

# MetaCentrum & CERIT-SC

**Tomáš Rebok**

MetaCentrum, CESNET z.s.p.o.

CERIT-SC, Masarykova univerzita

(rebok@ics.muni.cz)

---

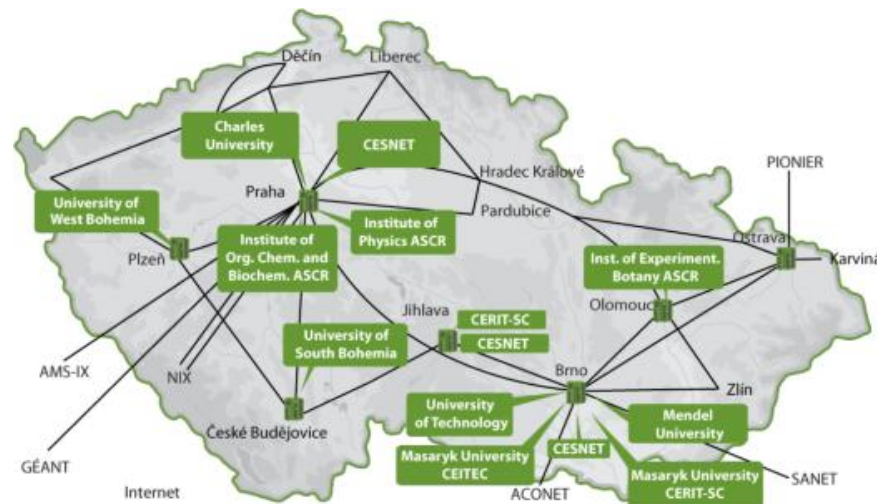
# Obsah

- Výpočetní služby
- Služby pro podporu vědy a výzkumu
- Úložné služby
- Služby pro podporu vzdálené spolupráce
- Další podpůrné služby
  
- Školící hands-on seminář

# Výpočetní služby

# MetaCentrum @ CESNET

- aktivita sdružení CESNET
- od roku 1996 **koordinátor Národní Gridové Infrastruktury**
  - integruje velká/střední HW centra (clustery, výkonné servery a úložiště) několika univerzit/organizací v rámci ČR
    - prostředí pro (spolu)práci v oblasti výpočtů a práce s daty
  - integrováno do **evropské gridové infrastruktury (EGI)**



# Výpočetní cluster

- skupina vzájemně propojených „běžných“ počítačů



(dříve)

# Výpočetní cluster

- skupina vzájemně propojených „běžných“ počítačů



(nyní)

# MetaCentrum NGI

- **koordinátor národního gridu**
- pokud jste/budete vlastníci HW zdrojů, NGI Vám může pomoci s:
  - *nákupem a integrací vlastních zdrojů (existujících i plánovaných) do gridového prostředí (**slabá vs. silná integrace**)*
    - pomoc při výběru, instalaci a provozu clusterů, jednotná správa systémového a aplikačního SW
    - správa účtů, systém pro správu úloh
    - společný provozní dohled, přizpůsobení místním potřebám
    - priorita nebo výhradní přístup na své zdroje
- **uživatelé sdružováni do tzv. virtuálních organizací**
  - = skupina uživatelů majících „něco společného“

# MetaCentrum VO (Meta VO)

- **přístupné zaměstnancům a studentům VŠ/univerzit, AV ČR, výzkumným ústavům, atp.**

- komerční subjekty pouze pro veřejný výzkum

- **nabízí:**

<http://metavo.metacentrum.cz>

- **výpočetní zdroje**
- **úložné kapacity**
- **aplikační programy**

- **po registraci k dispozici zcela zdarma**

- „placení“ formou **publikací s poděkováním**

- prioritizace uživatelů při plném vytížení zdrojů





# MetaVO – základní charakteristika

- **po registraci zdroje dostupné bez administrativní zátěže**
  - → ~ okamžitě (dle aktuálního vytížení)
  - **žádné žádosti o zdroje**
- **každoroční prodlužování uživatelských účtů**
  - periodická informace o **trvajícím akademické příslušnosti uživatelů**
    - využití infrastruktury eduID.cz pro minimalizaci zátěže uživatele
  - **oznamování publikací s poděkováním MetaCentru/CERIT-SC**
    - doklad pro žádosti o budoucí financování z veřejných zdrojů
- **best-effort služba**

# Meta VO – dostupný výpočetní hardware

- **výpočetní zdroje: cca 9770 jader (x86\_64)**
  - uzly s nižším počtem výkonných jader: 2x4-8 jader
  - uzly se středním počtem jader (SMP stroje): 32-80 jader
  - paměť až 1 TB na uzel
  - uzel s vysokým počtem jader: 288 jader, 6 TB operační paměti
  - další „exotický“ hardware:
    - uzly s GPU kartami, SSD disky, ...

# Meta VO – dostupný úložný hardware

- **cca 1 PB (1063 TB) pro pracovní data**

- úložiště 3x v Brně, 1x v Plzni, 1x v ČB, 1x v Praze, 1x v Jihlavě, 1x v Ostravě

- uživatelská kvóta 1-3 TB na každém z úložišť

- **cca 3,8 PB (plán cca 16 PB) pro archivní data**

- uživatelská kvóta 5 TB

# Meta VO – dostupný software

- **~ 220 různých aplikací**
  - viz <http://meta.cesnet.cz/wiki/Kategorie:Aplikace>
- **průběžně udržované vývojové prostředí**
  - GNU, Intel, PGI, ladící a optimalizační nástroje (TotalView, Allinea), ...
- **generický matematický software**
  - Matlab, Maple, gridMathematica, ...
- **komerční i volný software pro aplikační chemii**
  - Gaussian 09, Gaussian-Linda, Gamess, Gromacs, ...
- **materiálové simulace**
  - Wien2k, ANSYS Fluent CFD, Ansys Mechanical...
- **strukturní biologie, bioinformatika**
  - CLC Genomics Workbench, Geneious, Turbomole, Molpro, ...
  - řada volně dostupných balíčků
- ...

# Meta VO – výpočetní prostředí

- **dávkové úlohy**

- popisný skript úlohy
- oznámení startu a ukončení úlohy

- **interaktivní úlohy**

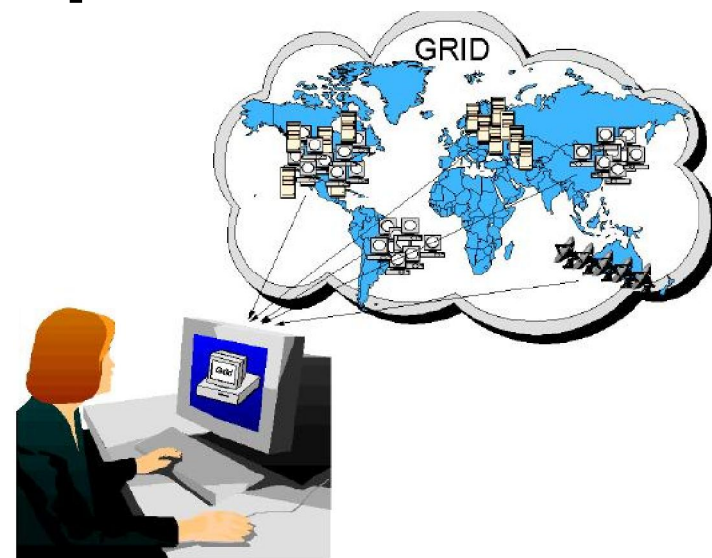
- textový i grafický režim

- **cloudové rozhraní**

- základní kompatibilita s Amazon EC2
- uživatelé nespouští úlohy, ale virtuální stroje

opět zaměřeno na vědecké výpočty

možnost vyladit si obraz a přenést ho do MetaCentra/CERIT-SC (Windows, Linux)



# Meta VO – cloudové služby I.

- **využití virtualizace:**
  - **výhody:** plná kontrola na úrovni OS, realizace výpočtu plně na uživateli
  - **nevýhody:** vhodné pro nasazení menšího rozsahu
- **poskytovány předpřipravené virtuální obrazy, možnost vlastních obrazů (Windows, Linux)**
- **primárně určeno pro testování a výpočty, nikoli pro webhostingové služby**
  - výpočty, testy, výzkum, vývoj, ...



# Meta VO – jak se stát uživatelem?

- ***podejte si přihlášku***

- <http://metavo.metacentrum.cz> , sekce „Přihláška“
- EduID.cz => **ověření Vaší akademické identity** proběhne s využitím Vaší domovské instituce

- ***seznamte se s dokumentací a základy OS Linux***

- <http://metavo.metacentrum.cz> , sekce „Dokumentace“
- <http://www.poznejlinux.cz/linuxbook/xhtmll-chunks/ch07.html>
- <http://www.math.muni.cz/~xkuranov/vyuka/tex/p4.html>

- ***počítejte***



# Pozice výpočetních infrastruktur v ČR I.

- *IT4innovations (Ostrava)*

- **3312 výpočetních jader** („malý“ superpočítač/cluster)
- + velký superpočítač (??? jader, provoz 2015)
- parametry:
  - výpočetní čas přidělován **formou výzkumného projektu**
  - nutná **formální žádost** (posuzována vědecká a technická připravenost + finanční participace)
  - **veřejné soutěže** vypisovány 2x ročně
  - v případě akceptace žádosti **snazší dostupnost zdrojů** (minimum souběžně počítajících uživatelů)
- určení:
  - rozsáhlé (odladěné) výpočty na +/- homogenní infrastruktuře

# Pozice výpočetních infrastruktur v ČR II.

- **Národní Gridová Infrastruktura (NGI) MetaCentrum**
  - cca **9770** výpočetních jader (vč. zdrojů CERIT-SC)
  - parametry:
    - výpočetní čas zdarma dostupný bez explicitních žádostí o zdroje
    - dostupnost různých typů HW, včetně „exotického“
    - zdroje sdíleny s ostatními uživateli (občas horší dostupnost)
  - určení:
    - běžné výpočty menšího až středního rozsahu (výpočty většího rozsahu možné jen po domluvě)
    - příprava výpočtů pro počítání na IT4innovations (~ technická připravenost)
- **CERIT-SC @ ÚVT MU**
  - poskytovatel HW a SW zdrojů do produkčního prostředí NGI
  - hlavní důraz na **služby pro podporu vědy a výzkumu**

# Služby pro podporu vědy a výzkumu

# Centrum CERIT-SC

- **výzkumné centrum vybudované na ÚVT MU**
  - transformace Superpočítačového centra Brno (SCB) při Masarykově univerzitě do nové podoby
- **významný člen/partner národního gridové infrastruktury**
  - I. **poskytovatel HW a SW zdrojů**
    - SMP uzly (1600 jader)
    - HD uzly (2624 jader)
    - **SGI UV uzel (288 jader, 6 TB paměti)**
    - úložné kapacity (~ 3,5 PB)
    - SW výbava totožná s MetaVO
  - II. **služby nad rámec „běžného“ HW centra –**  
**zázemí pro kolaborativní výzkum**



# CERIT-SC – cíle Centra

## Hlavní cíle Centra:

### I. Podpora experimentů s novými formami, architekturou a konfiguracemi e-Infrastruktury

- **vysoce flexibilní infrastruktura** (experimentům příznivé prostředí)
- **vlastní výzkum**, zaměřený na principy a technologie e-Infrastruktury a její optimalizaci

### II. Studium a posun možností špičkové e-Infrastruktury úzkou výzkumnou spoluprací mezi informatiky a uživateli takovéto infrastruktury

- výpočetní a úložné kapacity jsou **pouze nástrojem**
- zaměření na **inteligentní a nové** použití těchto nástrojů
  - synergický posun **informatiky a spolupracujících věd (kolaborativní výzkum)**
  - **pro informatiku generování nových otázek**
  - **pro vědy generování nových příležitostí**

# CERIT-SC – kolaborativní výzkum

## *Spolupráce a podpora výzkumu formou:*

- vedení DP a PhD prací studentů FI MU
- vedení/konzultace DP a PhD prací externích studentů
- participace na národních/evropských projektech
  - ELIXIR, ICOS, ...

