

# HTCondor za **inženirje**

Računska okolja in parametrične študije



Matevž Dolenc  
[matevzdolenc.com](http://matevzdolenc.com)

# Vsebina

Uvod

Visoko-propustno računsko okolje

Uporabniški scenariji

Primer: Visoko-propustno računsko okolje za analizo potresnega tveganja

# Potrebe po novih računskih virih

Raziskave in računske metode postajajo vse bolj zapletene

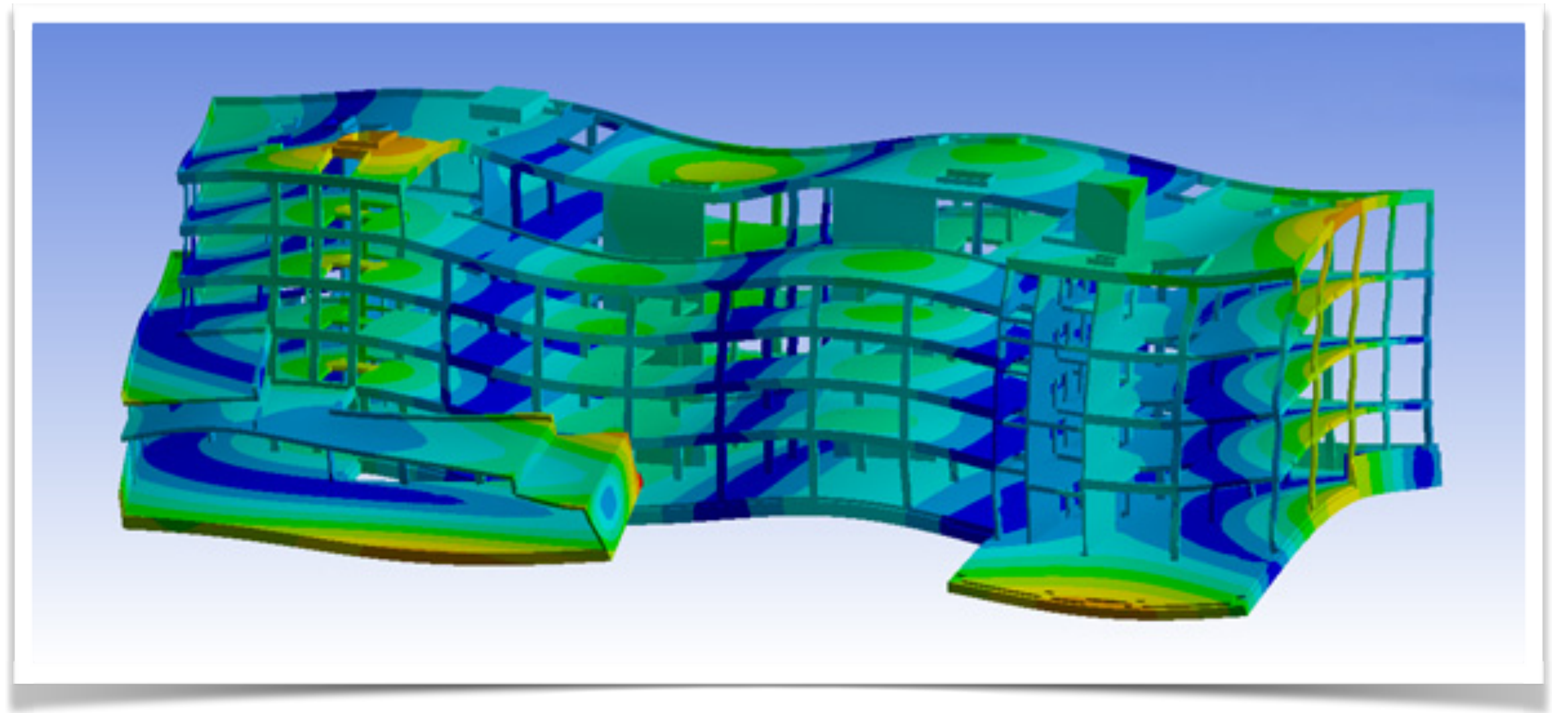
Novi, natančnejši modeli

Časovna omejenost raziskav

Integracija raznolikih podatkov

Strukturirani in nestrukturirani podatki

Delo na daljavo



# HPC ≠ HTC

## HPC - High-performance computing

Visoko-zmogljivo računsko okolje

statična računska okolja (gruče)

enoviti problemi

protokoli: MPI (Message Passing Interface), PVM, ...

## HTC - High-throughput computing

Visoko-propustno računsko okolje

dinamična računska okolja

parametrične študije - možnost paralelizacije

programski sistemi: HTCondor, Torque, ...

