

Univerza
v Ljubljani

Fakulteta
za gradbeništvo
in geodezijo

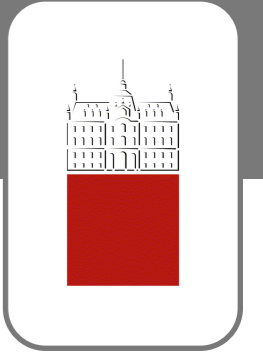


Visoko-propustna računska okolja v gradbeništvu

Ljubljana, 18.4.2008

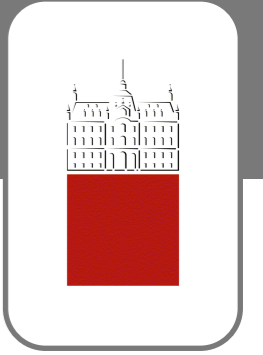
dr. Matevž Dolenc

mdolenc@itc.fgg.uni-lj.si

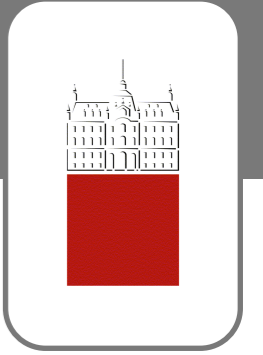


- ▶ Uvod
- ▶ Visoko-propustno računsko okolje
- ▶ Uporabniški scenariji
- ▶ Projekt: Visoko-propustno računsko okolje za analizo potresnega tveganja
- ▶ Zaključek

Potrebe po novih računskih virih



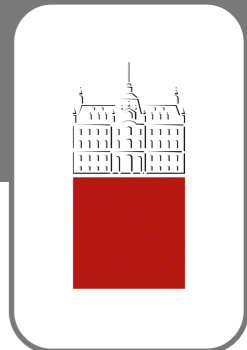
- ▶ Raziskave in računske metode postajajo vse bolj zapletene
 - Novi, natančnejši modeli
 - Časovna omejenost raziskav
- ▶ Integracija raznolikih podatkov
- ▶ Delo na daljavo



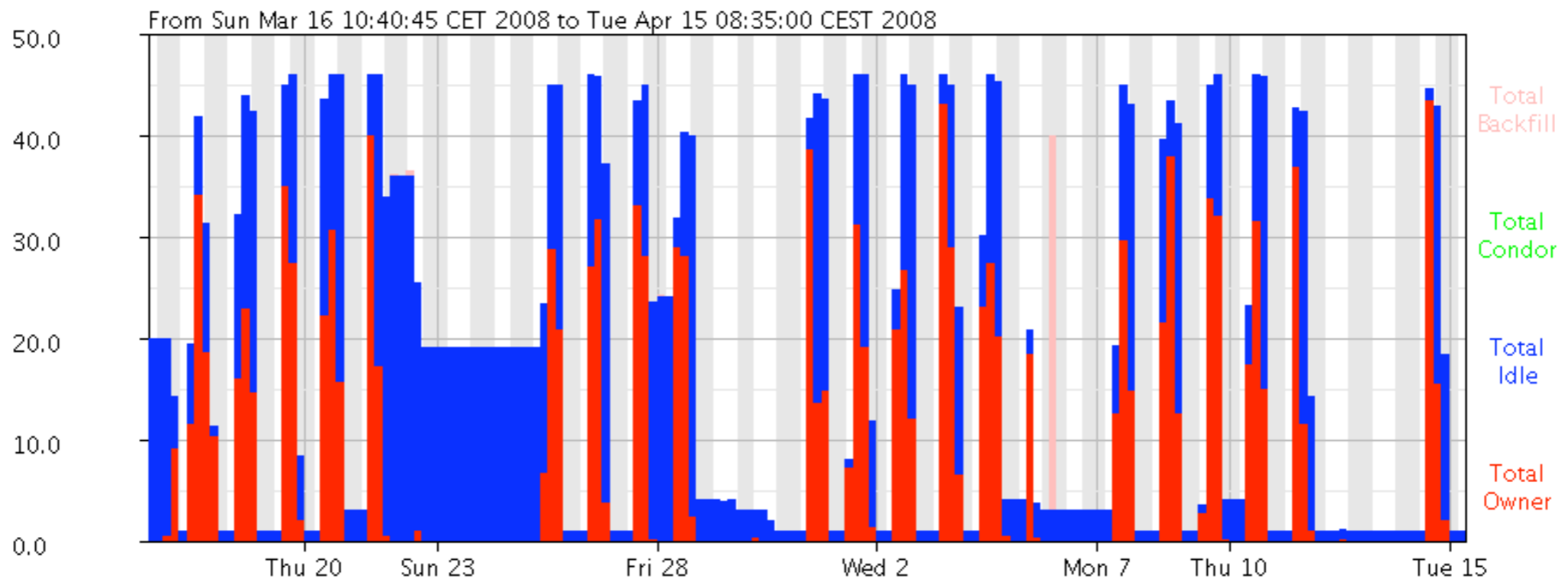
- ▶ HPC - High-performance computing
 - Visoko-zmogljivo računsko okolje
 - statična računska okolja (gruče); enoviti problemi
 - protokoli: MPI, PVM, ...

- ▶ HTC - High-throughput computing
 - Visoko-propustno računsko okolje
 - dinamična računska okolja; parametrične študije - možnost paralelizacije
 - programski sistemi: Condor, Torque, SGE, Xgrid

Izkoriščenost računalnikov v računalniških učilnicah UL-FGG

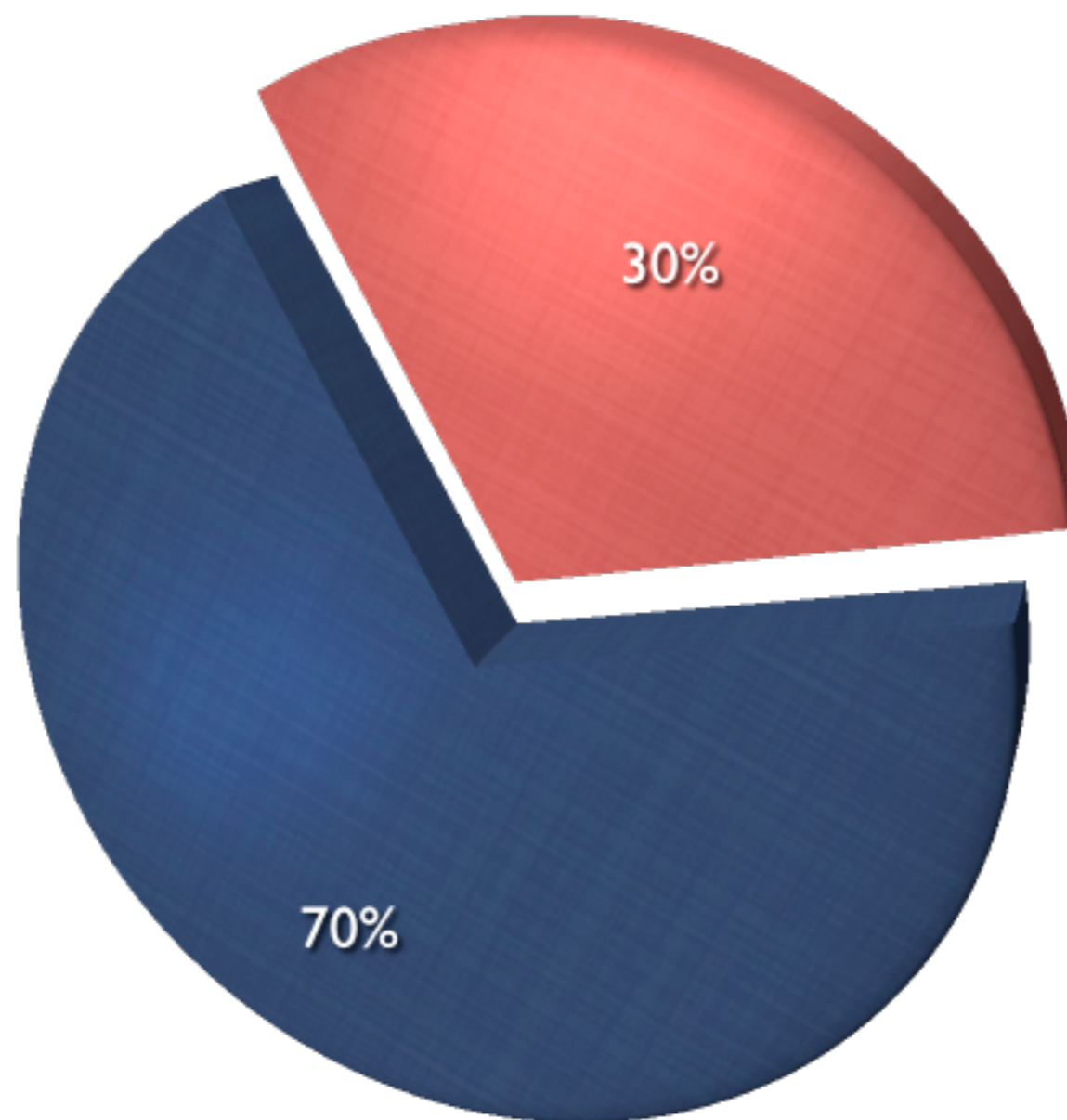
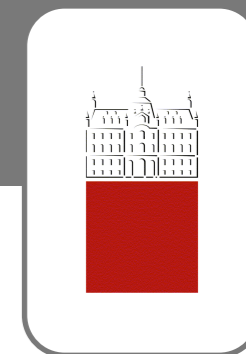