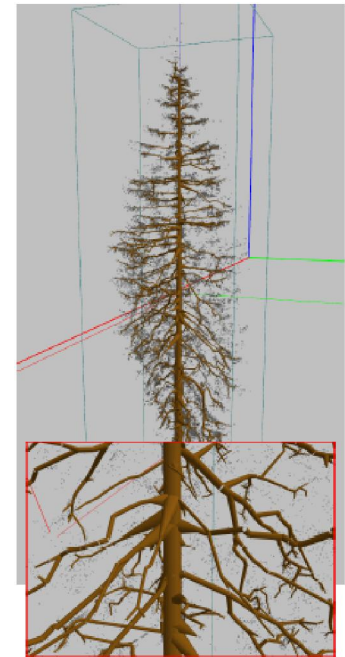
## *Silné odborné zázemí:*

- organizačně **součást Ústavu výpočetní techniky MU**
- dlouholetá tradice **spolupráce s Fakultou informatiky MU**
- dlouholetá tradice **spolupráce se sdružením CESNET**
- SCB (nyní CERIT-SC) je zakladatel MetaCentra

# Příklady spolupráce s partnery I.

## Rekonstrukce stromu z jeho laserového skenu

- partner: *CzechGlobe* (prof. Marek, doc. Zemek, dr. Hanuš, dr. Kaplan)
- **cíl projektu: návrh algoritmu pro rekonstrukci stromu (smrků)**
  - z mraku nasnímaných 3D bodů
    - strom nasnímán laserovým snímačem LIDAR
    - výstupem jsou souřadnice XYZ + intenzita odrazu
  - *očekávaný výstup*: 3D struktura popisující strom
- **hlavní problémy**: překryvy (mezery v datech)



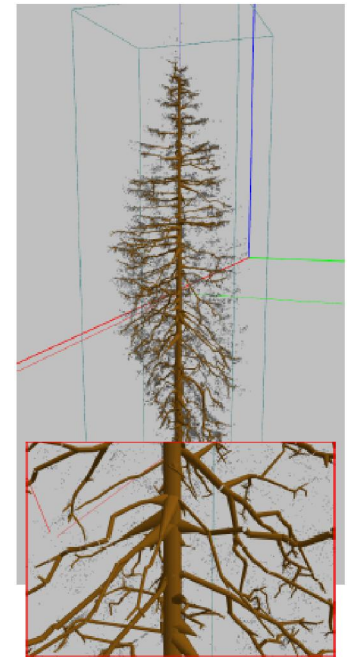
# Příklady spolupráce s partnery I.

## Rekonstrukce stromu z jeho laserového skenu – cont'd

- v rámci DP navržena *inovativní metoda* rekonstrukce 3D modelů smrkových stromů

- rekonstruované modely využity v návazném výzkumu

- získávání statistických informací o množství dřevité biomasy a o základní struktuře stromů
- parametrizované opatřování zelenou biomasou (mladé větve + jehličky) – součást PhD práce
- importování modelů do nástrojů umožňujících analýzu šíření slunečního záření s využitím DART modelů

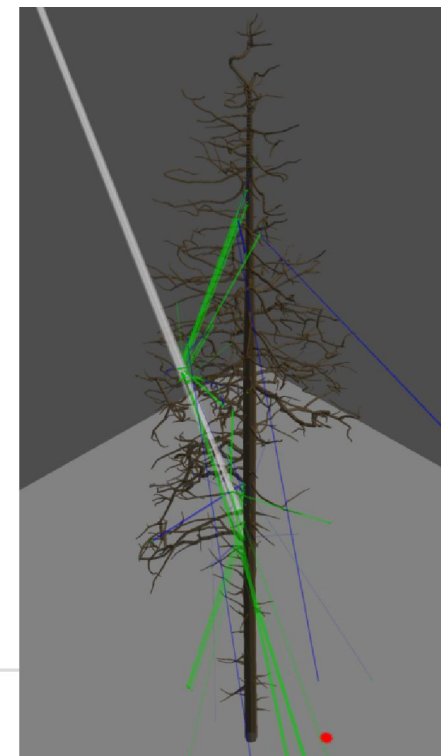
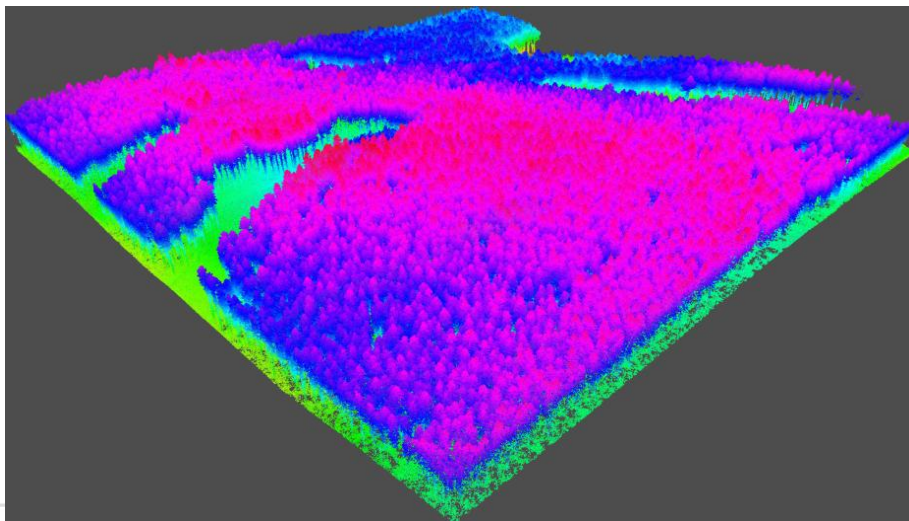




## Příklady spolupráce s partnery II.

### Rekonstrukce lesních porostů z full-wave LiDAR skenů

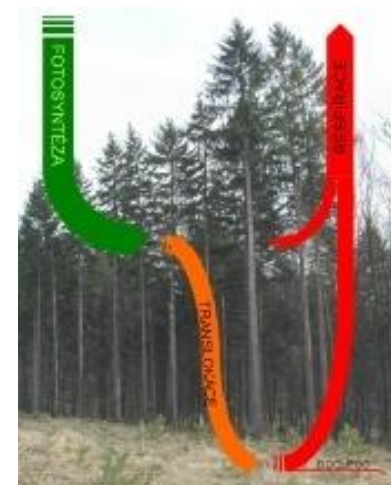
- probíhající téma PhD práce, příprava společného projektu
- **cíl: co nejvěrnější 3D rekonstrukce celých lesních porostů z leteckých full-wave LiDARových skenů**
  - možné využití hyperspektrálních skenů, termálních skenů, in-situ měření, ...



## Příklady spolupráce s partnery III.

### ■ Použití neuronových sítí pro doplňování chybějících dat eddy-kovariančních měření

- partner: *CzechGlobe* (prof. Marek, dr. Pavelka)
- **cíl projektu: nalezení nové, plně automatické metody pro doplňování chybějících měření**
  - formou učení na historických datech
    - *doprovodné charakteristiky* – teplota, tlak, vlhkost, ...
- **hlavní problémy:**
  - nutnost brát v úvahu i historická data
  - les se vyvíjí (roste)



# ■ Příklady spolupráce s partnery IV.

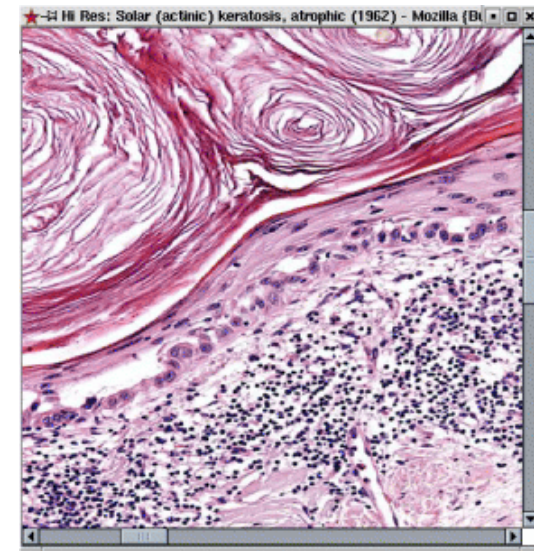
## Identifikace oblastí zasažených geometrickými distorzemi v leteckých skenech krajiny

- partner: *CzechGlobe* (prof. Marek, dr. Hanuš)
- **cíl projektu: nalezení nové, plně automatické metody pro identifikaci oblastí, ve kterých došlo při skenování k náhlému vychýlení letadla**
  - a tím zkreslení skenovaných dat
  - → *analýza obrazu*
  - existující přístupy vhodné spíše pro detekci problémů ve skenech objektů pravidelných tvarů (domy) než pro detekci v rozmanitém porostu
- **hlavní problémy: rozmanitá struktura stromů**

# Příklady spolupráce s partnery V.

## ■ Virtuální mikroskop, patologické atlasy

- partner: *LF MU* (doc. Feit)
- **cíl projektu: implementace virtuálního mikroskopu pro dermatologický atlas (webová aplikace)**
  - zobrazuje skeny tkání
    - rozlišení až 170000x140000 pixelů
    - složeno z dlaždic (až 30000 ks)
  - umožňuje „doostřovat“ jako skutečný mikroskop
- **hlavní problémy:**
  - optimalizace zpracování snímků, autentizace



# ■ Příklady spolupráce s partnery VI.

## Hledání problematických uzavírek v silniční síti ČR

- partner: *Centrum Dopravního Výzkumu v.v.i., Olomouc*  
(dr. Bíl, dr. Vodák)
- **cíl projektu: nalezení metody pro identifikaci problémových uzavírek v silniční síti ČR (aktuálně Zlínského kraje)**
  - Identifikace uzavírek vedoucích (dle definovaných ohodnocovacích funkcí) k problémům v dopravě
- **hlavní problémy: výpočetní náročnost**

## Příklady spolupráce s partnery VII.

- **Biobanka klinických vzorků (BBMRI\_CZ)**
  - *partner: Masarykův onkologický ústav, Recamo*
- **Modely šíření epileptického záchvatu a dalších dějů v mozku**
  - *partner: LF MU, ÚPT AV, CEITEC*
- **Fotometrický archiv astronomických snímků**
- **Extrakce fotometrických údajů o objektech z astronomických snímků**
- **Automatické fitování kontinua echelletovských spekter**
  - *3x partner: Ústav teoretické fyziky a astrofyziky PŘF MU*
- **Bioinformatická analýza dat z hmotnostního spektrometru**
  - *partner: Ústav experimentální biologie PŘF MU*
- **Synchronizace časových značek v leteckých snímcích krajiny**
  - *partner: CzechGlobe*
- ...

## Úložné služby

---

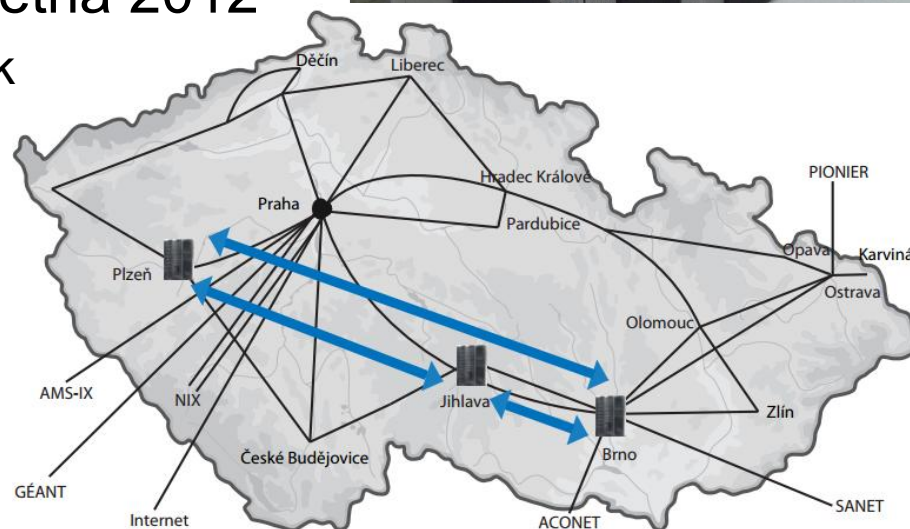
# Dlouhodobé uchování dat – proč?