# Problem

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  $\approx 3$  ure

prenos podatkov:  $(x, y, z) \approx 5$  MB,  $F(x, y, z) \approx 50$  MB

# Rešitev?

BAT, BASH procedure

zaporedno izvajanje na enem računalniku

“vzporedno” izvajanje na večih računalnikih

Programiranje svoje rešitve

Obstaja še kaj?



# Rešitev: osebni HTCondor

Kje dobim Condor?

<https://research.cs.wisc.edu/htcondor/>

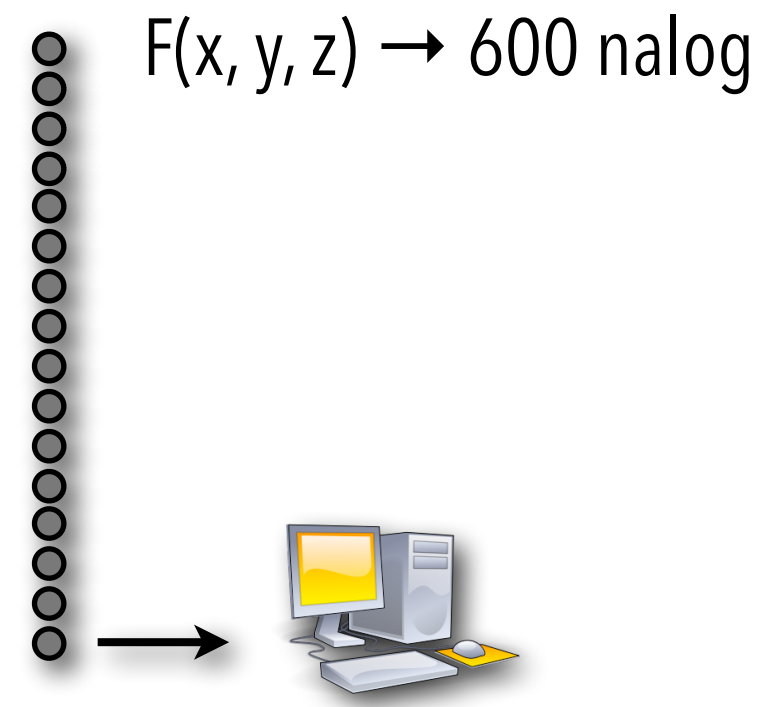
Operacijski sistemi:

Windows, Linux, Mac OS X

Za namestitev osebnega Condor-ja ne potrebujete administratorskega gesla

Po namestitvi lahko Peter pošlje naloge parametrične študije na svoj osebni HTCondor računski sistem

# Petrov osebni HTCondor računski sistem



## Kakšne so prednosti?

Osebni HTCondor bo ...

nadzoroval potek izvajanja nalog

naloge izvajal v predpisanem vrstnem redu

hranil dnevnik dogodkov

Seveda se naloge še vedno izvajajo zaporedno.



# Kako pripravimo 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

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

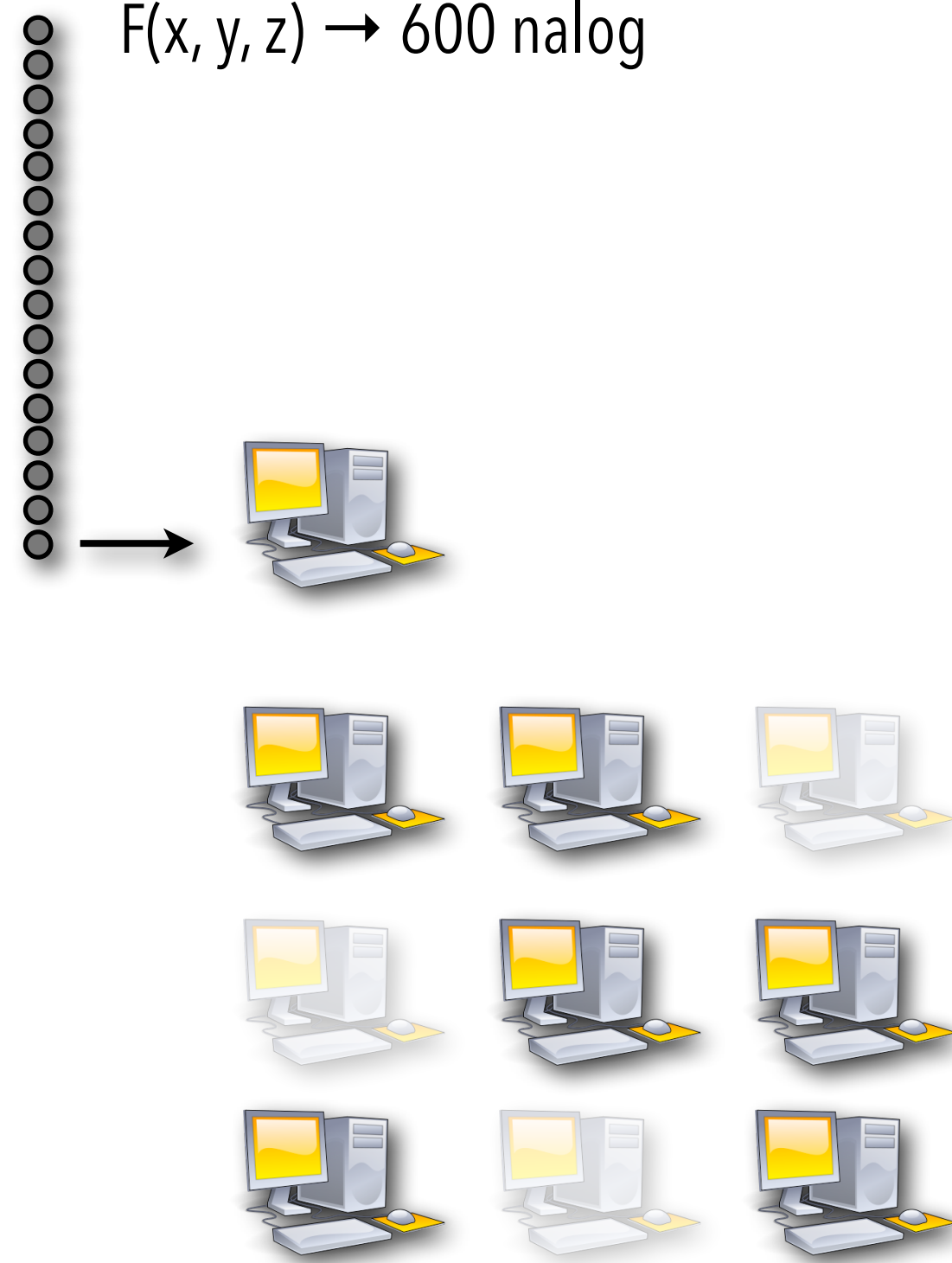
```
executable    = print-ip.bat
output        = print-ip.out
error         = print-ip.err
log           = print-ip.log
```

```
queue
```