UL-FGG Condor Pool Machine Statistics for Month



Vir: <http://grmada.fgg.uni-lj.si/condor-view>

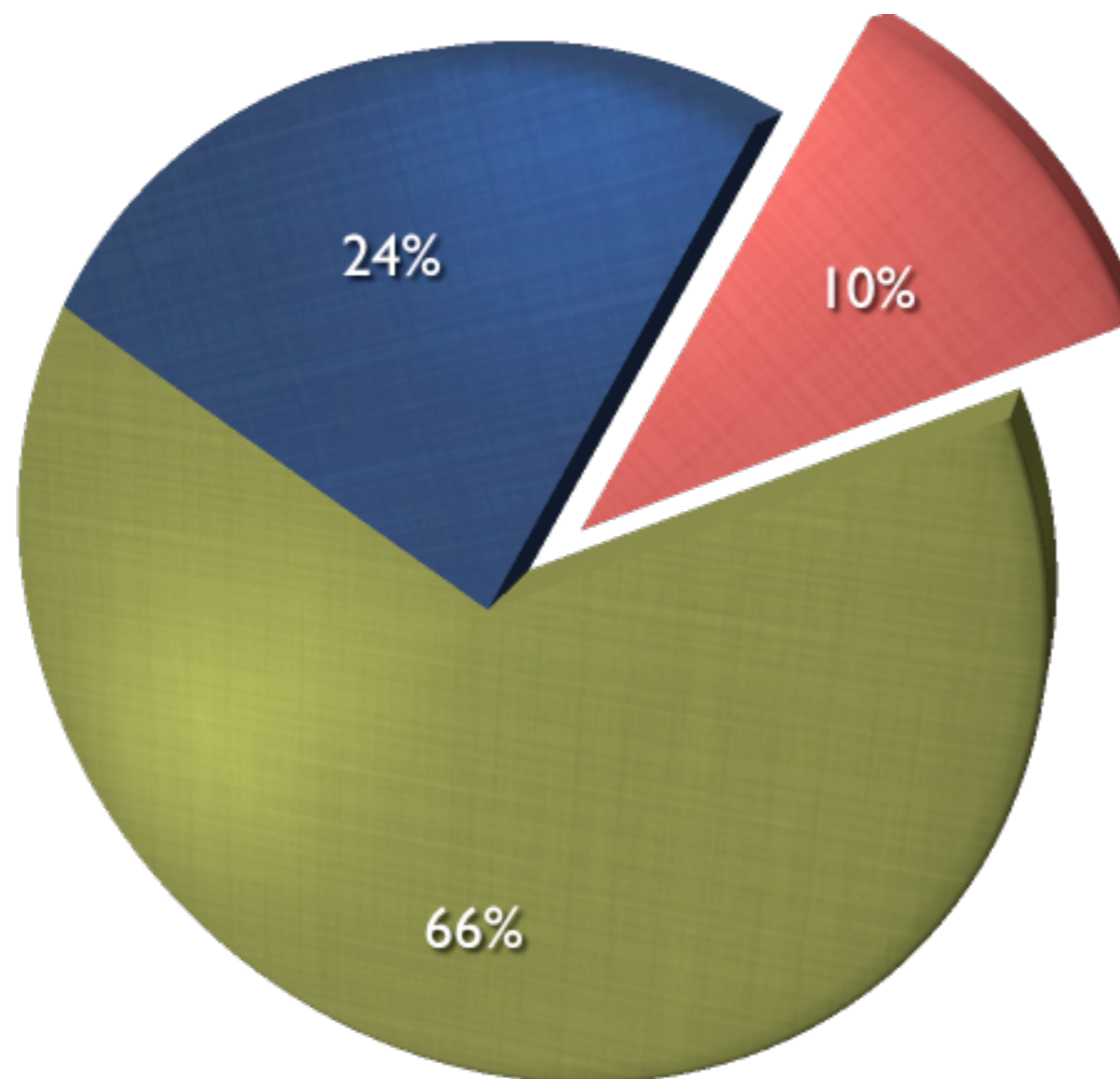
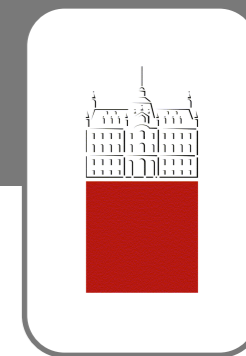
Izkoriščenost računalnikov v računalniških učilnicah UL-FGG



● Računalnik prost

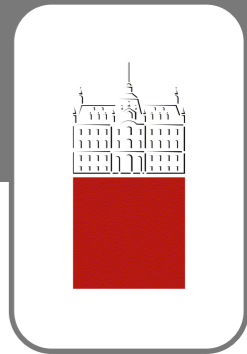
● Računalnik zaseden

Izkoriščenost računalnikov v računalniških učilnicah UL-FGG

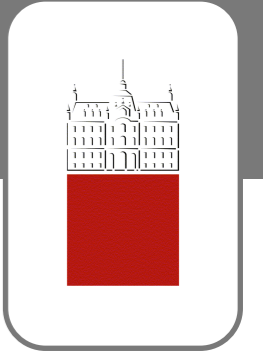


● Računalnik prost ● Računalnik zaseden ● Računalnik ni vkllopljen

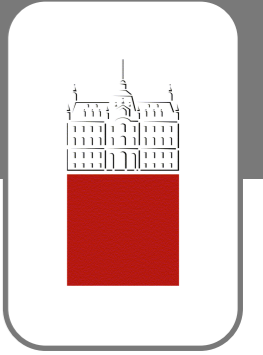
Izkoriščenost računalnikov v računalniških učilnicah UL-FGG



Računalniška učilnica I/5, UL-FGG

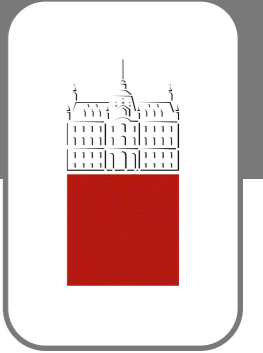


- ▶ Peter mora opraviti parametrično študijo
- ▶ Parametrična študija $F(x, y, z)$, kjer x lahko zavzame 20 vrednosti, y 10 vrednosti in z 3 vrednosti
 - število kombinacij: $20 \cdot 10 \cdot 3 = 600$
 - za izračun funkcije $F(x, y, z)$ so potrebne ≈ 3 ure
 - prenos podatkov: $(x, y, z) \approx 5$ MB, $F(x, y, z) \approx 50$ MB

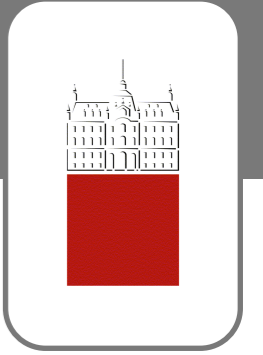


- ▶ BAT procedure
 - zaporedno izvajanje na enem računalniku
 - “vzporedno” izvajanje na večih računalnikih
- ▶ Programiranje svoje rešitve
- ▶ Obstaja še kaj?