- potřeba dlouhodobě uchovávat vědecká data vzrůstá
    - uchování primárních dat z experimentů a výpočtů
    - s ohledem na potenciál jejich dalšího využití
    - k návratu a revizi publikovaných výsledků
  - centralizovaná úložná infrastruktura
    - prostředek k dlouhodobému, spolehlivému a ekonomicky výhodnému uchování dat
      - lze zajistit pouze na úrovni binárních dat (nelze zajistit kurátorství dat)
    - umožňuje pořídit zařízení dostatečné velikosti
      - spravované specializovaným týmem
      - s možností sdílení dat mezi skupinami uživatelů
-



# Budovaná infrastruktura datových úložišť

- trojice úložišť: **Plzeň, Jihlava, Brno**
  - plánovaná fyzická kapacita **cca 16+ PB**
  - **duální připojení do páteřní sítě**
- **Plzeň** v pilotním provozu od května 2012
  - cca 500 TB disků + 3300 TB pásek
- **Jihlava a Brno (ÚVT MU)**
  - finišují dodávky/instalace
  - obě založeny na MAID technologii



# Možnosti využití datových úložišť I.

- zálohy
    - uživatelé mají primární data u sebe
    - na úložiště odkládají zálohu pro případ havárie
  - archivace
    - uživatelé na úložiště odkládají cenná primární data
    - uživatelé nemají vlastní prostředky pro dlouhodobé uchování takových dat
  - sdílení dat
    - distribuovaný tým potřebuje společně pracovat nad většími objemy dat, případně je zveřejňovat
  - „něco jiného“
    - v rámci možností lze podpořit i jiné scénáře
-

## Možnosti využití datových úložišť II.

- a naopak: **na co se vzdálené úložiště příliš nehodí**
    - interaktivní práce zejména s větším množstvím malých souborů
    - ukládání dat s potřebou přístupu v reálném čase
      - prioritou je spolehlivost uložení, dostupnost méně
      - „pokud při nedostupnosti dat zemře pacient, pak sem taková data nepatří“
-

# Infrastruktura DÚ „pod pokličkou“ I.

*Aneb „Co je potřeba vědět o specifických těchto úložištích?“*

## Úložiště jsou hierarchická

- vrstvy médií různé kapacity a rychlosti
    - rychlé disky/pomalejší disky/MAID/pásy
    - drahý provoz → levnější provoz
      - optimalizace poměru kapacity, přístupové doby, pořizovací ceny a nákladů na údržbu
  - a automatizovaný systém pro přesuny dat mezi nimi
    - déle nepoužívaná data odkládána do pomalejších vrstev
    - pro uživatele transparentní, resp. téměř transparentní
      - přístup k dlouho nepoužitému souboru trvá déle
-

# Infrastruktura DÚ „pod pokličkou“ II.

## Správa uživatelů

- jednotná správa uživatelských účtů skrze celou VI CESNET
  - federace eduID.cz
  - virtuální organizace:
    - VOs pro skupiny uživatelů
    - **VO Storage** pro jednotlivé uživatele

## Přístupy k úložišti

- *souborové*
    - NFSv4 (známé uživatelům MetaCentra)
    - výhledově CIFS (známý „síťový disk“ z Windows)
    - rsync, scp, FTPS, ...
  - *gridové úložiště v systému dCache*
  - *bloková zařízení*
-

## DÚ – služby dostupné uživatelům

- prostředí pro **zálohování, archivaci, a sdílení dat**
  - **úložiště pro speciální aplikace**
  - **úschovna dat – *FileSender***
    - webová služba pro jednorázový přenos velkých souborů
      - velkých: aktuálně 500 GB
      - <http://filesender.cesnet.cz>
    - alespoň jedna strana komunikace musí být oprávněný uživatel infrastruktury
      - autentizace federací eduID.cz
    - oprávněný uživatel **může nahrát soubor a poslat příjemci oznámení**
    - pokud oprávněný uživatel potřebuje **získat soubor od externího uživatele, pošle mu pozvánku**
-

# FileSender – ukázka I.



The screenshot shows the FileSender website interface. At the top left is the FileSender logo, which includes a yellow truck icon and the text "FILESENDER" with a red chili pepper. To the right, it says "an initiative by" followed by logos for aarnet, UNINETT, HEAnet, and SURF NET. Below these are two buttons: "Pomoc" and "O programu". A status bar indicates: "| UP: 1820 files (2305GB) | DOWN: 2065 files (1876GB) | 1.5-rc1 HTML 5 ✓". The main content area has a heading "Vítejte na FileSender" and a paragraph: "FileSender je bezpečná cesta pro sdílení velkých souborů mezi všemi! Přihlaš se a nahraj své soubory nebo pozvi ostatní, ať soubory nahrají oni." Below this is a "Přihlásit" button with a large grey arrow pointing to the right. At the bottom center is the CESNET logo.

## FileSender – ukázka II.



[O federaci](#) | [Politika](#) | [Kontakt](#) | [Nápověda](#)

### Zvolte svou domovskou organizaci

Přístup ke zdroji na serveru '[filesender.cesnet.cz](#)' vyžaduje autentizaci.

- Uložit tuto volbu do ukončení relace prohlížeče.
- Uložit tuto volbu nastálo.

Operátorem federace [eduID.cz](#) je [CESNET, z.s.p.o.](#)



**CESNET**

### Přihlášení

**Uživatelské jméno**

**Heslo**



## FileSender – ukázka III.



 FILESENDER 

— an initiative by —  
   

[Nahrát nový soubor](#) [Pozvánky](#) [Mé soubory](#) [Pomoc](#) [O programu](#) [Odhlásit](#)

Vítejte Tomáš Košnar | UP: 1820 files (2305GB) | DOWN: 2065 files (1876GB) | 1.5-rc1 **HTML 5** ✓

### Nahrát soubor

**Příjemce:**

**Odesílatel:** tomas.kosnar@cesnet.cz

**Předmět: (volitelné)**

**Zpráva: (volitelné)**

**Datum expirace:**

**Vyberte soubor:**  Soubor nevybrán

**Souhlasím s podmínkami užití této služby.**  
[Zobrazit/Skrýt]



# Služby pro podporu vzdálené spolupráce

---

## Prostředí pro podporu spolupráce

### Profil služeb:

- Podpora interaktivní spolupráce v reálném čase
  - videokonference
  - webkonference
  - speciální přenosy
  - IP telefonie
- Podpora pasivní účasti na akcích
  - streaming a videoarchív
- Spolupráce a konzultace
- Výzkum a vývoj

<http://vidcon.cesnet.cz>

---

## Prostředí pro spolupráci – videokonference I.

- infrastruktura pro přenos **kvalitního obousměrného obrazu** (max. HD), **širokopásmového zvuku** a **pasivních podkladů** (jednosměrné prezentace)
    - přístup prostřednictvím specializovaných HW/SW jednotek (H.323, SIP)
      - koncové stanice si pořizuje instituce
  - technologie vhodná pro:
    - schůzky distribuovaných skupin
    - konzultace se zahraničními pracovišti
    - doktorandské semináře a zkoušky
    - přijímací řízení zahraničních studentů
    - přednášky těžko dosažitelných expertů v posluchárně
    - vzdálená účast přednášejících na konferencích
    - ...
-

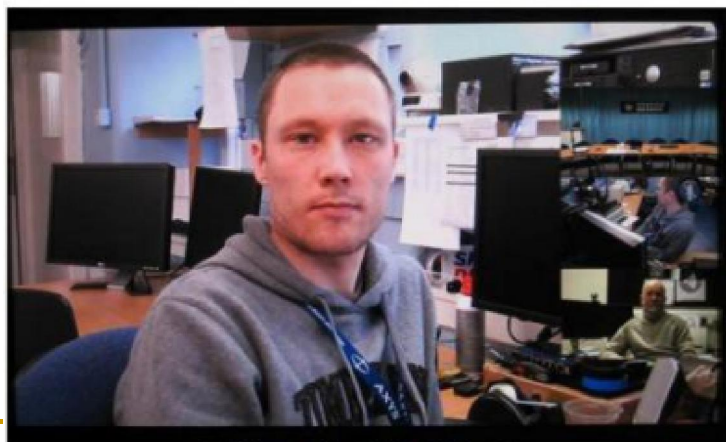
## Prostředí pro spolupráci – videokonference II.

- **poskytované služby:**
    - virtuální místnosti pro vícebodová spojení (MCUs)
      - aktuálně „na požádání“
      - rezervační systém Shongo ve fázi implementace
    - napojení na nahrávání a streaming
    - pomoc s výběrem HW/SW klientů
      - infrastruktura je heterogenní
      - cílem je kompatibilita
    - nabídka sdílených licencí pro SW klienty
-

## Prostředí pro spolupráci – videokonference III.



Four Sites Quad Split



Full Screen Site with Multiple PIPs



Presentation Large with Four Sites video POP images

S počtem účastníků NErostou  
nároky na stanice

## Prostředí pro spolupráci – webkonference I.

- nabízí obousměrný obraz (SD), zvuk a bohaté podklady
    - sdílení souborů a aplikací
    - tabule
    - poznámky
    - hlasování
    - chat
  - technologie vhodná pro:
    - schůzky skupin (ideálně jednotlivců)
    - semináře s větším počtem účastníků
      - ideálně bez potřeby videa od všech účastníků
-

## Prostředí pro spolupráci – webkonference II.

- infrastruktura – **Adobe Connect**
    - virtuální místnosti s perzistentním obsahem
  - klient v prostředí webového prohlížeče (Adobe Flash)
    - bez nutnosti instalace (základ)
    - personální vybavení je shodné se SW videokonferencemi (webkamera, sluchátka, mikrofon)
  - **fyzické místnosti s více účastníky** je možné zapojit, ale je potřeba vhodné vybavení (mikrofony, prvky potlačení zpětné vazby, lepší kamera)
-



# Prostředí pro spolupráci – webkonference III.

The screenshot displays a web conference interface with the following components:

- Meeting Controls:** Meeting, Layouts, Pods, Audio, and Help menus.
- Share Window:** A technical diagram titled "test" showing two Mac Pro towers connected via a 10GbE network. Each Mac Pro is connected to a Kona3 card, which is in turn connected to a BaseLight Four camera and a Sony SXR4K camera via dual-link HD-SDI cables. A yellow circle highlights the left Mac Pro.
- Video Window:** A video feed showing a person in a room, with a "Stop My Webcam" button.
- Attendees:** A list showing 2 attendees: Hosts (1) - Joe Růžička and Presenters (1) - android.
- Files:** A table with columns "Name" and "Size". One file "Tree.jpg" is listed with a size of 752 KB.
- Chat:** A chat window titled "Chat (Everyone)" with the message "The chat history has been cleared" and "Jan Růžička: Klasický chat".
- Notes:** A notes window containing the text "tady jsou poznámky, které je možno poslat mailem".

# Prostředí pro spolupráci – webkonference IV.



S počtem účastníků s videem rostou nároky na stanice

## Prostředí pro spolupráci – streaming

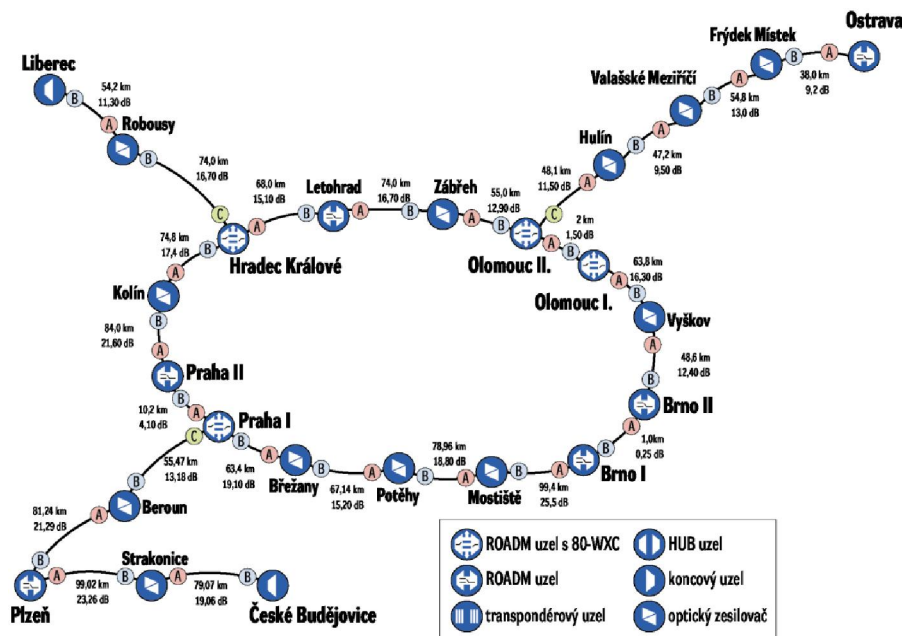
- Infrastruktura
    - farma streamovacích serverů (Windows Media, Wowza – H.264, Flash)
    - schopnost obsloužit tisíce odběratelů
    - propojení s videokonferenční infrastrukturou i speciálními přenosy
  - Snímání a kódování si primárně zajišťuje uživatel
  - Možná pomoc se zajištěním vysílání z akcí
    - nutno domluvit s rozumným předstihem
-