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

# Petrov osebni HTCondor računski sistem

$F(x, y, z) \rightarrow 600$  nalog



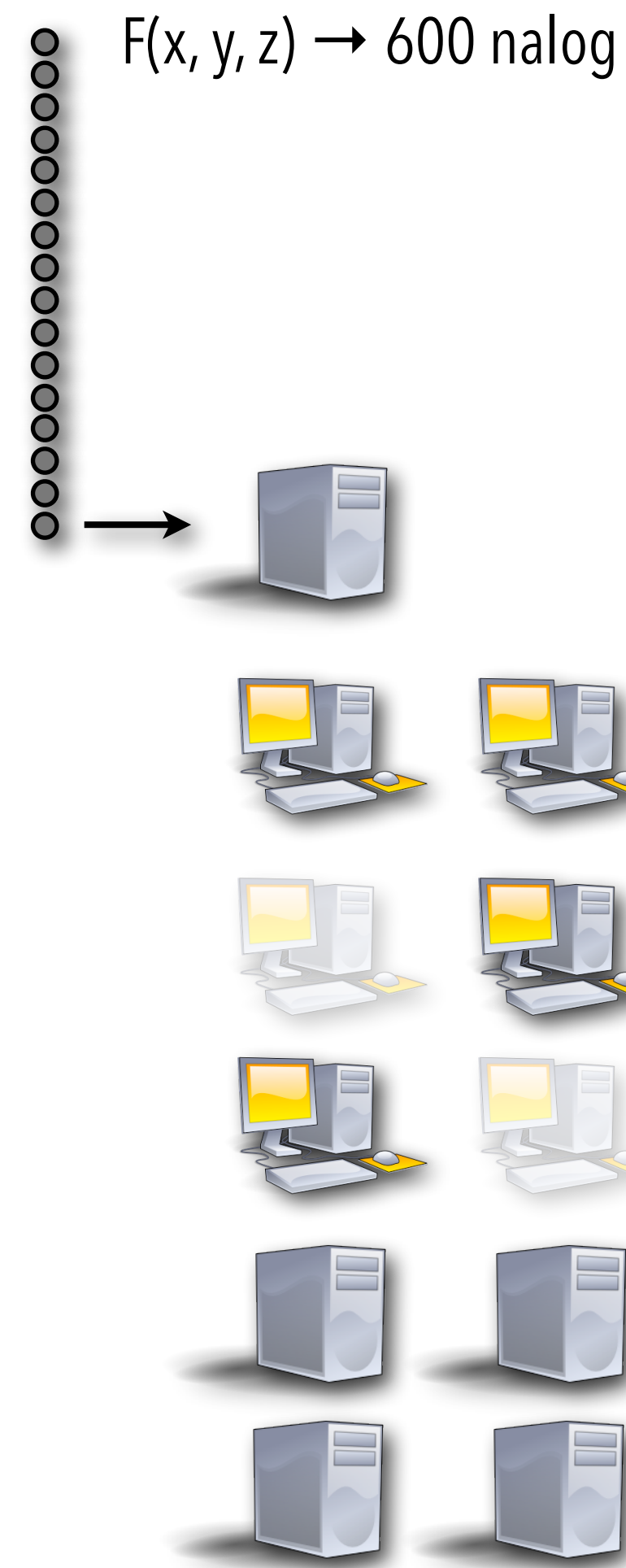
Peter lahko uporabi tudi računalnike svojih sodelavcev ampak samo pod pogojem, da ji lahko tudi oni uporabljajo za izvajanje analiz