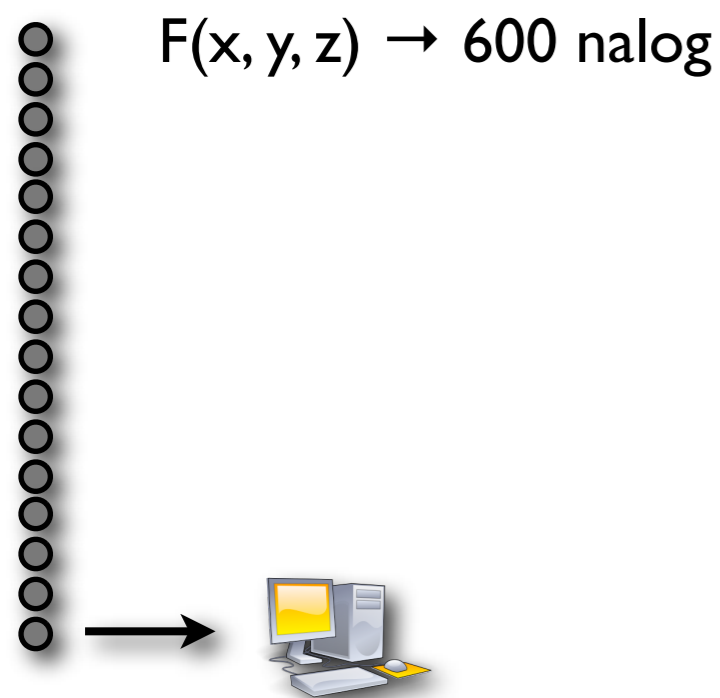
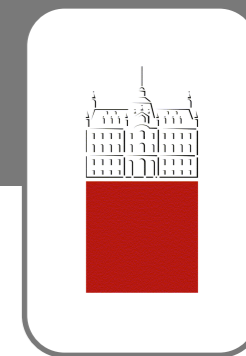




- ▶ Različna imena
 - Volunteer computing, Public Resource Computing, Meta Computing, Internet computing
- ▶ Uporabniki donirajo proste računske cikle za različne projekte
 - SETI@Home, Einstein@Home, ...
- ▶ Berkeley Open Infrastructure for Network Computing (BOINC)
 - <http://boinc.berkeley.edu>

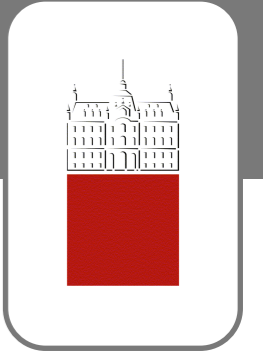


- ▶ Kje dobim Condor?
 - <http://www.cs.wisc.edu/condor>
- ▶ Operacijski sistemi:
 - Windows XP, Windows Vista, Linux, Mac OS X
- ▶ Za namestitev osebnega Condor-ja ne potrebujete administratorskega gesla
- ▶ Po namestitvi lahko Peter pošlje naloge parametrične študije na svoj osebni Condor

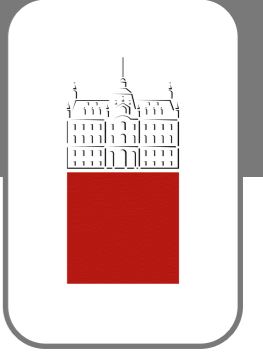


- ▶ Kakšne so prednosti?
- ▶ Osebni Condor bo ...
 - nadzoroval potek izvajanja nalog
 - naloge izvajal v predpisanem vrstnem redu
 - hranil dnevnik dogodkov
- ▶ Seveda se naloge še vedno izvajajo zaporedno.

Kako Condor-ju pošljemo nalogo

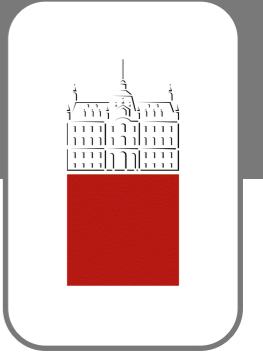


- ▶ Paketna naloga
 - Naloga ne zahteva interakcije z uporabnikom
 - Naloga uporablja `STDIN`, `STDOUT`, `STDERR`
- ▶ Nalogo opišemo
 - Tekstovna datoteka
 - Opišemo lahko več nalog
 - Naloge imajo lahko različne vhodne/izhodne argumente



- ▶ Condor definira različna okolja za podporo raznovrstnih programov
 - Okolje določa osnovne parametre v katerih se program izvaja
- ▶ Definirana okolja (angl. universe)
 - Standard: podpora za checkpointing
 - Vanilla: za programe, ki jih ne moremo povezati s Condor knjižnicami
 - PVM: paralelno izvajanje programov
 - Java: izvajanje Java programov

Primer naloge



```
universe          = vanilla
requirements      = OpSys == "WINNT51"
environment       = path=c:\winnt\system32
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

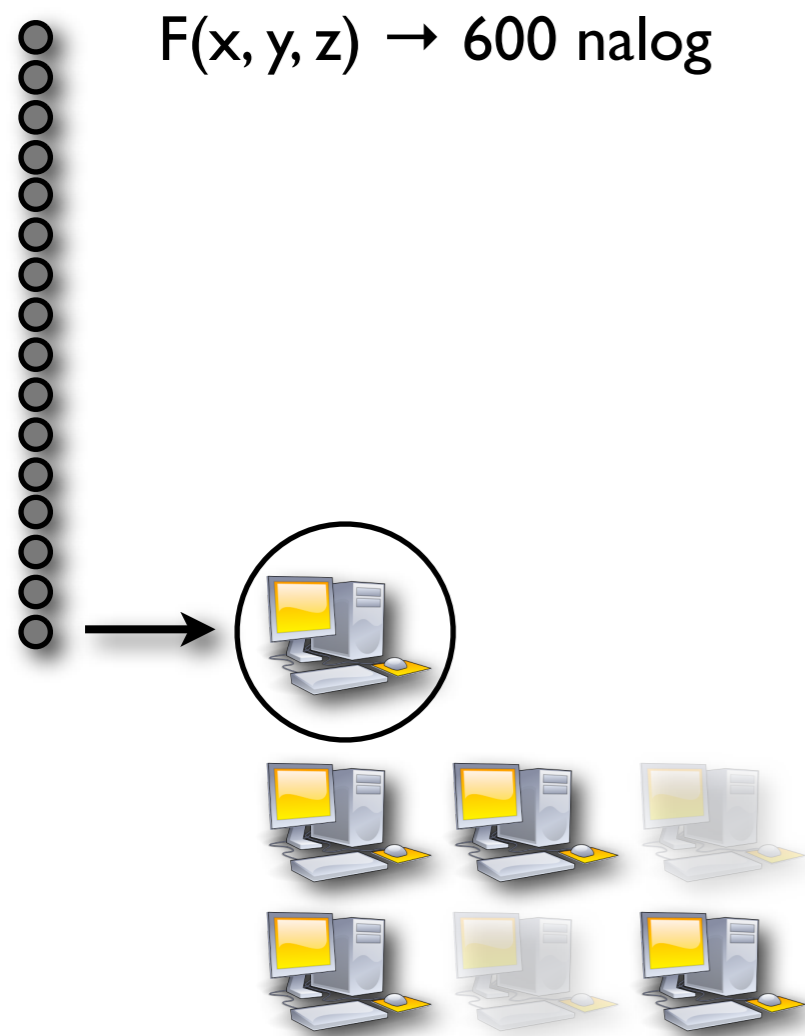
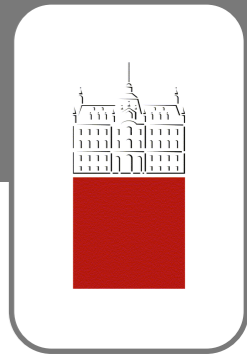
executable        = print-ip.bat
output            = print-ip.out
error             = print-ip.err
log              = print-ip.log

queue
```