## Další podpůrné služby

---

## Komunikační infrastruktura

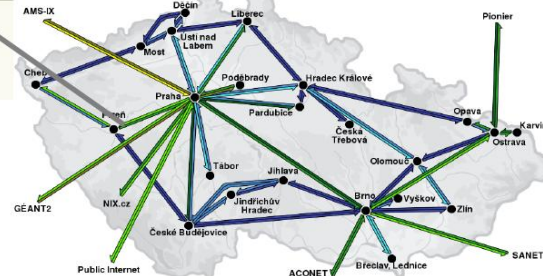
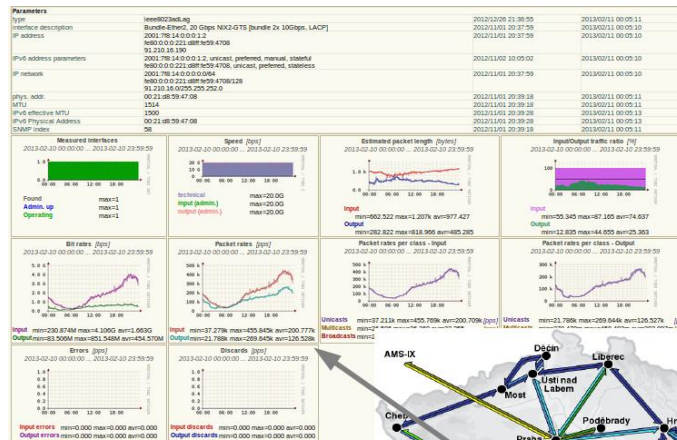
- Základní komponenta e-infrastruktury: **vysokorychlostní počítačová síť CESNET2**
  - **spolehlivost sítě** zajištěna duálním připojením uzlů
  - **výkon sítě:**
    - jádro sítě 100 Gbps
    - uzly do jádra připojeny 40-100 Gbps
  - přímé propojení (na fyzické vrstvě do **pan-evropské sítě pro výzkum a vzdělávání GÉANT**)



# Monitoring komunikační infrastruktury

## Sledování provozu sítě

- sběr, zpracování, zpřístupnění, vizualizace informací o infrastruktuře a o IP provozu
- automatická detekce a notifikace jevů, anomálií apod.
- monitorování kvalitativních charakteristik sítě



## Bezpečnost

### Řešení bezpečnostních incidentů

- platforma (technická, organizační) pro **řešení a asistenci při řešení bezpečnostních incidentů** v e-infrastruktuře CESNET a administrativní doméně komunity
  - cesnet.cz, cesnet2.cz, ces.net, liberrouter.org, liberrouter.net, ipv6.cz, acad.cz, eduroam.cz a v IP adresách interní infrastruktury sítě CESNET2
- bezpečnostní tým CESNET-CERTS
- *další služby:*
  - **školení pro (nejen) studenty prvních ročníků**
  - další osvětová činnost
    - školení, semináře, workshopy, ...



<http://csirt.cesnet.cz>

---

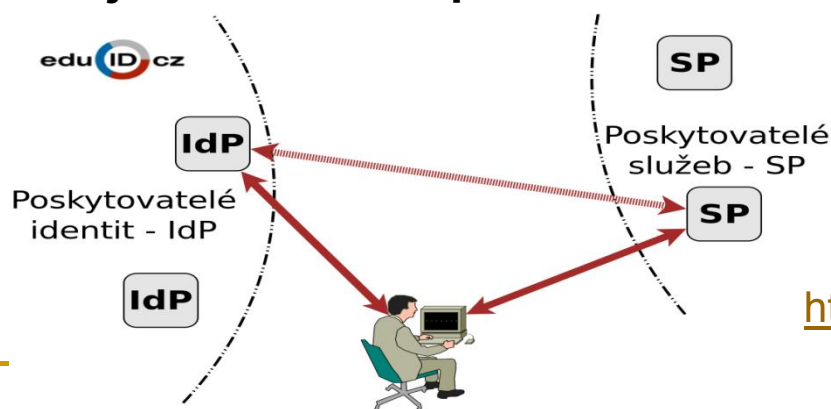
## Federalizovaná správa identit

### Česká akademická federace identit eduID.cz



- autentizační infrastruktura pro vzájemné využívání identit uživatelů při řízení přístupu k síťovým službám
  - uživatel využívá **pouze jedno heslo pro přístup k více aplikacím**
  - **správci aplikací neudrží autentizační data uživatelů**, ani neprovádí autentizaci
  - autentizace uživatele probíhá **vždy v kontextu domovské organizace**, **citlivé autentizační údaje** uživatele **neopouští domovskou síť**

- **Hostel IdP** pro uživatele z institucí nezapojených do eduID.cz
  - např. AV ČR



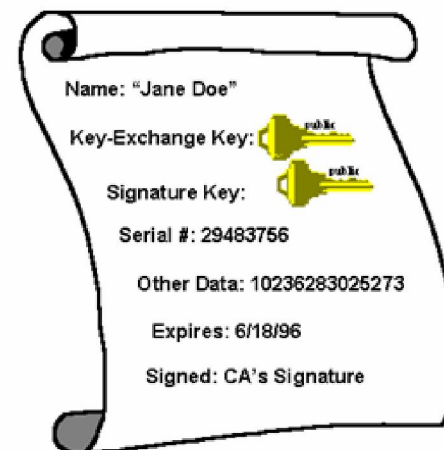
<http://www.eduid.cz>



# Certifikáty pro uživatele a servery (PKI)

## Certifikační autorita CESNET CA

- vydávání certifikátů od TERENA (*Trans-European Research and Education Networking Association*)
- *služby CESNET CA:*
  - vydávání osobních certifikátů
  - vydávání certifikátů pro servery a služby
  - certifikace registračních úřadů
  - certifikace certifikačních úřadů



## Podpora IP mobility a roamingu

### Eduroam.cz

- snaha umožnit uživatelům transparentní používání sítí (českých i zahraničních) zapojených do projektu Eduroam
- *služby CESNET Eduroam:*
  - koordinace a propagace souvisejících aktivit
  - začleňování nových organizací
  - provoz infrastruktury RADIUS serverů



## Další služby VI CESNET

- Konzultace a školení
  - bezpečnostní školení
  - technické konzultace
  - Cisco akademie
- Pokročilé síťové služby
  - fotonické a lambda služby
  - časové služby v síti
- Prostředí pro vývoj a testování aplikací/protokolů (PlanetLab)
- Transfer technologií
  - návrh optických sítí a systémů „na míru“
  - poskytování licencí k vyvinutým zařízením
- Interní služby
  - systém správy účtů uživatelů infrastruktur VI CESNET a CERIT-SC (Perun)
- ...

Více viz

<http://www.cesnet.cz/sluzby>

## Závěr

- **VI CESNET:**
    - **výpočetní služby (MetaCentrum NGI & MetaVO)**
    - *úložné služby (archivace, zálohování, výměna dat, ...)*
    - *služby pro podporu vzdálené spolupráce (videokonference, webkonference, streaming, ...)*
    - další podpůrné služby (...)
  - **Centrum CERIT-SC:**
    - *výpočetní služby (produkční i flexibilní infrastruktura)*
    - *služby pro podporu kolaborativního výzkumu*
    - správa identit uživatelů jednotná s VI CESNET
  - **Hlavní sdělení prezentace: „Pokud v poskytovaných službách nenalézáte řešení Vašich konkrétních potřeb, **ozvěte se** – společnými silami se pokusíme řešení nalézt...“**
-

# Děkuji Vám za pozornost!

`rebok@ics.muni.cz`



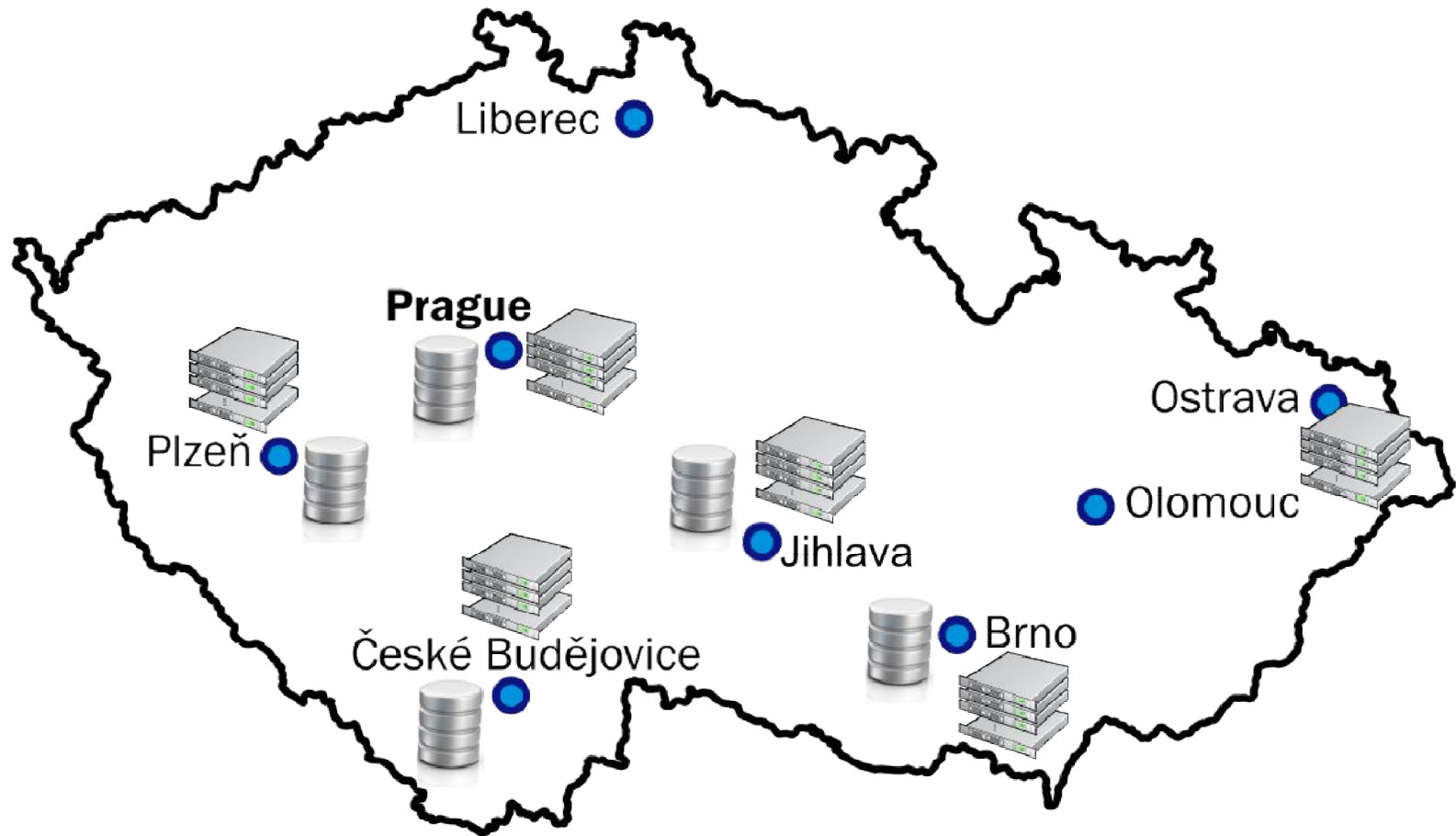
# Hands-on seminar

---

# Overview

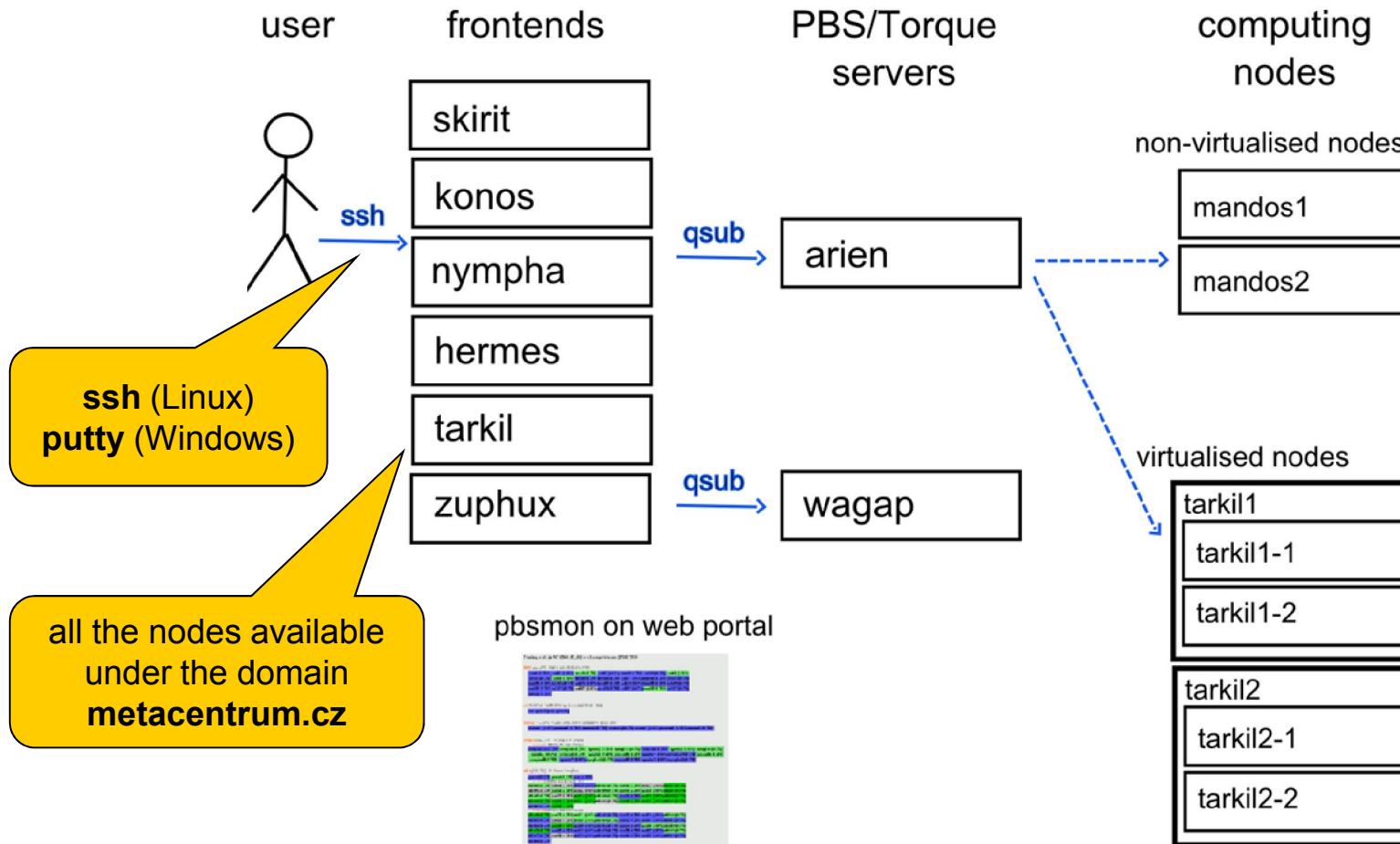
- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
- **Grid infrastructure overview**
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
- Real-world examples

