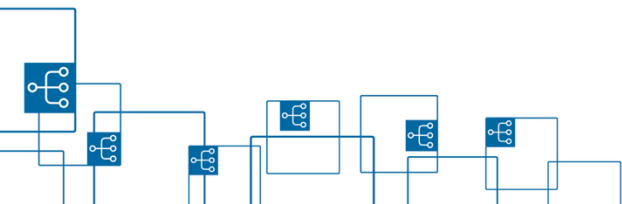
SERVER	PRIORITA
pbs-m1	95 uživatelů má před vámi prioritu

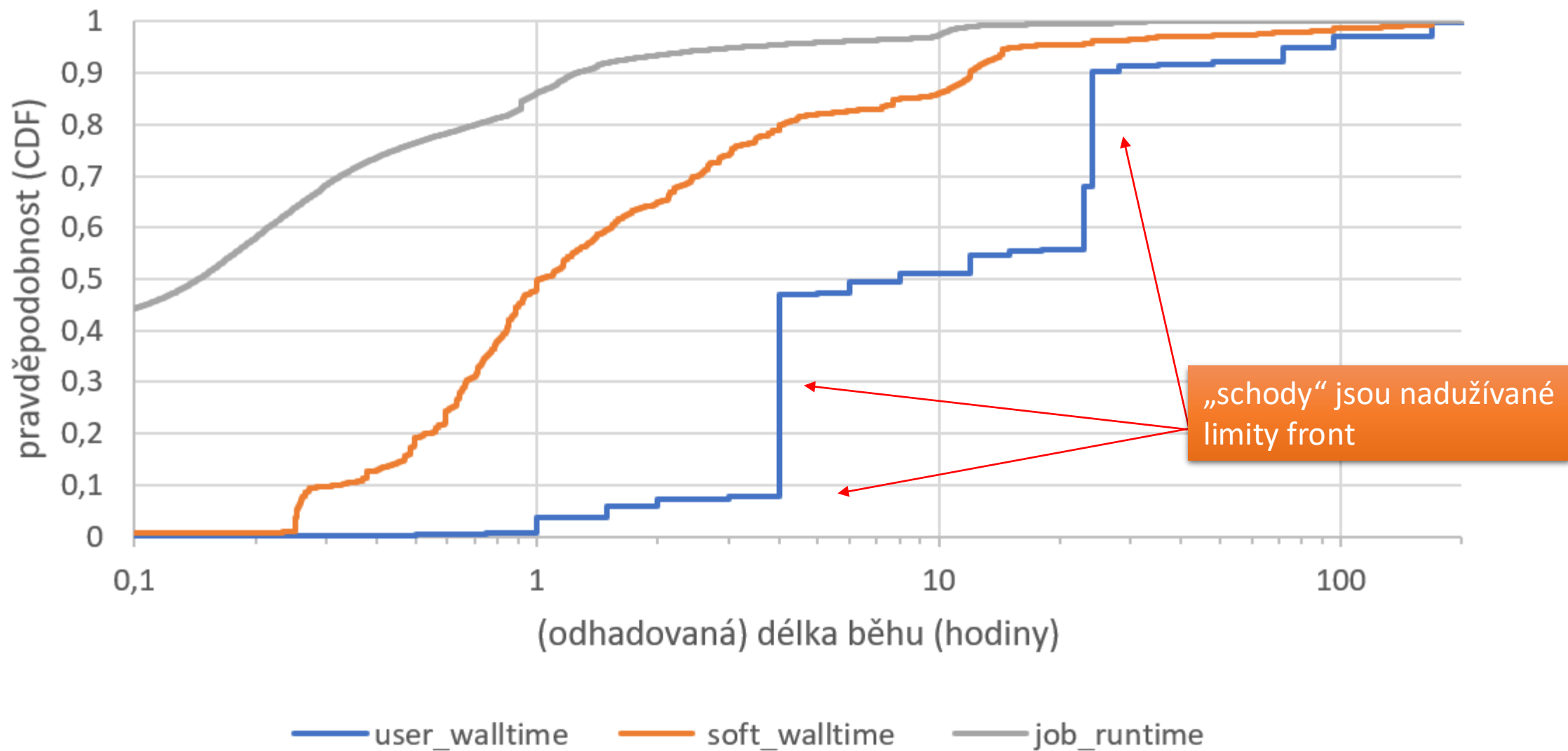


- Délka úloh
 - Obecně **je mnohem více strojů pro kratší úlohy**
 - 35k jader pro krátké úlohy
 - Jen cca 16k jader pro měsíční úlohy
- Pokud si nejste jistí, použijte sestavovač qsub příkazu
 - <https://my.metacentrum.cz/qsub-assembler>



- Walltime určuje, v jaké frontě úloha skončí
 - Čím kratší, tím větší šance na více volných strojů
 - A také **větší šance, že bude moci zaplnit „díru“ v rozvrhu**
- Je v pořádku jej nadhodnotit
 - Ale uživatelé typicky zadávají jen 2h, 4h, 24h, 48h atd.
 - Tím si škodí, **všechny úlohy pak pro plánovač vypadají „stejně“**