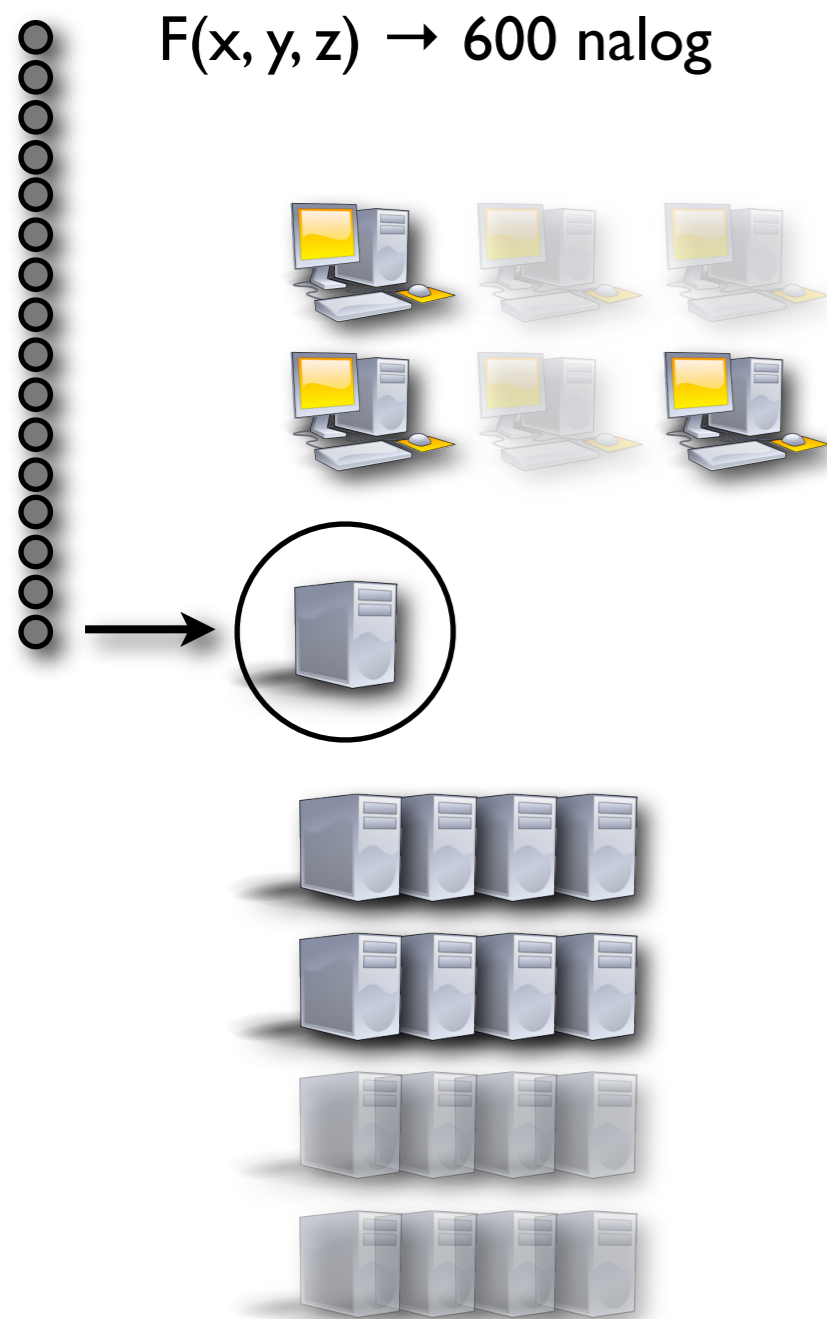
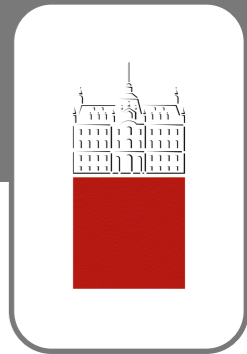
print-ip.sub

```
@echo off
echo Start.
echo Here is the output from "ipconfig" command:
c:\windows\system32\ipconfig
echo End.
```

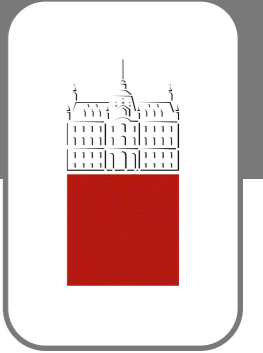
print-ip.bat



- ▶ Peter lahko uporabi tudi računalnike svojih sodelavcev ampak samo pod pogojem, da ji lahko tudi oni uporabljajo za izvajanje analiz
 1. Peter svoj računalnik določi za “centralni manager”
 2. Na preostale računalnike namesti Condor
- ▶ Peter lahko sedaj izvaja sočasno več nalog na različnih računalnikih



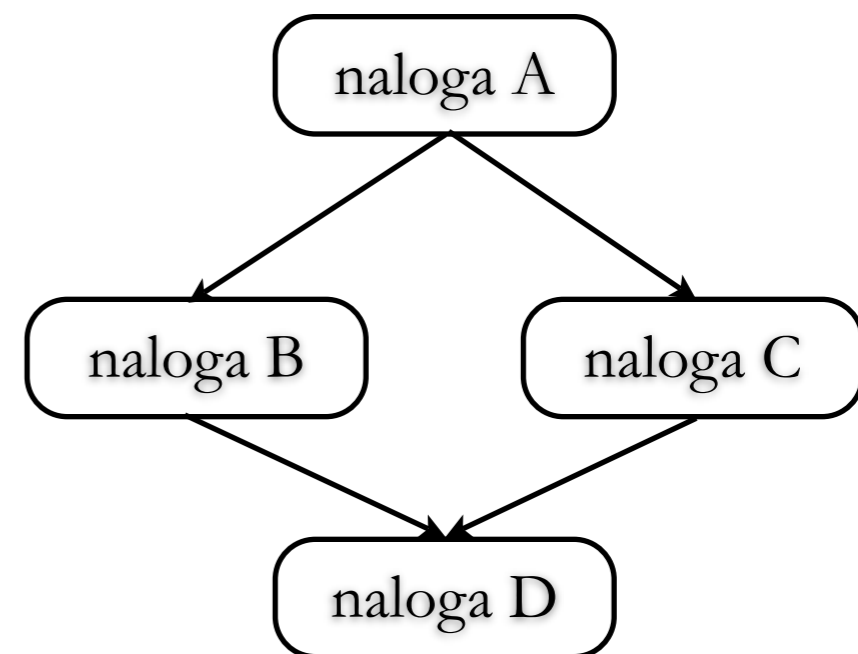
- ▶ V Petrovi organizaciji kupijo namenske Condor strežnike
 1. Za “centralni manager” se določi enega izmed namenskih strežnikov
 2. Na namenske strežnike se namesti Condor
- ▶ Peter in sodelovci lahko sedaj še bolj učinkovito uporabljajo Condor