# Grid infrastructure overview I.

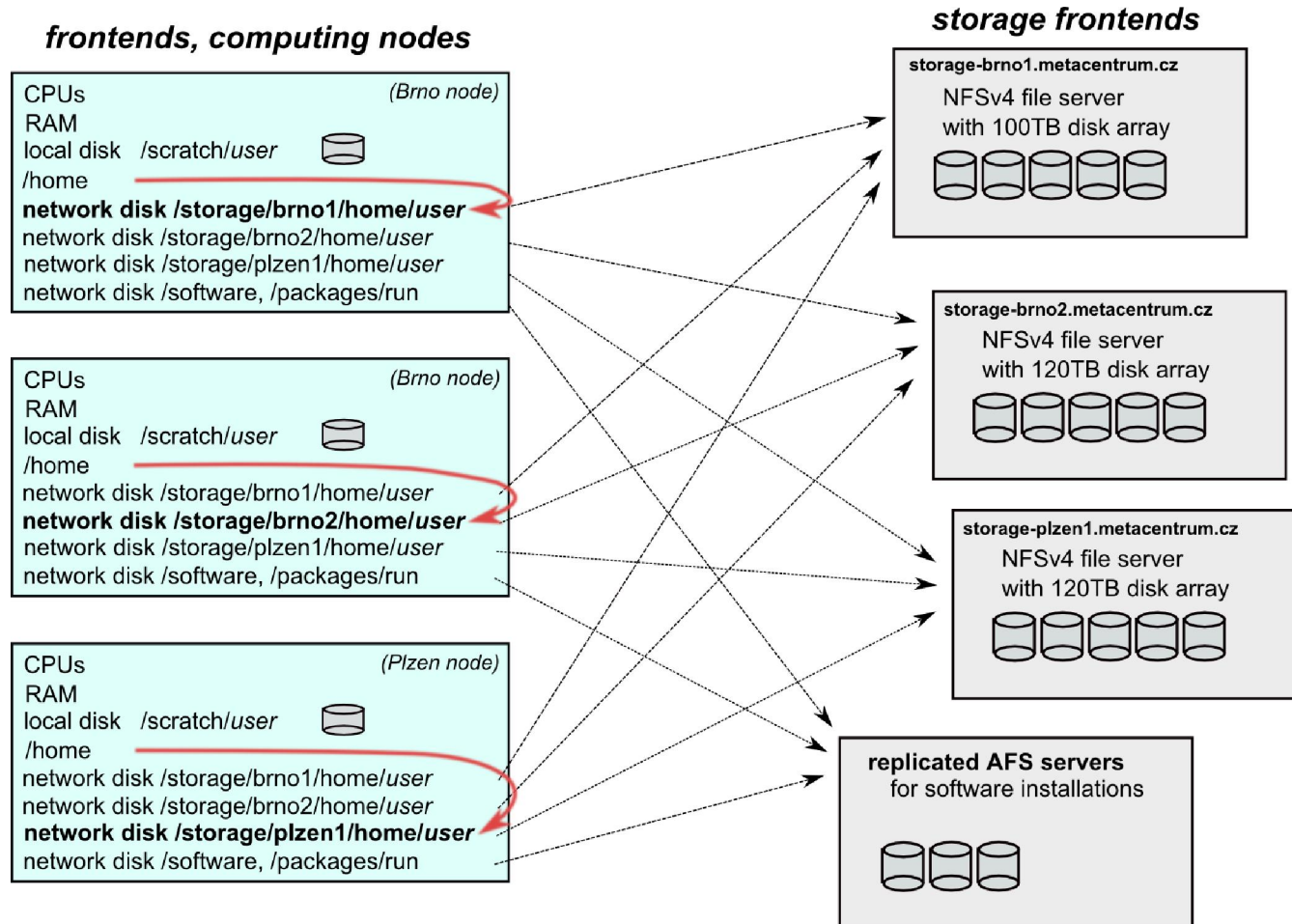




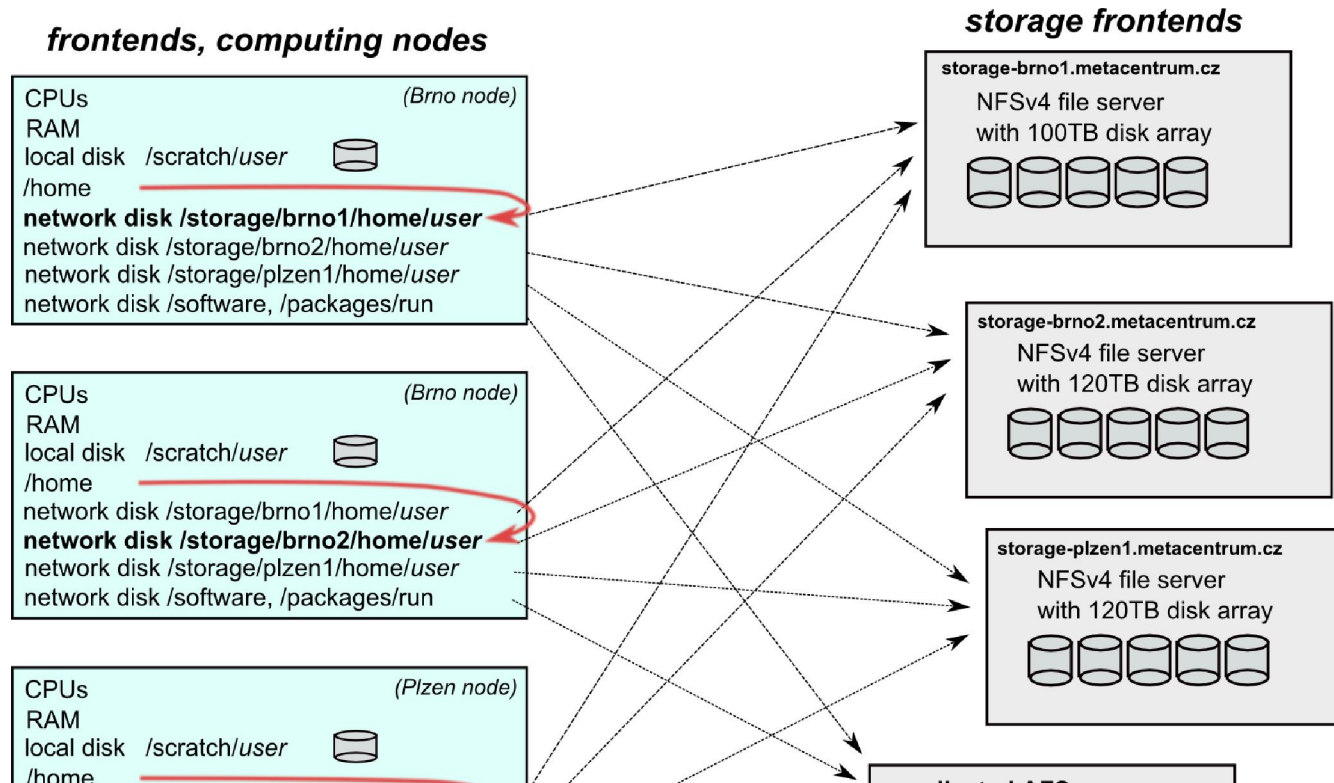
# Grid infrastructure overview II.



# Grid infrastructure overview II.



# Grid infrastructure overview II.



## Current improvement:

- the /storage/XXX/home/\$USER as default login directory

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- **How to ... specify requested resources**
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples

# How to ... specify requested resources I.

- before running a job, one needs to have an idea **what resources** the job requires
  - and how many of them
- means for example:
  - number of **nodes**
  - number of **cores per node**
  - an **upper estimation** of job's **runtime**
  - amount of **free memory**
  - amount of **scratch space** for temporal data
  - number of requested **software licenses**
  - etc.
- the resource requirements are then **provided to the qsub utility** (when submitting a job)
  
- **details about resources' specification:**  
[http://meta.cesnet.cz/wiki/Plánovací\\_systém\\_-\\_detailní\\_popis#Specifikace\\_požadavků\\_na\\_výpočetní\\_zdroje](http://meta.cesnet.cz/wiki/Plánovací_systém_-_detailní_popis#Specifikace_požadavků_na_výpočetní_zdroje)

# How to ... specify requested resources II.

## Graphical way:

- *qsub assembler*: <http://metavo.metacentrum.cz/cs/state/personal>
- allows to:
  - graphically specify the requested resources
  - check, whether such resources are available
  - generate command line options for *qsub*
  - check the usage of MetaVO resources

## Textual way:

- **more powerful** and (once being experienced user) **more convenient**
- see the following slides/examples →

# How to ... specify requested resources II.

## Node(s) specification:

- *general format:* `-l nodes=...`

### Examples:

- 2 nodes:
  - `-l nodes=2`
- 5 nodes:
  - `-l nodes=5`
- by default, allocates just a single core on each node
  - → should be used together with **processors per node (PPN)** specification
- if “`-l nodes=...`” is not provided, just a single node with a single core is allocated

# How to ... specify requested resources IV.

## Processors per node (PPN) specification:

- *general format:* `-1 nodes=...:ppn=...`
- 2 nodes, both of them having 3 processors:
  - `-1 nodes=2:ppn=3`
- 5 nodes, each of them with 2 processors:
  - `-1 nodes=5:ppn=2`

## *More complex specifications are also supported:*

- 3 nodes: one of them with just a single processor, the other two with four processors per node:
  - `-1 nodes=1:ppn=1+2:ppn=4`
- 4 nodes: one with a single processor, one with two processors, and two with four processors:
  - `-1 nodes=1:ppn=1+1:ppn=2+2:ppn=4`



# How to ... specify requested resources V.

## Other useful nodespec features:

- nodes just from a **single (specified) cluster** (suitable e.g. for MPI jobs):
  - *general format:* `-l nodes=...:cl_<cluster_name>`
  - e.g., `-l nodes=3:ppn=1:cl_skirit`
- nodes with a **(specified) computing power** (based on SPEC benchmark):
  - *general format:* `-l nodes=...:minspec=XXX OR -l nodes=...:maxspec=XXX`
  - e.g., `-l nodes=3:ppn=1:minspec=10:maxspec=20`
- nodes located in a **specific location** (suitable when accessing storage in the location)
  - *general format:* `-l nodes=...:<brno|plzen|...>`
  - e.g., `-l nodes=1:ppn=4:brno`
- **exclusive node assignment:**
  - *general format:* `-l nodes=...#excl`
  - e.g., `-l nodes=1#excl`
- **negative specification:**
  - *general format:* `-l nodes=...:^<feature>`
  - e.g., `-l nodes=1:ppn=4:^cl_manwe`
- ...