Peter svoj računalnik določi za "centralni manager"

Na preostale računalnike namesti HTCondor

Peter lahko sedaj izvaja sočasno več nalog na različnih računalnikih

# Petrov osebni HTCondor računski sistem



V Petrovi organizaciji kupijo namenske HTCondor strežnike

Za "centralni manager" se določi enega izmed namenskih strežnikov

Na namenske strežnike se namesti HTCondor

Peter in sodelovci lahko sedaj še bolj učinkovito uporabljajo HTCondor

# Soodvisne naloge

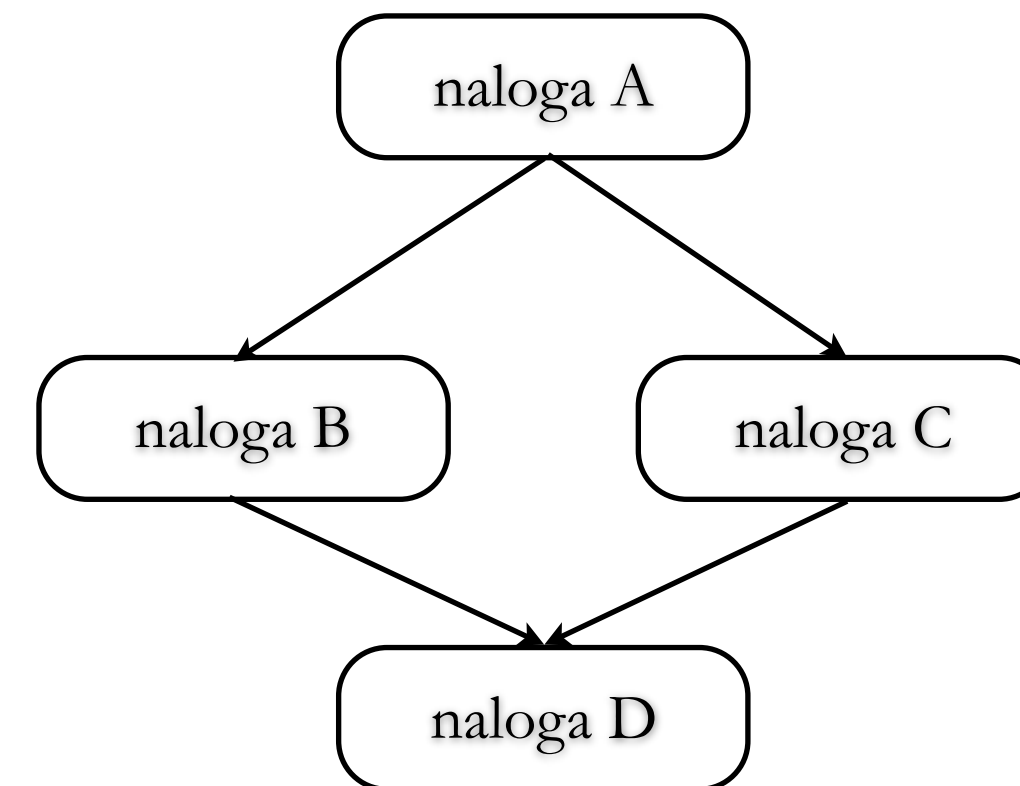
## Directed Acyclic Graph Manager (DAGMan)