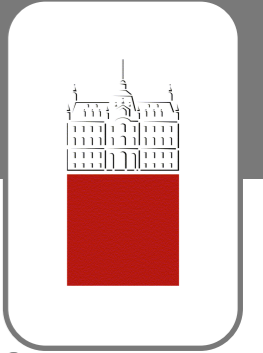
- ▶ **Directed Acyclic Graph Manager (DAGMan)**
 - podsistem Condor-ja omogoča določitev odvisnosti med nalogami
 - posamezna naloga predstavlja vozlišče grafa
 - naloga ima lahko poljubno število staršev oz. otrok (ne sme biti zank)

```
Job A a.sub
Job B b.sub
Job C c.sub
Job D d.sub
Parent A Child B C
Parent B C Child D
```

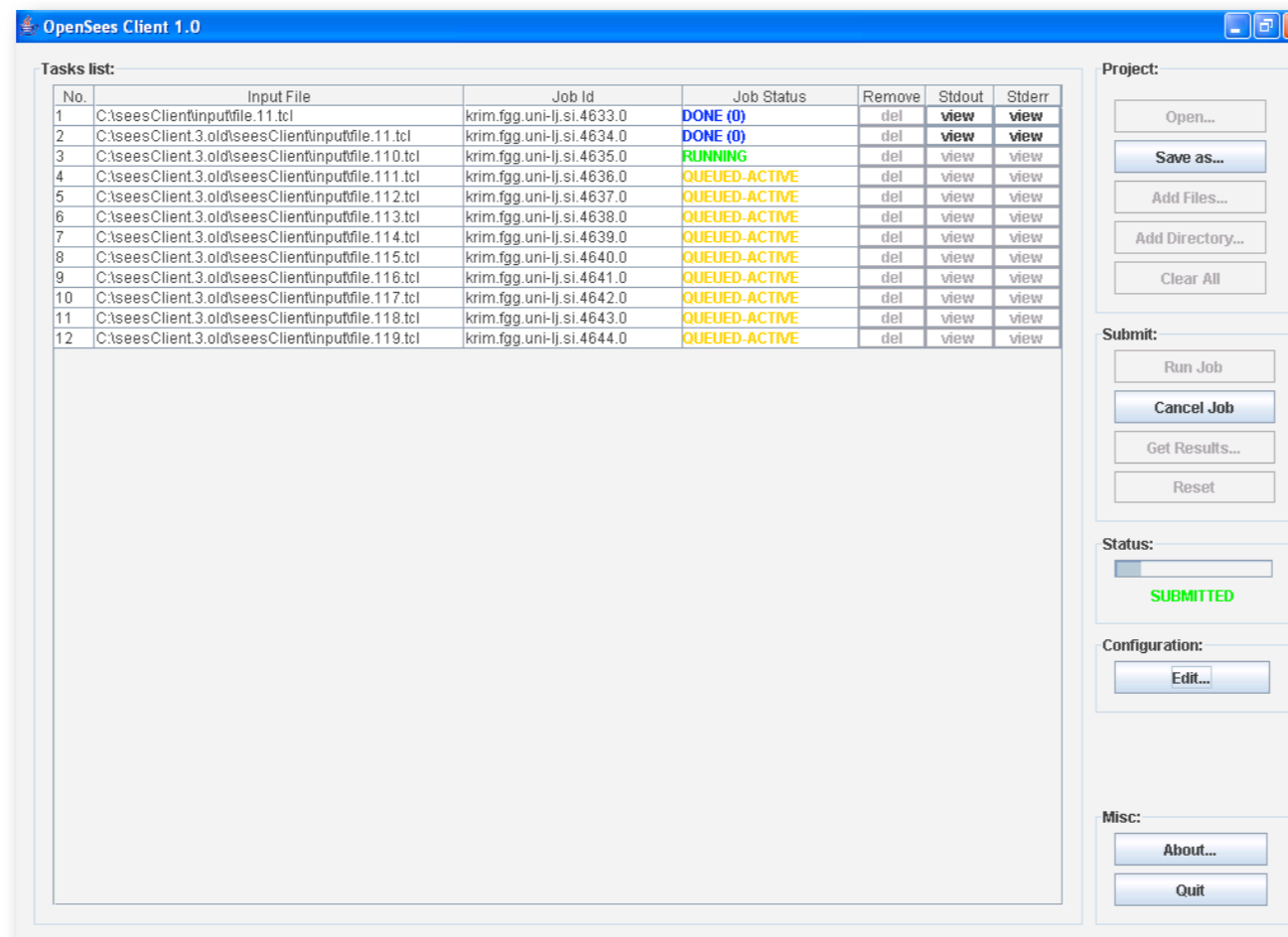
diamond.dag



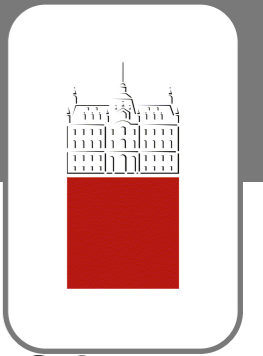
Distributed Resource Management Application API (DRMAA)



- ▶ Programski vmesnik za različne programske jezike (C/C++, Java, Python, Perl, Ruby)
- ▶ OpenDSP: WS implementacija DRMAA API-ja

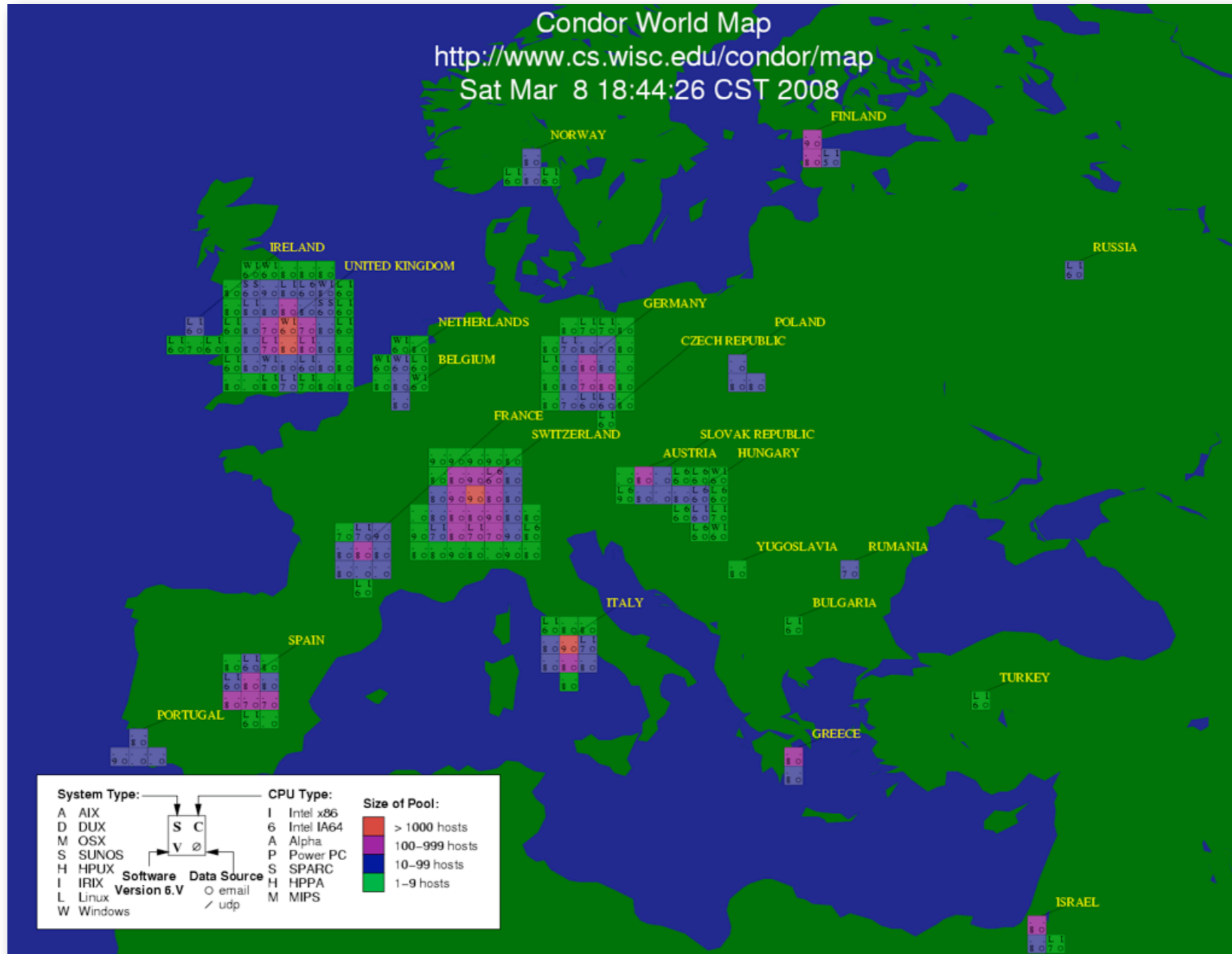
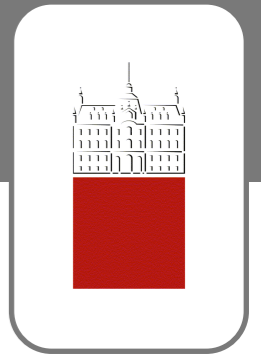