A list of nodes' features can be found here: <http://metavo.metacentrum.cz/pbsmon2/props>

# How to ... specify requested resources VI.

## Specifying memory resources (default = 400mb):

- *general format:* `-l mem=...<suffix>`
  - e.g., `-l mem=300mb`
  - e.g., `-l mem=2gb`

## Specifying job's maximum runtime (default = 24 hours):

- it is necessary to specify an upper limit on job's runtime:
- *general format:* `-l walltime=[Xw] [Xd] [Xh] [Xm] [Xs]`
  - e.g., `-l walltime=13d`
  - e.g., `-l walltime=2h30m`
- previous specifications via queues (`short/normal/long`) still possible, however **not recommended**

# How to ... specify requested resources VII.

## Specifying requested scratch space:

- useful, when the application performs **I/O intensive operations** OR for **long-term computations** (reduces the impact of network failures)
  - the scratches are **local to the nodes** (smaller) and/or
  - **shared for the nodes** of a specific cluster over Infiniband (bigger) -- currently “doom”, “hildor”, “mandos” and “ramdal” clusters only
    - thus being as fast as possible
- **scratch space:** `-l scratch=...<suffix>`
  - e.g., `-l scratch=500mb`
- there is a **private scratch directory for particular job**
  - `/scratch/$USER/job_$PBS_JOBID` directory for job's scratch
- there is a **SCRATCHDIR environment variable** available in the system
  - points to the assigned scratch space/location

# How to ... specify requested resources VII.

## Specifying requested scratch space:

- useful, when the application performs **I/O intensive operations** OR for **long-term computations** (reduces the impact of network failures)
  - the scratches are **local to the nodes** (smaller) and/or
  - **shared for the nodes** of a specific cluster over Infiniband (bigger) -- currently “doom”, “hildor”, “mandos” and “ramdal” clusters only

## Current improvements:

- **SCRATCH:**
  - additional property to indicate a specific scratch type requested
    - `-l scratch_type=[local|shared|ssd] [:first]`
- **Planned features:**
  - reservations/quotas on the scratches

# How to ... specify requested resources VIII.

## Specifying requested software licenses:

- necessary when an application requires a SW licence
  - the job becomes started once the requested licences are available
  - the information about a licence necessity is **provided within the application description** (see later)
- **general format:** `-l <lic_name>=<amount>`
  - e.g., `-l matlab=2`
  - e.g., `-l gridmath8=20`

...

## (advanced) Dependencies on another jobs

- allows to create a workflow
  - e.g., to start a job once another one successfully finishes, breaks, etc.
- see qsub's “-w” option (`man qsub`)

# How to ... specify requested resources VIII.

## Specifying requested software licenses:

- necessary when an application requires a SW licence
  - the job becomes started once the requested licences are available
  - the information about a licence necessity is **provided within the application description** (see later)
- **general format:** `-l <lic_name>=<amount>`
  - e.g., `-l matlab=2`
  - e.g., `-l gridmath8=20`

...

(a

## More information available at:

- [https://wiki.metacentrum.cz/wiki/Spouštění\\_úloh\\_v\\_plánovači#Stru.C4.8Dn.C3.A9\\_shrnut.C3.AD\\_pl.C3.A1nov.C3.A1n.C3.AD\\_.C3.BAloh](https://wiki.metacentrum.cz/wiki/Spouštění_úloh_v_plánovači#Stru.C4.8Dn.C3.A9_shrnut.C3.AD_pl.C3.A1nov.C3.A1n.C3.AD_.C3.BAloh)

# How to ... specify requested resources IX.

## Questions and Answers:

- *Why is it necessary to specify the resources in a proper number/amount?*
  - because when a job consumes more resources than announced, it will be **killed** by us (you'll be informed)
    - otherwise it may influence other processes running on the node
- *Why is it necessary not to ask for excessive number/amount of resources?*
  - the jobs having smaller resource requirements are started (i.e., get the time slot) **faster**
- *Any other questions?*



# How to ... specify requested resources X.

## Examples:

- *Ask for a single node with 4 CPUs, 1gb of memory.*
  - `qsub -l nodes=1:ppn=4 -l mem=1gb`
- *Ask for a single node (1 CPU) – the job will run approx. 3 days and will consume up to 10gb of memory.*
  - ???
- *Ask for 2 nodes (1 CPU per node) not being located in Brno.*
  - ???
- *Ask for two nodes – a single one with 1 CPU, the other two having 5 CPUs and being from the manwe cluster.*
  - ???
- ...





# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- **How to ... run an interactive job**
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples

# How to ... run an interactive job I.

## Interactive jobs:

- result in getting a prompt on a single (**master**) node
  - one may perform interactive computations
  - the other nodes, if requested, remain allocated and accessible (see later)
  
- How to **ask for an interactive job**?
  - add the option “-I” to the qsub command
  - e.g., `qsub -I -l nodes=1:ppn=4:cl_mandos`
  
- **Example** (valid for this demo session):
  - `qsub -I -q MetaSeminar -l nodes=1`

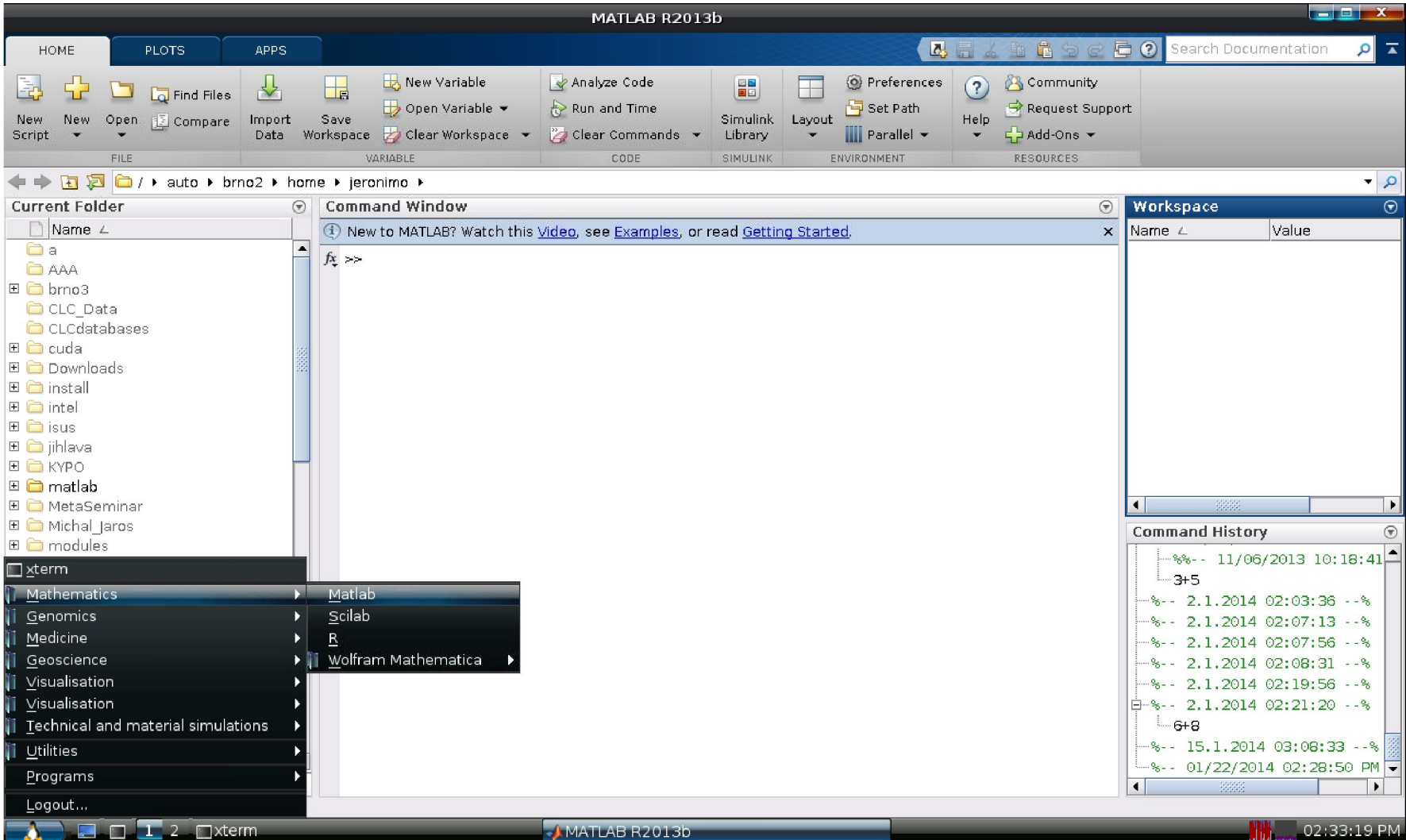
# How to ... run an interactive job II.

**Textual mode:** simple

**Graphical mode:**

- *(preferred)* **remote desktops based on VNC servers (pilot run):**
- available from frontends as well as computing nodes (interactive jobs)
  - `module add gui`
  - `gui start [-s] [-w] [-g GEOMETRY] [-c COLORS]`
    - uses one-time passwords
    - allows to access the VNC via a supported **TigerVNC client** or **WWW browser**
    - **allows SSH tunnels** to be able to connect with a wide-range of clients
    - allows to specify several parameters (e.g., desktop resolution, color depth)
    - `gui info [-p] ...` displays active sessions
    - `gui stop [sessionID] ...` allows to stop/kill an active session
- **see more info at**  
[https://wiki.metacentrum.cz/wiki/Vzdálený\\_desktop](https://wiki.metacentrum.cz/wiki/Vzdálený_desktop)

# How to ... run an interactive job II.



The image shows the MATLAB R2013b software interface. The top menu bar includes HOME, PLOTS, and APPS. The ribbon contains various toolbars for file operations, workspace management, code execution, and environment settings. The current folder is set to `auto > brno2 > home > jeronimo`.

The Command Window is active, displaying the prompt `>>`. A context menu is open over the Command Window, listing various applications:

- Mathematics
  - Matlab
- Genomics
  - Scilab
- Medicine
  - R
- Geoscience
  - Wolfram Mathematica
- Visualisation
- Visualisation
- Technical and material simulations
- Utilities
- Programs
- Logout...

The Workspace window is empty. The Command History window shows a list of previous commands and their execution times, including `3+5` and `6+8`.

The taskbar at the bottom shows the MATLAB R2013b application and a terminal window titled `xterm`. The system clock indicates the time is 02:33:19 PM.

# How to ... run an interactive job II.

## Graphical mode (further options):

- *(fallback)* **tunnelling a display through ssh** (Windows/Linux):
  - connect to the frontend node having SSH forwarding/tunneling enabled:
    - Linux: `ssh -X skirit.metacentrum.cz`
    - Windows:
      - install an XServer (e.g., Xming)
      - set Putty appropriately to enable X11 forwarding when connecting to the frontend node
        - Connection → SSH → X11 → Enable X11 forwarding
  - ask for an interactive job, **adding “-x” option** to the `qsub` command
    - e.g., `qsub -I -x -l nodes=... ..`
- *(tech. gurus)* **exporting a display** from the master node to a Linux box:
  - `export DISPLAY=mycomputer.mydomain.cz:0.0`
  - on a Linux box, run `xhost +` to allow all the remote clients to connect
    - be sure that your display manager allows remote connections

# How to ... run an interactive job III.

## Questions and Answers:

- *How to **get an information** about the **other nodes allocated** (if requested)?*
  - `master_node$ cat $PBS_NODEFILE`
  - works for batch jobs as well
- *How to **use the other nodes allocated**? (holds for batch jobs as well)*
  - MPI jobs use them automatically
  - otherwise, use the **pbsdsh** utility (see "`man pbsdsh`" for details) to run a remote command
  - if the pbsdsh does not work for you, use the **ssh** to run the remote command
- *Any other questions?*



# How to ... run an interactive job III.

## Questions and Answers:

- How to **get an information** about the **other nodes allocated** (if

### Hint:

- there are several useful environment variables one may use
- - `$ set | egrep "PBS|TORQUE"`
- e.g.:
  - PBS\_JOBID ... job's identifier
  - PBS\_NUM\_NODES, PBS\_NUM\_PPN ... allocated number of nodes/processors
  - PBS\_O\_WORKDIR ... submit directory (alert: /home path!)
  - ...

un a



# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- **How to ... use application modules**
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples



# How to ... use application modules I.

## Application modules:

- the **modullar subsystem** provides a user interface to modifications of user environment, which are necessary for running the requested applications
- allows to “add” an application to a user environment
  
- **getting a list** of available application modules:
  - `$ module avail`
  - <http://meta.cesnet.cz/wiki/Kategorie:Aplikace>
    - provides the documentation about modules' usage
    - besides others, includes:
      - information whether it is necessary to ask the scheduler for an available licence
      - information whether it is necessary to express consent with their licence agreement