podsystem 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
```

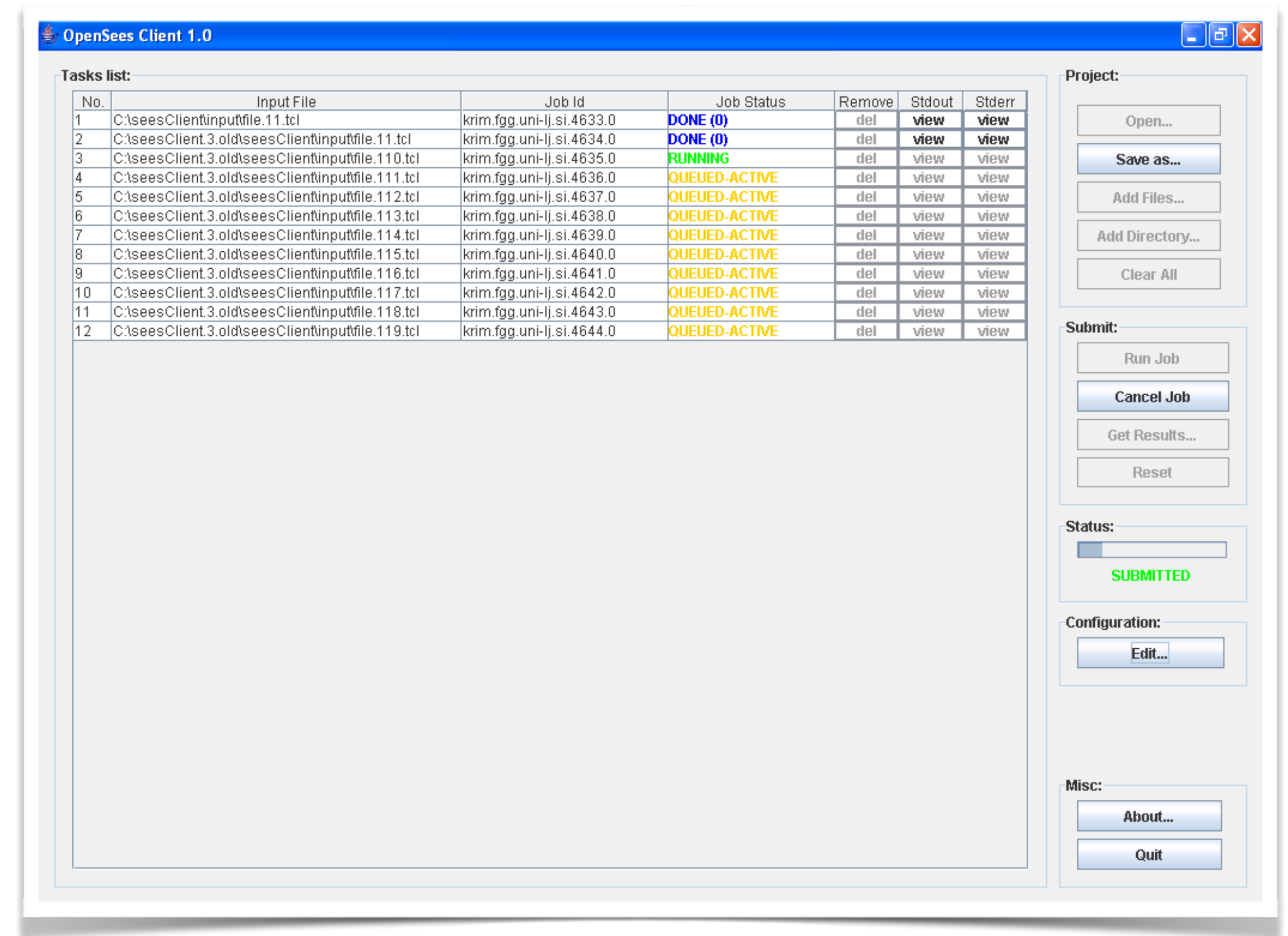


# Programske rešitve

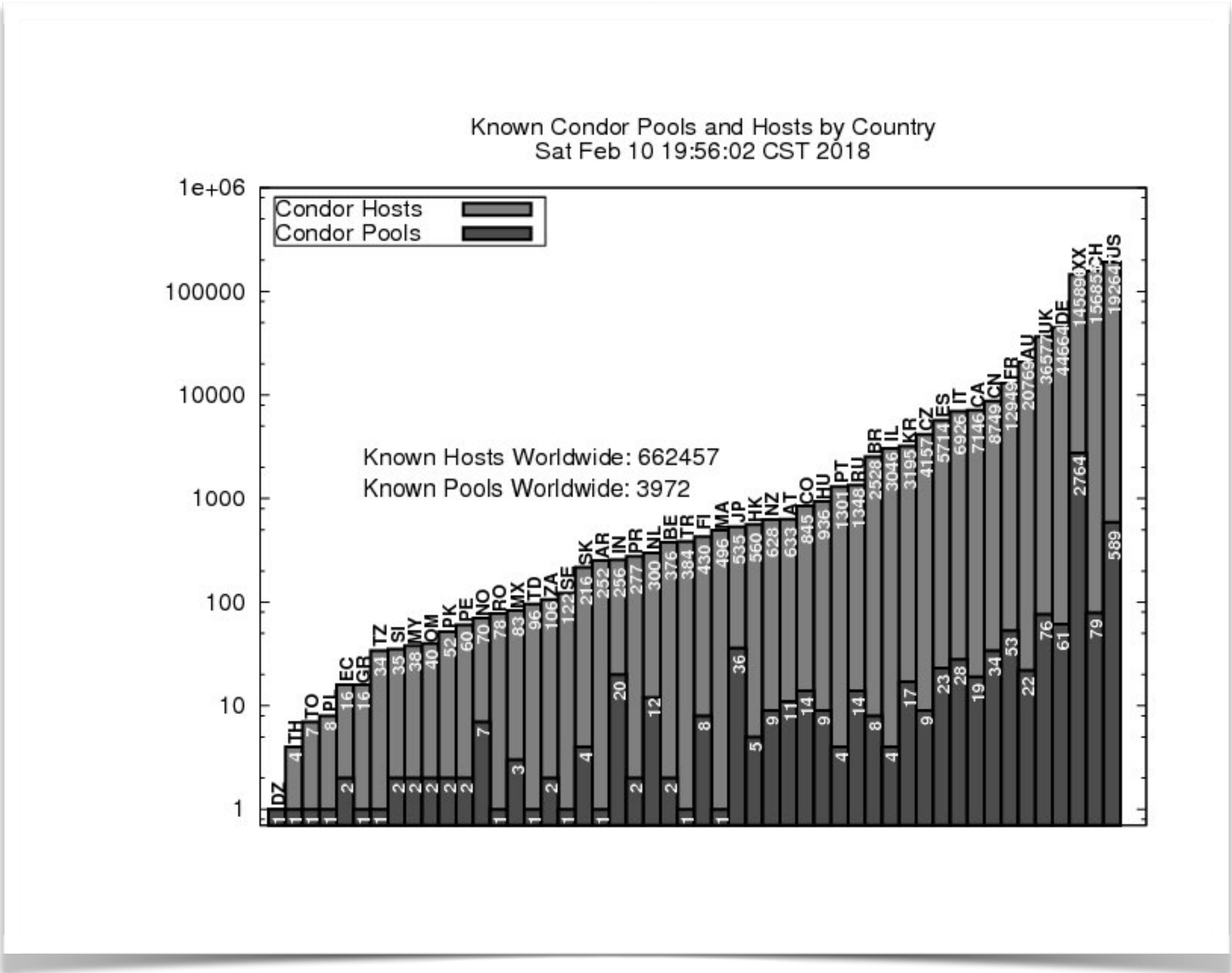
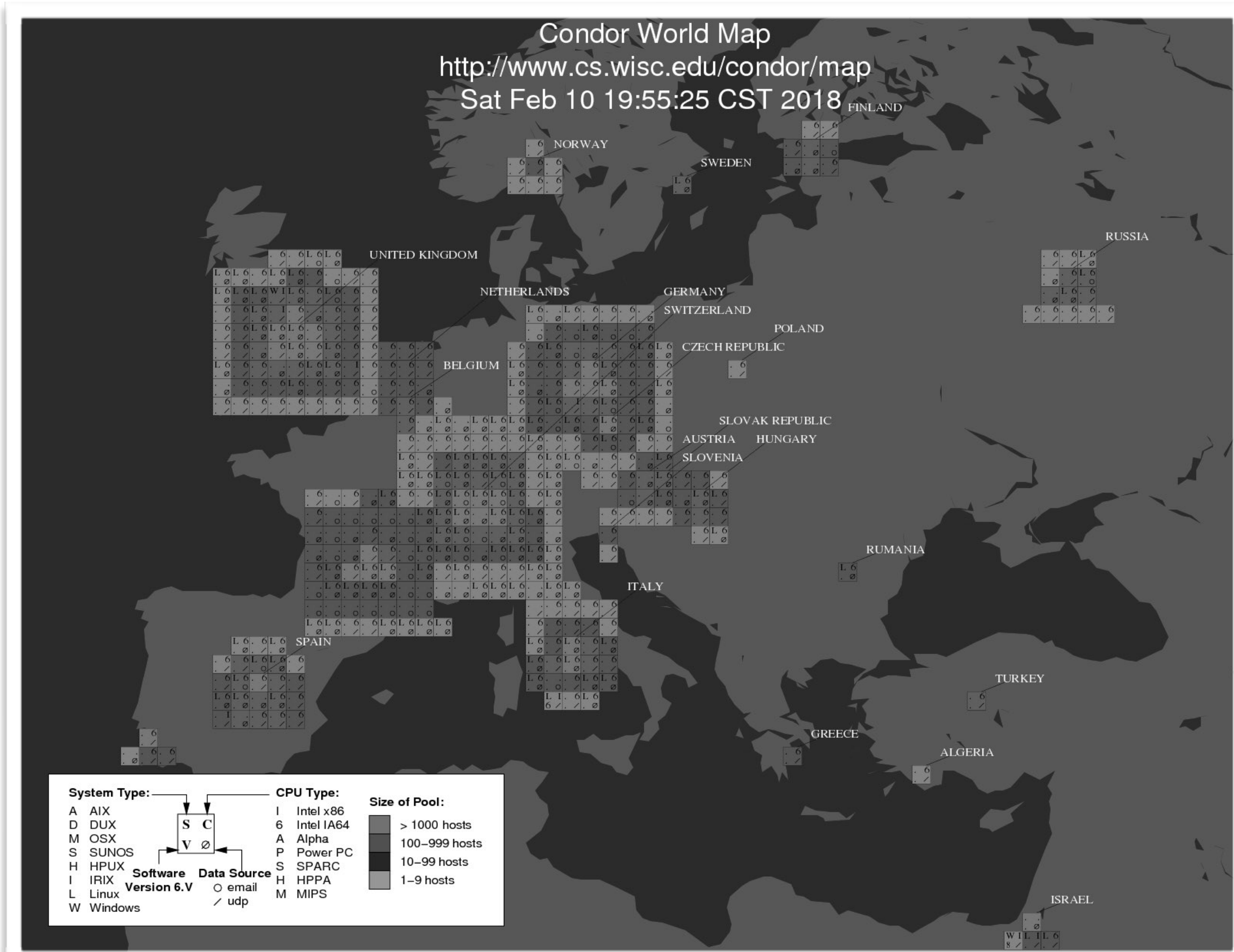