Migracija nalog in “checkpointing”



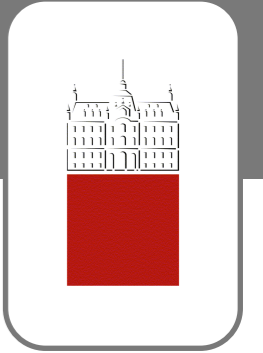
- ▶ Condor samodejno poskrbi za migracijo nalog, ki se niso dokončale
- ▶ Checkpointing
 - Condor lahko nalogo nadaljuje od mesta v programu, kjer se je naloga ustavila
 - Program mora biti povezan s Condor knjižnicami (program ni potrebno spreminjati)
 - Checkpointing ne deluje na operacijskem sistemu Windows.

Uporaba Condor-ja v Evropi



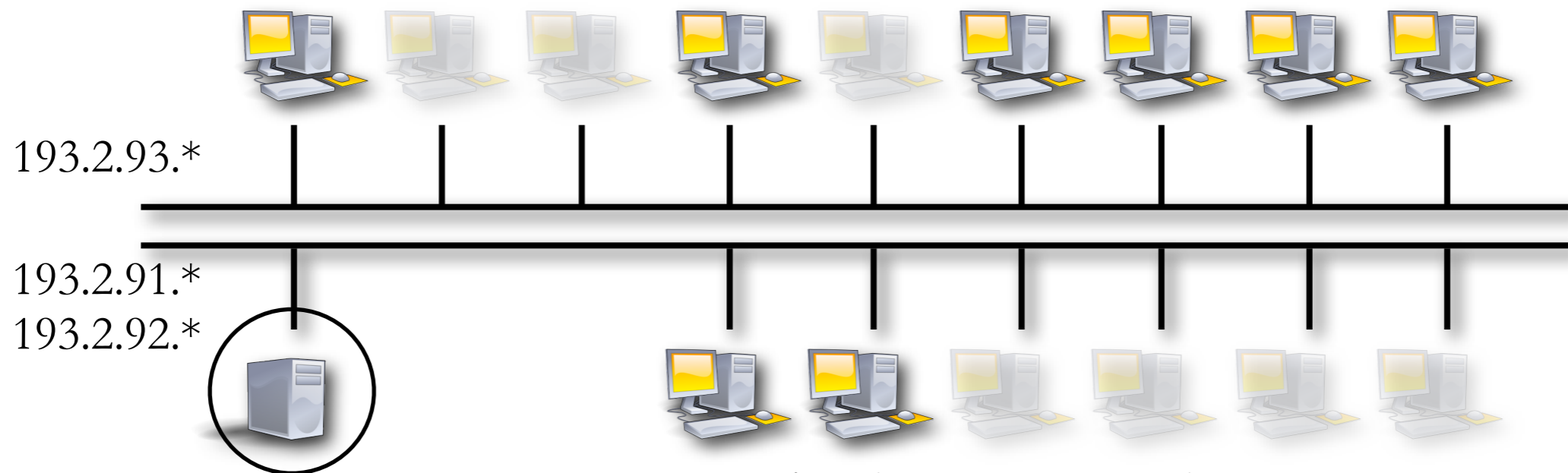
Vir: <http://www.cs.wisc.edu/condor/map/>

Condor na UL-FGG



Condor skupina računalnikov:

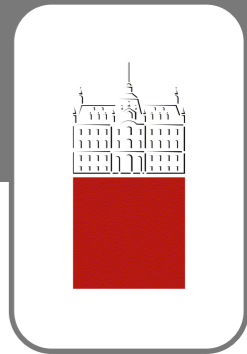
- ▶ Računalniški učilnici I/5 in III/6
- ▶ 50 računalnikov, Windows XP
- ▶ onemogočeno pošiljanje Condor nalog



Condor skupina računalnikov:

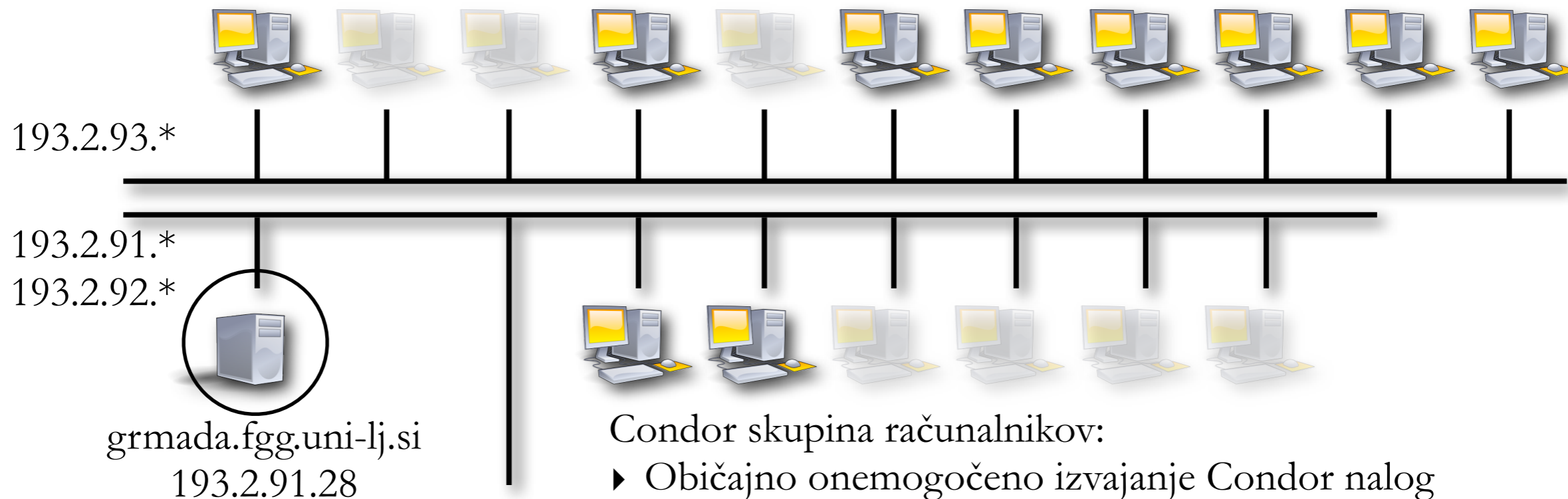
- ▶ Običajno onemogočeno izvajanje Condor nalog