 - My se to snažíme vylepšovat (tzv. soft-walltime)
 - **Automatický prediktor**, založený na historických datech daného uživatele



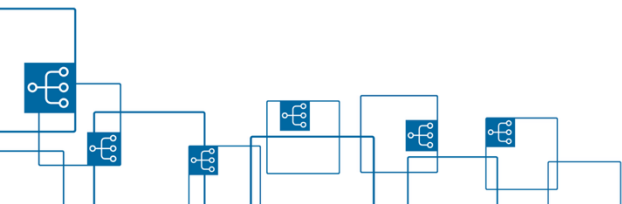


- Šířka úloh

- Úzké (a krátké úlohy) se obecně snáze spouštějí
- Vyplňují volná místa a nepřekáží rezervacím

- **Nikdy nekombinujeme „široké“, náročné úlohy a krátké/malé**

- Naše vlastní úlohy předběhnou naši velikou úlohu
- Zruší její rezervaci (fairshare nás dočasně odsune z čela fronty)
- **Velká úloha se nemusí nikdy spustit!!!**

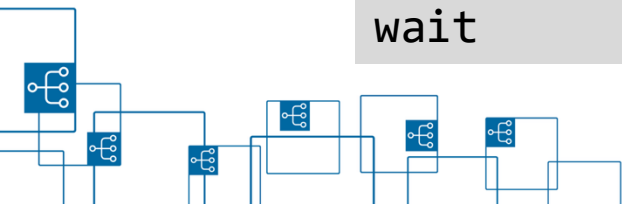


- **OpenPBS limituje max. počet úloh a procesorů pro 1 uživatele**
 - Max. 500 běžících úloh (7000 ve frontách)
 - Max. 4000 CPU
- Pokud uživatel nemá dostatečně paralelní úlohy, lze je „obalit“