# How to ... use application modules II.

## Application modules:

- **loading** an application into the environment:
  - `$ module add <modulename>`
  - e.g., `module add maple`
- **listing** the already loaded modules:
  - `$ module list`
- **unloading** an application from the environment:
  - `$ module del <modulename>`
  - e.g., `module del openmpi`
- **Note:** *An application may require to express consent with its licence agreement before it may be used (see the application's description). To provide the agreement, visit the following webpage: <http://metavo.metacentrum.cz/cs/myaccount/eula>*
- for more information about application modules, see [http://meta.cesnet.cz/wiki/Aplikační\\_moduly](http://meta.cesnet.cz/wiki/Aplikační_moduly)

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- **How to ... run a batch job**
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples

# How to ... run a batch job I.

## Batch jobs:

- perform the computation as described in their **startup script**
  - the submission results in getting a **job identifier**, which further serves for getting more information about the job (see later)
- How to **submit a batch job**?
  - add the reference to the startup script to the qsub command
  - e.g., `qsub -l nodes=3:ppn=4:cl_mandos <myscript.sh>`
- **Example** (valid for this demo session):
  - `qsub -q MetaSeminar -l nodes=1 myscript.sh`
  - results in getting something like `"12345.arien.ics.muni.cz"`

## How to run a batch job I

### Hint:

- B
- create the file `myscript.sh` with the following content:
    - `$ vim myscript.sh`
  - - `#!/bin/bash`
    - `# my first batch job`
    - `uname -a`
  - see the standard output file (`myscript.sh.o<JOBID>`)
    - `$ cat myscript.sh.o<JOBID>`
  - **Example** (valid for this demo session):
    - `qsub -q MetaSeminar -l nodes=1 myscript.sh`
    - results in getting something like `"12345.arien.ics.muni.cz"`

for

# How to ... run a batch job II.

## Startup script preparation/skelet: (non IO-intensive computations)

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

```
DATADIR="/storage/brno2/home/$USER/" # shared via NFSv4
```

```
cd $DATADIR
```

```
# ... initialize & load modules, perform the computation ...
```

- **further details** – see [http://meta.cesnet.cz/wiki/Plánovací\\_systém\\_-\\_detailní\\_popis#Příklady\\_použití](http://meta.cesnet.cz/wiki/Plánovací_systém_-_detailní_popis#Příklady_použití)

# How to ... run a batch job III.

## Startup script preparation/skelet: (IO-intensive computations or long-term jobs)

```
#!/bin/bash

# set a handler to clean the SCRATCHDIR once finished
trap "rm -r $SCRATCHDIR" TERM EXIT

# set the location of input/output data
# DATADIR="/storage/brno2/home/$USER/"
DATADIR="$PBS_O_WORKDIR"

# prepare the input data
cp $DATADIR/input.txt $SCRATCHDIR || exit 1

# go to the working directory and perform the computation
cd $SCRATCHDIR

# ... initialize & load modules, perform the computation ...

# copy out the output data
# if the copying fails, let the data in SCRATCHDIR and inform the user
cp $SCRATCHDIR/output.txt $DATADIR || { trap - TERM EXIT && echo "Copy output
data failed. Copy them manually from `hostname`" >&2 ; exit 1 ;}
```

# How to ... run a batch job IV.

## Using the application modules within the batch script:

- to use the **application modules** from a **batch script**, add the following line into the script (before loading the module):

- if you use different shell, change the shell identifier (bash → sh | tcsh | ksh | csh | ...)

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

```
...
```

```
module add maple
```

## Getting the job's standard output and standard error output:

- once finished, there appear **two files** in the directory, which the job has been started from:

- `<job_name>.o<jobID>` ... standard output

- `<job_name>.e<jobID>` ... standard error output

- the `<job_name>` can be modified via the "-N" qsub option



# How to ... run a batch job V.

## Job attributes specification:

in the case of batch jobs, the requested resources and further job information (*job attributes* in short) may be specified either on the command line (see "man qsub") or directly within the script:

- by adding the "#PBS" directives (see "man qsub"):

```
#PBS -N Job_name
#PBS -l nodes=2:ppn=1
#PBS -l mem=320kb
#PBS -m abe
#
< ... commands ... >
```

- the submission may be then simply performed by:

```
❑ $ qsub myscript.sh
```

# How to ... run a batch job VI. (complex example)

```
#!/bin/bash
#PBS -l nodes=1:ppn=2
#PBS -l mem=500mb
#PBS -m abe

# set a handler to clean the SCRATCHDIR once finished
trap "rm -r $SCRATCHDIR" TERM EXIT

# set the location of input/output data
DATADIR="$PBS_O_WORKDIR"

# prepare the input data
cp $DATADIR/input.mpl $SCRATCHDIR || exit 1

# go to the working directory and perform the computation
cd $SCRATCHDIR

# initialize the module subsystem and load the appropriate module
. /packages/run/modules-2.0/init/bash
module add maple

# run the computation
maple input.mpl

# copy out the output data (if it fails, let the data in SCRATCHDIR and inform the user)
cp $SCRATCHDIR/output.gif $DATADIR || { trap - TERM EXIT && echo "Copy output data failed.
Copy them manually from `hostname`" >&2 ; exit 1 ;}
```

# How to ... run a batch job VII.

## Questions and Answers:

- *Should you prefer batch or interactive jobs?*
  - definitely the **batch ones** – they use the computing resources **more effectively**
  - use the interactive ones just for testing your startup script, GUI apps, or data preparation

- *Any other questions?*



# How to ... run a batch job VIII.

## Example:

- Create and submit a batch script, which performs a simple Maple computation, described in a file:

```
plotsetup(gif, plotoutput=`myplot.gif`,  
          plotoptions=`height=1024,width=768`);  
plot3d( x*y, x=-1..1, y=-1..1, axes = BOXED, style =  
        PATCH);
```

- process the file using Maple (from a batch script):
  - hint: `$ maple <filename>`

# How to ... run a batch job VIII.

## Example:

- Create and submit a batch script, which performs a simple Maple computation, described in a file:

```
plotsetup(gif, plotoutput=`myplot.gif`,  
          plotoptions=`height=1024,width=768`);  
plot3d( x*y, x=-1..1, y=-1..1, axes = BOXED, style =  
        PATCH);
```

- process the file using Maple (from a batch script):
  - hint: `$ maple <filename>`

## Hint:

- see the solution at  
`/storage/brno2/home/jeronimo/MetaSeminar/20140206-ZCU/Maple`

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- **How to ... determine a job state**
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples

# How to ... determine a job state I.

## Job identifiers

- every job (no matter whether interactive or batch) is **uniquely identified** by its identifier (JOBID)
  - e.g., `12345.arien.ics.muni.cz`
- to obtain any information about a job, the **knowledge of its identifier is necessary**
  - how to list all the recent jobs?
    - graphical way – PBSMON: <http://metavo.metacentrum.cz/pbsmon2/jobs/allJobs>
    - `frontend$ qstat` (run on any frontend)
  - how to list all the recent jobs of a specific user?
    - graphical way – PBSMON: <https://metavo.metacentrum.cz/pbsmon2/jobs/my>
    - `frontend$ qstat -u <username>` (again, any frontend)

# How to ... determine a job state II.

## How to determine a job state?

- graphical way – see PBSMON
  - list all your jobs and click on the particular job's identifier
  - <http://metavo.metacentrum.cz/pbsmon2/jobs/my>
- textual way – `qstat` command (see `man qstat`)
  - brief information about a job: `$ qstat JOBID`
    - informs about: job's state (*Q=queued*, *R=running*, *E=exiting*, *C=completed*, ...), job's runtime, ...
  - complex information about a job: `$ qstat -f JOBID`
    - shows all the available information about a job
    - useful properties:
      - `exec_host` -- the nodes, where the job did really run
      - `resources_used`, `start/completion time`, `exit status`, ...



# How to ... determine a job state III.

## Hell, when my jobs will really start?

- nobody can tell you ☺
  - the **God/scheduler decides** (based on the other job's finish)
  - we're working on an estimation method to inform you about its probable startup
  
- check the **queues' fulfilment**:  
<http://metavo.metacentrum.cz/cs/state/jobsQueued>
  - the higher fairshare (queue's AND job's) is, the earlier the job will be started
- **stay informed** about job's startup / finish / abort (via email)
  - by default, just an information about job's abortation is sent
  - → when submitting a job, add “-m abe” option to the `qsub` command to be informed about all the job's states
    - or “#PBS -m abe” directive to the startup script

# How to ... determine a job state IV.

## Monitoring running job's stdout, stderr, working/temporal files

1. via ssh, log in directly to the execution node(s)
  - how to get the job's execution node(s)?
  - to examine the working/temporal files, navigate directly to them
    - logging to the execution node(s) is necessary -- even though the files are on a shared storage, their content propagation takes some time
  - to examine the stdout/stderr of a running job:
    - navigate to the `/var/spool/torque/spool/` directory and examine the files:
      - `$PBS_JOBID.OU` for standard output (stdout – e.g., “1234.arien.ics.muni.cz.OU”)
      - `$PBS_JOBID.ER` for standard error output (stderr – e.g., “1234.arien.ics.muni.cz.ER”)

## Job's forcible termination

- `$ qdel JOBID` (the job may be terminated in any previous state)
- during termination, the job turns to *E (exiting)* and finally to *C (completed)* state

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- **How to ... run a parallel/distributed computation**
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- Real-world examples

# How to ... run a parallel/distributed computation I.

## Parallel jobs (OpenMP):

- if your application is able to use multiple threads via a shared memory, **ask for a single node with multiple processors**

```
$ qsub -l nodes=1:ppn=...
```

- **make sure**, that before running your application, the **OMP\_NUM\_THREADS** environment variable is **appropriately set**
  - otherwise, your application will use all the cores available on the node
    - → and influence other jobs...
  - usually, setting it to **PPN** is OK

```
$ export OMP_NUM_THREADS=$PBS_NUM_PPN
```

# How to ... run a parallel/distributed computation II.

## Distributed jobs (MPI):

- if your application consists of multiple processes communicating via a message passing interface, **ask for a set of nodes** (with arbitrary number of processors)

```
$ qsub -l nodes=...:ppn=...
```

- **make sure**, that before running your application, the appropriate **openmpi/mpich2/mpich3/lam** module is loaded into the environment

```
$ module add openmpi
```

- then, you can use the `mpirun/mpiexec` routines

```
$ mpirun myMPIapp
```

- it's **not necessary** to provide these routines neither with the number of nodes to use ("`-np`" option) nor with the nodes itself ("`--hostfile`" option)
  - the computing nodes are **automatically detected** by the openmpi/mpich/lam

# How to ... run a parallel/distributed computation III.

## Distributed jobs (MPI): accelerating their speed I.

- to accelerate the speed of MPI computations, ask just for the nodes interconnected by a **low-latency Infiniband interconnection**
  - all the nodes of a cluster are interconnected by Infiniband
  - there are several clusters having an Infiniband interconnection
    - mandos, minos, hildor, skirit, tarkil, nympha, gram, luna, manwe (MetaCentrum)
    - zewura, zegox, zigur, zapat (CERIT-SC)
  
- *submission example:*

```
$ qsub -l nodes=4:ppn=2:infiniband:cl_mandos myMPIscript.sh
```
  
- *starting an MPI computation using an Infiniband interconnection:*
  - in a common way: `$ mpirun myMPIapp`
    - the Infiniband will be automatically detected

# How to ... run a parallel/distributed computation III.

## Distributed jobs (MPI): accelerating their speed I.

- to accelerate the speed of MPI computations, ask just for the nodes interconnected by a **low-latency Infiniband interconnection**
  - all the nodes of a cluster are interconnected by Infiniband
  - there are several clusters having an Infiniband interconnection
    - mandos, minos, hildor, skirit, tarkil, nympha, gram, luna, manwe (MetaCentrum)
    - zewura, zegox, zigur, zapat (CERIT-SC)

### ■ *submission example:*

```
$ qsub -l nodes=4:ppn=2:infiniband:cl_mandos myMPIscript.sh
```

## Planned improvements:

- an intelligent “infiniband” attribute
  - just the nodes interconnected with a shared IB switch will be chosen

# How to ... run a parallel/distributed computation IV.

## Distributed jobs (MPI): accelerating their speed II.

- to test the functionality of an Infiniband interconnection:
  - create a simple program `hello.c` as described here:  
<http://www.slac.stanford.edu/comp/unix/farm/mpi.html>
  - compile with `mpicc`

```
$ module add openmpi
$ mpicc hello.c -o hello
```
  - run the binary (within a job) with the following command:

```
$ mpirun --mca btl ^tcp hello
```



# How to ... run a parallel/distributed computation IV.

## Distributed jobs (MPI): accelerating their speed II.

- to test the functionality of an Infiniband interconnection:
  - create a simple program `hello.c` as described here:  
<http://www.slac.stanford.edu/comp/unix/farm/mpi.html>
  - compile with "mpicc"  

```
$ module add openmpi
```

```
$ mpicc hello.c -o hello
```
  - run the binary (within a job) with the following command:  

```
$ mpirun --mca btl ^tcp hello
```

### Hint:

- see the solution at `/storage/brno2/home/jeronimo/MetaSeminar/20140206-ZCU/IB_hello`

# How to ... run a parallel/distributed computation V.

## Questions and Answers:

- *Is it possible to simultaneously use both OpenMP and MPI?*
  - Yes, it is. But be sure, how many processors your job is using
    - appropriately set the “-np” option (MPI) and the OMP\_NUM\_THREADS variable (OpenMP)
      - **OpenMPI:** a single process on each machine (`mpirun -pernode ...`) being threaded based on the number of processors (`export OMP_NUM_THREADS=$PBS_NUM_PPN`)

- Any other questions?



# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- **Another mini-HowTos ...**
- What to do if something goes wrong?
  
- Real-world examples

## Another mini-HowTos ... I.

- **how to make your application available within MetaVO?**
  - *commercial apps:*
    - **assumption:** you own a license, and the license allows the application to be run on our infrastructure (nodes not owned by you, located elsewhere, etc.)
    - once installed, we can **restrict its usage** just for you (or for your group)
  - *open-source/freeware apps:*
    - you can compile/install the app in your HOME directory
    - **OR** you can install/compile the app on your own and ask us to make it available in the software repository
      - compile the application in your HOME directory
      - **prepare a modulefile** setting the application environment
        - inspire yourself by modules located at `/packages/run/modules-2.0/modulefiles`
      - **test the app/modulefile**
        - `$ export MODULEPATH=$MODULEPATH:$HOME/myapps`
    - see [https://wiki.metacentrum.cz/wiki/Jak\\_si\\_sám\\_nainstalovat\\_aplikaci](https://wiki.metacentrum.cz/wiki/Jak_si_sám_nainstalovat_aplikaci)
  - **OR you can ask us for preparing the application for you**

## Another mini-HowTos ... II.

- **how to ask for nodes equipped by GPU cards?**
  - determine, **how many GPUs** your application will need (`-l gpu=X`)
    - consult the HW information page: <http://metavo.metacentrum.cz/cs/state/hardware.html>
  - determine, **how long** the application will run (if you need more, let us know)
    - `gpu_queue` ... maximum runtime 1 day
    - `gpu_long_queue` ... maximum runtime 1 week
  - make the submission:
    - `$ qsub -l nodes=1:ppn=4:gpu=1 -q gpu_long -l mem=10g -l walltime=4d ...`
    - specific GPU cards by restricting the cluster: `qsub -l nodes=...:cl_doom ...`
  - **do not change** the `CUDA_VISIBLE_DEVICES` environment variable
    - it's automatically set in order to determine the GPU card that has been reserved for your application
  - details about GPU cards performance within MetaVO:
    - see [http://metavo.metacentrum.cz/export/sites/meta/cs/seminars/seminar5/gpu\\_fila.pdf](http://metavo.metacentrum.cz/export/sites/meta/cs/seminars/seminar5/gpu_fila.pdf)
  - general information: [https://wiki.metacentrum.cz/wiki/GPU\\_stroje](https://wiki.metacentrum.cz/wiki/GPU_stroje)

## Another mini-HowTos ... III.

- **how to transfer large amount of data to MetaVO nodes?**
  - copying through the frontends/computing nodes may not be efficient (hostnames are *storage-XXX.metacentrum.cz*)
    - XXX = brno1, brno2, brno3-cerit, plzen1, budejovice1, praha1, ...
  - → connect directly to the storage frontends (via **SCP** or **SFTP**)
    - `$ sftp storage-brno1.metacentrum.cz`
    - `$ scp <files> storage-plzen1.metacentrum.cz:<dir>`
    - etc.
    - use FTP only together with the Kerberos authentication
      - otherwise insecure
- **how to access the data arrays?**
  - **easier:** use the SFTP/SCP protocols (suitable applications)
  - **OR mount the storage arrays directly to your computer**
    - [https://wiki.metacentrum.cz/wiki/Připojení\\_datových\\_úložišť\\_k\\_vlastní\\_pracovní\\_stanici\\_přes\\_NFSv4](https://wiki.metacentrum.cz/wiki/Připojení_datových_úložišť_k_vlastní_pracovní_stanici_přes_NFSv4)

## Another mini-HowTos ... IV.

### ■ how to get information about your quotas?

- by default, all the users have quotas on the storage arrays (per array)
  - may be different on every array
- to get an information about your quotas and/or free space on the storage arrays
  - **textual way:** log-in to a MetaCentrum frontend and see the “*motd*” (information displayed when logged-in)
  - **graphical way:**
    - *your quotas:* <https://metavo.metacentrum.cz/cs/myaccount/kvoty>
    - *free space:* <http://metavo.metacentrum.cz/pbsmon2/nodes/physical>

### ■ how to restore accidentally erased data

- the storage arrays (⇒ including homes) are regularly backed-up
  - several times a week
- → write an email to [meta@cesnet.cz](mailto:meta@cesnet.cz) specifying what to restore

## Another mini-HowTos ... V.

### ■ how to secure private data?

- by default, all the data are readable by everyone
- → use **common Linux/Unix mechanisms/tools** to make the data private
  - `r,w,x` rights for *user, group, other*
  - e.g., `chmod go= <filename>`
    - see `man chmod`
    - use “-R” option for recursive traversal (applicable to directories)
- → if you need a **more precise** ACL specification, use **NFS ACLs**
  - see [https://wiki.metacentrum.cz/wiki/Access\\_Control\\_Lists\\_na\\_NFSv4](https://wiki.metacentrum.cz/wiki/Access_Control_Lists_na_NFSv4)

### ■ how to share data among working group?

- ask us for creating a **common unix user group**
  - user administration will be up to you (GUI frontend is provided)
- **use common unix mechanisms** for sharing data among a group
  - see “`man chmod`” and “`man chgrp`”
- see [https://wiki.metacentrum.cz/wiki/Sdílení\\_dat\\_ve\\_skupině](https://wiki.metacentrum.cz/wiki/Sdílení_dat_ve_skupině)



## Another mini-HowTos ... VI.

- **how to perform cross-way submissions?**
  - our long-term goal is to **make the schedulers cooperate**
    - i.e., forward jobs which could be run by the other infrastructure
  - in the meantime, the cross-way submissions may become useful
    - it is necessary to explicitly specify the scheduling server

### From MetaCentrum frontends:

- skirit\$ qsub -q @wagap.cerit-sc.cz -l ...
- skirit\$ qstat -q @wagap.cerit-sc.cz
- skirit\$ qstat -f 12345.wagap.cerit-sc.cz
- skirit\$ qdel 12345.wagap.cerit-sc.cz
- ...

### From the CERIT-SC frontend:

- zuphux\$ qsub -q short@arien.ics.muni.cz -l ...
- zuphux\$ qstat -q @arien.ics.muni.cz
- zuphux\$ qstat -f 12345.arien.ics.muni.cz
- zuphux\$ qdel 12345.arien.ics.muni.cz
- ...

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- **What to do if something goes wrong?**
  
- Real-world examples

# What to do if something goes wrong?

1. check the MetaVO/CERIT-SC documentation, application module documentation
  - whether you use the things correctly
2. check, whether there haven't been any infrastructure updates performed
  - visit the webpage <http://metavo.metacentrum.cz/cs/news/news.jsp>
    - one may stay informed via an RSS feed
3. write an email to [meta@cesnet.cz](mailto:meta@cesnet.cz), resp. [support@cerit-sc.cz](mailto:support@cerit-sc.cz)
  - your email will create a ticket in our Request Tracking system
    - identified by a unique number → one can easily monitor the problem solving process
  - please, include **as good problem description as possible**
    - problematic job's JOBID, startup script, problem symptoms, etc.

# Overview

- Brief MetaCentrum introduction
- Brief CERIT-SC Centre introduction
  
- Grid infrastructure overview
- How to ... specify requested resources
- How to ... run an interactive job
- How to ... use application modules
- How to ... run a batch job
- How to ... determine a job state
- How to ... run a parallel/distributed computation
- Another mini-HowTos ...
- What to do if something goes wrong?
  
- **Real-world examples**

# Real-world examples

## ***Examples:***

- Maple
- Gaussian
- Gromacs
- Matlab (parallel & distributed & GPU)
- Ansys CFX
- Echo

## ■ demo sources:

```
/storage/brno2/home/jeronimo/MetaSeminar/20140206-ZCU
```

## **command:**

```
cp -r /storage/brno2/home/jeronimo/MetaSeminar/20140206-ZCU $HOME
```

# Real-world examples - Matlab

## *Parallel computations in Matlab*

- common Matlab case:
  - using the functions `parcluster` and `matlabpool`
  - providing the exact number of cores (parallel workers)
  - dealing with multiple instances on a single node (shared locations => problems), etc.

# Real-world examples - Matlab

## *Parallel computations in Matlab*

- common Matlab case:
  - using the functions `parcluster` and `matlabpool`
  - providing the exact number of cores (parallel workers)
  - dealing with multiple instances on a single node (shared locations => problems), etc.

# Real-world examples - Matlab

## *Parallel computations in Matlab*

- common Matlab case:
  - using the functions `parcluster` and `matlabpool`
  - providing the exact number of cores (parallel workers)
  - dealing with multiple instances on a single node (shared locations => problems), etc.
  
- **Parallel Matlab in MetaCentrum:**
  - `MetaParPool` function prepared to **automatically start the appropriate number of workers and to deal with multiple instances**
    - `size=MetaParPool('open')`
    - `size=MetaParPool('size')`
    - `MetaParPool('close')`



# Real-world examples - Matlab

Pa

## Simple usage:

- `MetaParPool('open');`

...

```
% your parallel computation
```

```
% (e.g., using parfor/spmd)
```

...

- `MetaParPool('close');`

- `MetaParPool` function prepared to **automatically start the appropriate number of workers and to deal with multiple instances**

- `size=MetaParPool('open')`

- `size=MetaParPool('size')`

- `MetaParPool('close')`

# Real-world examples - Matlab

## *Distributed computations in Matlab*

- common Matlab case:
  - submitting sub-jobs during the run of master computation
    - hard to configure
    - may lead to inappropriate waiting times as well as wasting of resources

# Real-world examples - Matlab

## *Distributed computations in Matlab*

- common Matlab case:
  - submitting sub-jobs during the run of master computation
    - hard to configure
    - may lead to inappropriate waiting times as well as wasting of resources

# Real-world examples - Matlab

## *Distributed computations in Matlab*

- common Matlab case:
  - submitting sub-jobs during the run of master computation
    - hard to configure
    - may lead to inappropriate waiting times as well as wasting of resources
  
- **Distributed Matlab in MetaCentrum:**
  - **MetaGridPool** function prepared to **automatically start the appropriate number of workers as well as to deal with multiple instances**
    - `jobmanager=MetaGridPool('open')`
    - `size=MetaGridPool('size')`
    - `MetaGridPool('close')`

# Real-world examples - Matlab

## *Distributed computations in Matlab*

- common Matlab case:

### Simple usage:

```
jm=MetaGridPool('open');
```

```
...
```

```
% your computation
```

- ```
job = createJob(jm);  
createTask(job, ...);  
submit(job);  
wait(job);  
...  
MetaGridPool('close');
```

ces

# Thank You for attending!



**rebok@ics.muni.cz**

**[www.cesnet.cz](http://www.cesnet.cz)**

**[www.metacentrum.cz](http://www.metacentrum.cz)**

**[www.cerit-sc.cz](http://www.cerit-sc.cz)**

## Useful seminars:

### IT4innovations:

- Úvod do obecných výpočtů na grafických kartách (7. února 2014)
- Programování v CUDA (19. února 2014)
- další viz [http://www.it4i.cz/?post\\_type=post](http://www.it4i.cz/?post_type=post)

### Sprinx Systems:

- Úvod do CUDA -- 1 den -- 14.3.2014, 21.3.2014, 2.5.2014 (pá)
- Výpočty na grafických procesorech -- 2 dny -- 10.-11.4.2014 (čt, pá)
- Výpočty na grafických procesorech (praktikum) -- 1 den -- 25.4.2014 (pá)
- MPI a OMP -- 1 den (nové) -- 14.3.2014 (pá)
- Kompilátory Intel a PGI -- 1 den (nové) -- 2.5.2014 (pá)
- R na CUDA -- 1 den (nové) -- 2.5.2014 (pá)
- viz <http://hpc.sprinx.cz/hpc-sluzby/prehled-sluzeb/hpc-skoleni>