Condor na UL-FGG - želje/načrti



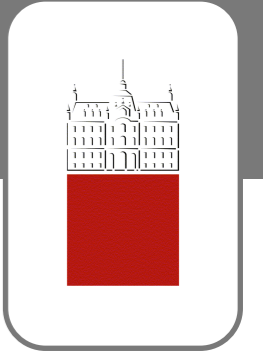
Condor skupina računalnikov:

- ▶ Računalniški učilnici I/5 in III/6 + vključitev preostalih učilnic (skupaj je 114 računalnikov v računalniških učilnicah)
- ▶ 50 računalnikov, Windows XP
- ▶ onemogočeno pošiljanje Condor nalog



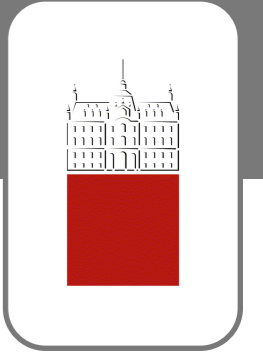
Skupina namenskih Condor strežnikov:

- ▶ Windows XP, Linux
- ▶ uporaba virtualnih računalnikov



- ▶ Zagon programa lokalno na računalniku
 - matlab.bat
- ▶ Možni razlogi za težave
 - pridobitev licence ob zagonu
- ▶ Rešitve
 - Prevajanje m-datotek v samostojno izvedljive programe
 - Test MATLAB “kompatibilnih” programov (SciLAB, Octave)

Condor in MATLAB



```
universe          = vanilla
requirements      = OpSys == "WINNT51"
environment       = path=c:\winnt\system32
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

Executable        = matlab.bat
Log               = matlab.log
Error             = matlab.err
Output           = mat.out

Arguments         = -r testrun
transfer_input_files = testrun.m

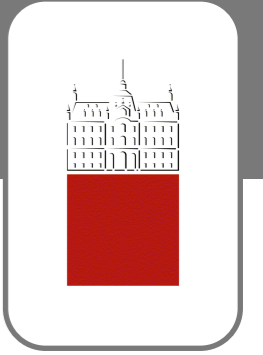
queue
```

matlab.sub

```
"c:\Program Files\MATLAB\R2007a\bin\win32\matlab.exe" -nodesktop -nosplash %1 %2
```

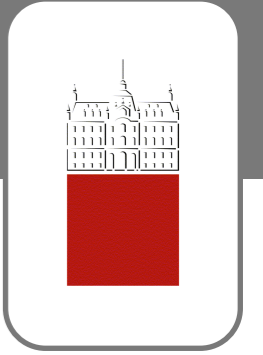
matlab.bat

Condor na UL-FGG - ocena stanja



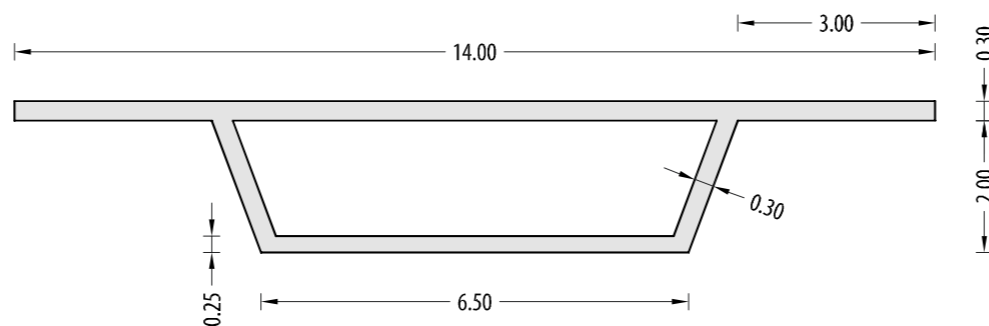
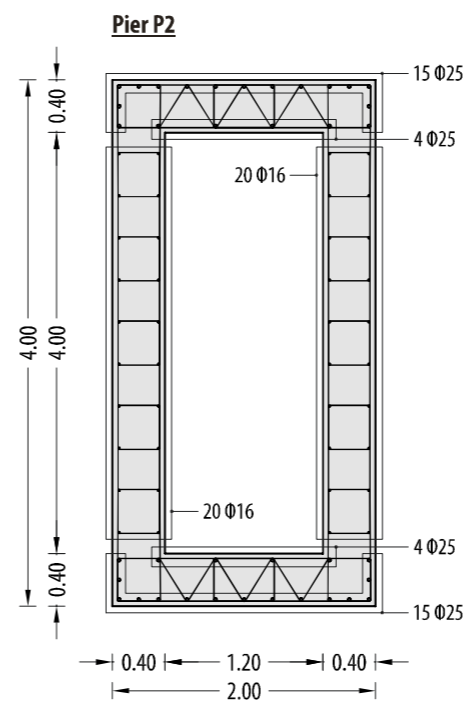
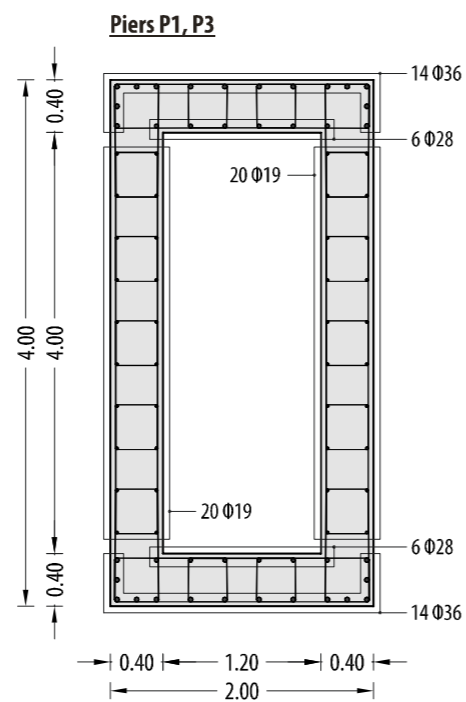
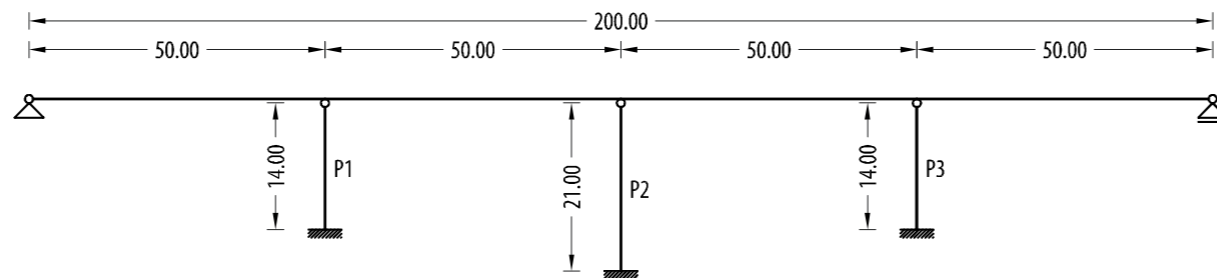
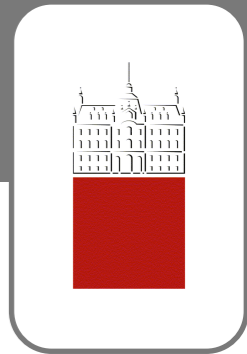
- ▶ Prva namestitvev: 2005
- ▶ Namestitvev v računalniških učilnicah: 2006
- ▶ Uporaba
 - Projekt InteliGrid
 - Projekt DataMiningGrid
 - Raziskovalci IKPIR
- ▶ Slabosti
 - Dokumentacija
 - Zračunavanje, prioritete, uporabniški vmesnik, ...