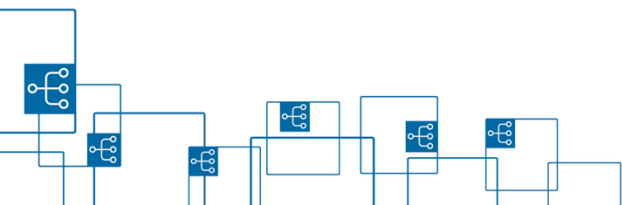
```
qsub -l select=1:ncpus=8:mem=400mb -l walltime=00:02:00 job.sh
```

```
#!/bin/bash

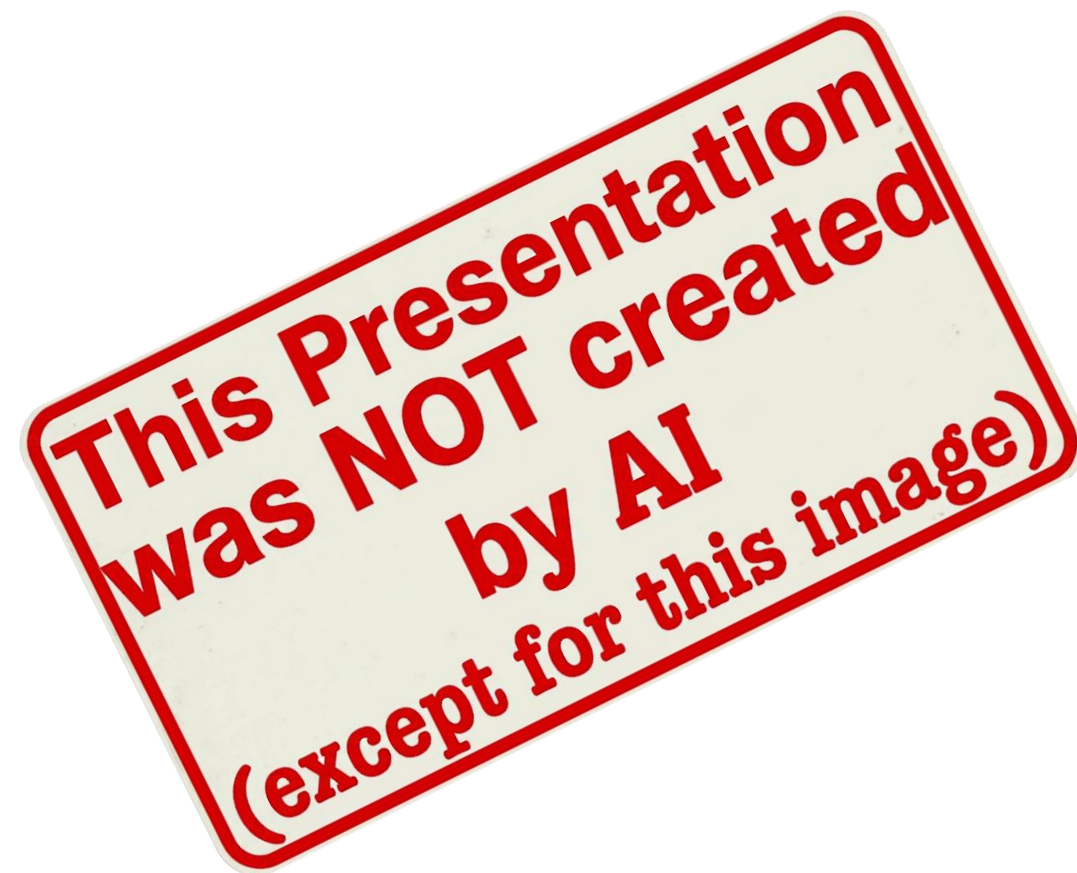
for i in $(seq 1 $PBS_NCPUS); do
    python sequential_computation.py $i &
done
wait
```



- Když všechno selže, můžete se na nás obrátit s žádostí o rezervaci
 - Funguje to jako fronta, která je přístupná pouze vám
 - A má vyhrazené zdroje
- Žádost je potřeba zdůvodnit, využití rezervace se kontroluje



klusacek@cesnet.cz



- Plánovací politiky se testují pomocí pokročilých simulátorů
 - AleaNG: simulátor plánování úloh
 - fairshareSim: simulátor fairshare
 - <https://github.com/aleasimulator/>
- Ve srovnání se zahraničními výpočetními centry si vedeme dobře
 - Reálné využití systému (velká centra často plýtvají)
 - Netriviální heterogenita systému
 - Podpora férovosti a důraz i na „běžné“ uživatele
- Naše zkušenosti a výsledky dlouhodobě publikujeme
 - Např. Job Scheduling Strategies for Parallel Processing (30. ročník)