Uporabniški scenarij: parametrične študije v potresnem inženirstvu

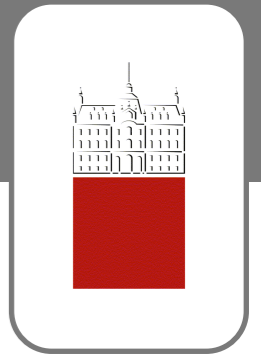


- ▶ IDA (Incremental Dynamic Analysis) analiza
 - parametrična analiza s katero za skupino akcelerogramov in za različne stopnje potresne intenzitete določimo potresni odziv konstrukcije
 - večja kot je skupina akcelerogramov bolj zanesljivo se lahko določi potresno tveganje
 - proporcionalno z večanjem skupine akcelerogramov se povečujejo tudi računski časi

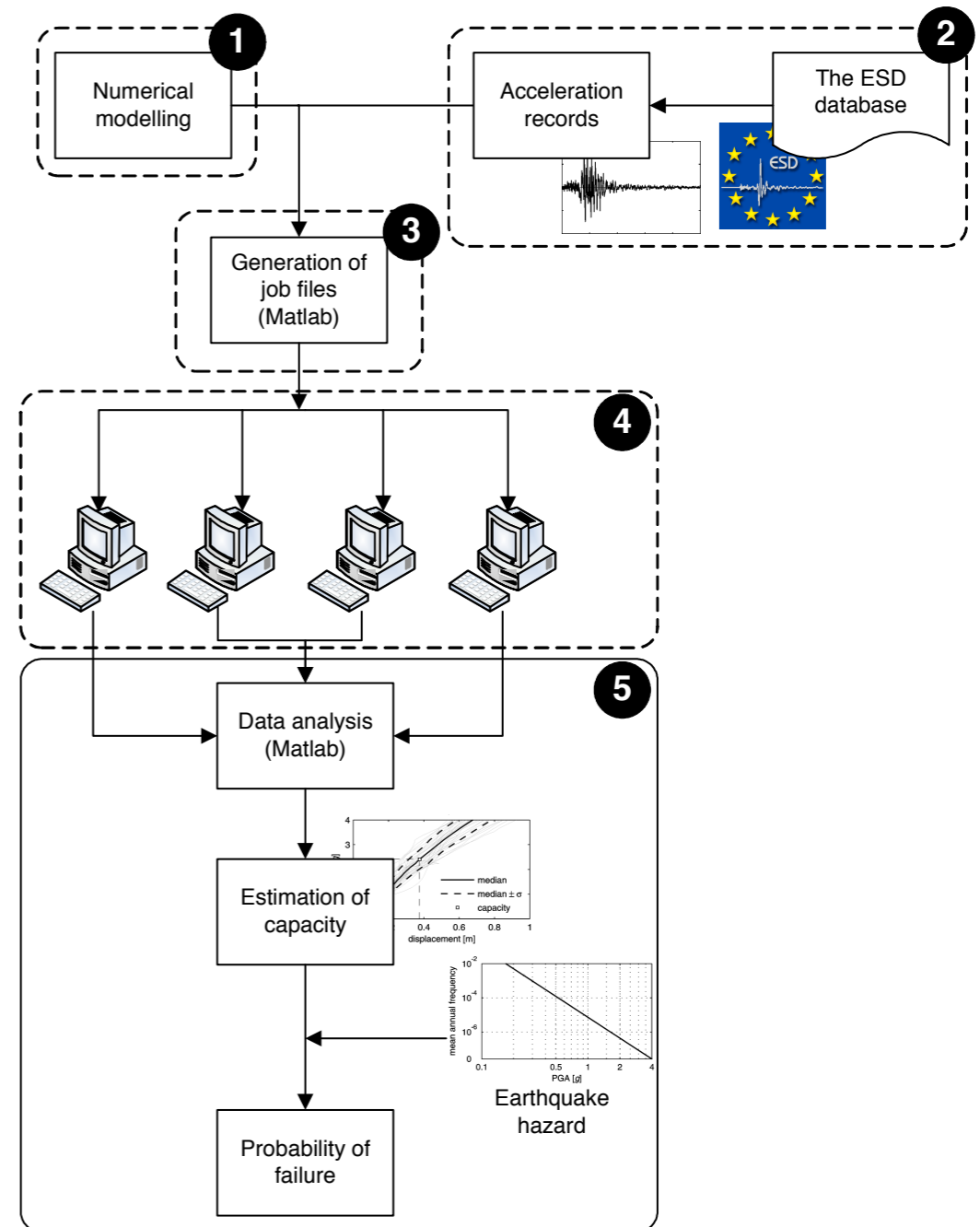
Testna konstrukcija



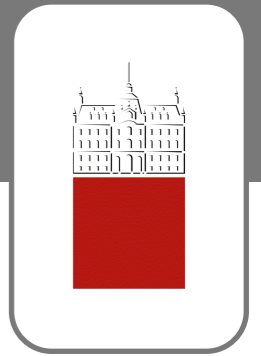
Postopek IDA analize



1. numerično modeliranje
2. izbira potresov
3. priprava opisane naloge
4. nelinearna analiza
5. analiza rezultatov



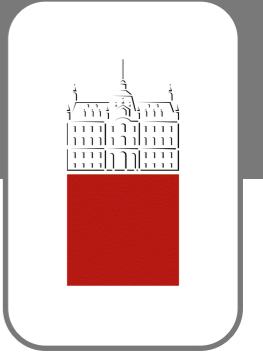
Ocena učinkovitosti uporabe visoko-propustnega računskega okolja



- ▶ Število analiz: 280
- ▶ Povprečno trajanje analize: 13 min

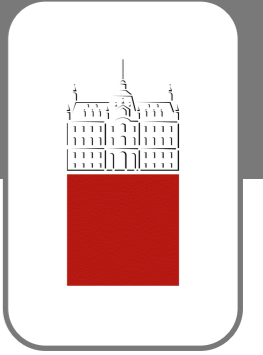
Število računalnikov	Trajanje izračuna [ur]	Faktor
1	61.3	1
5	14.7	4.17
10	7.1	8.63
25	2.5	24.52

Projekt: Visoko-propustno računsko okolje za analizo potresnega tveganja



- ▶ **Naslov:**
 - Visoko-propustno računsko okolje za analizo potresnega tveganja
- ▶ **Trajanje:**
 - februar 2008 - januar 2011
- ▶ **Obseg:**
 - 1550 raziskovalnih ur letno

Projekt: Visoko-propustno računsko okolje za analizo potresnega tveganja



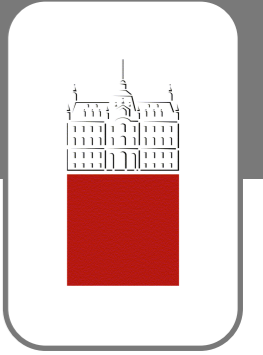
▶ Partnerji

- UL-FGG: IKPIR (nosilec doc. dr. Matjaž Dolšek)
- dr. Jaka Zevnik (ELEA iC d.o.o.)
- dr. Peter Kante (Primorje d.d.)
- dr. Iztok Peruš

▶ Ključne besede

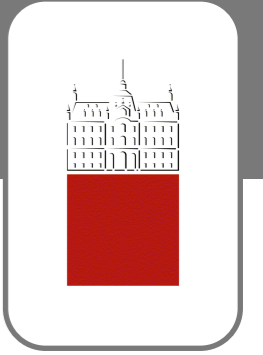
- potresno tveganje, verjetnostna potresna analiza, potresnoodporno projektiranje, visoko-propustna računsko okolje, tehnologija grid, Web 2.0

Projekt: Visoko-propustno računsko okolje za analizo potresnega tveganja



► Predvideni rezultati

- izboljšati postopke za določitev potresnega tveganja
- razviti uporabniško prijazna orodja za oceno potresnega tveganja na različnih nivojih
- vzpostaviti prijazno visoko-propustno računsko okolje
- prenos pridobljenih znanj v prakso



- ▶ Izkoriščenost računalnikov
- ▶ Računska okolja
 - HPC \neq HTC
 - Condor, Condor na UL-FGG
- ▶ Primeri uporabe
 - Parametrične študije
- ▶ Kako naprej
 - Delo v okviru projekta
 - Strojna oprema