## Distributed Resource Management Application API (DRMAA)

Programski vmesnik za različne programske jezike  
(C/C++, Java, Python, Perl, Ruby)

OpenDSP: WS implementacija DRMAA API-ja



# Razširjenost HTCondor sistema





# Primer

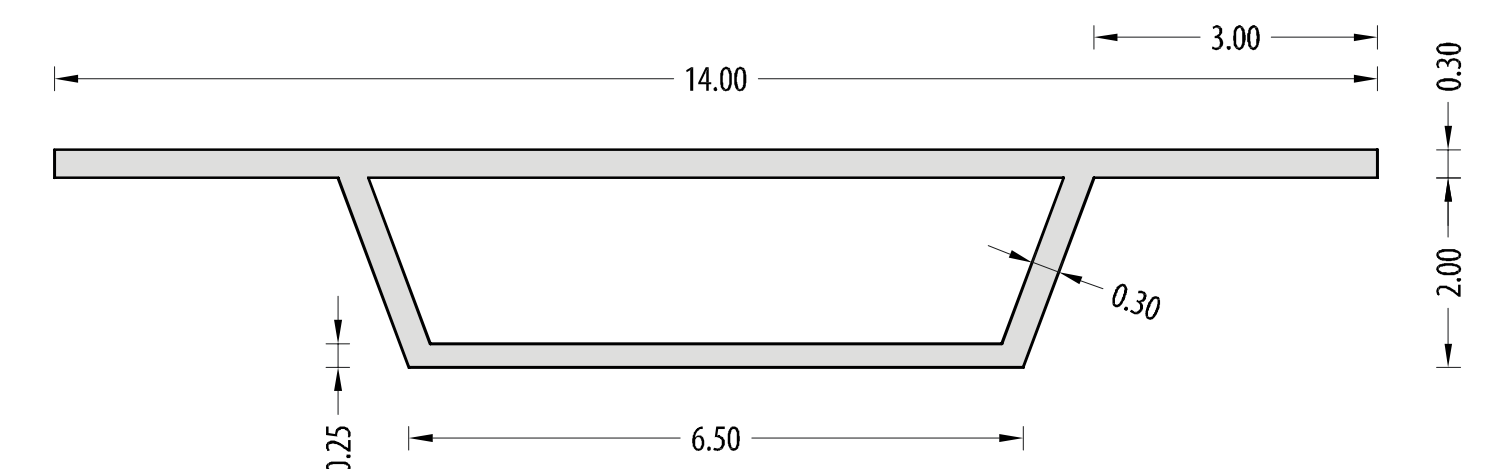
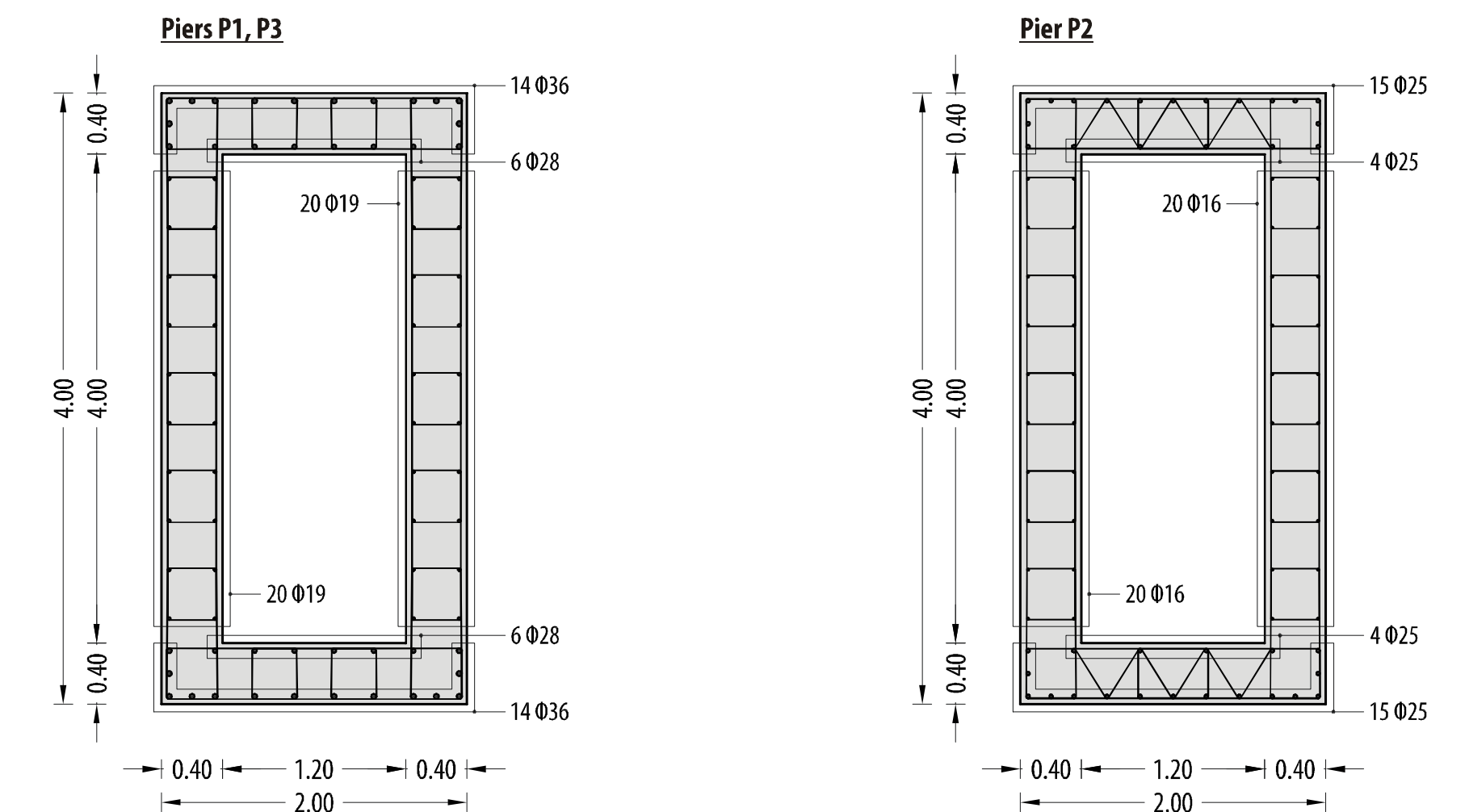
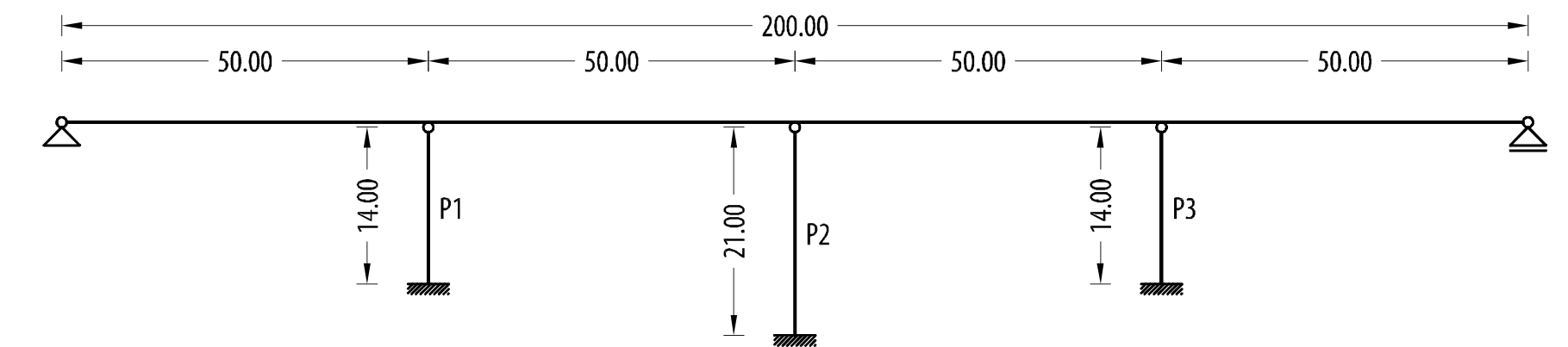
Visoko-propustno računsko okolje za analizo potresnega tveganja

IDA (Incremental Dynamic Analysis) analiza

parametrična analiza s katero za skupino akceleroگرامov in za različne stopnje potresne intenzitete določimo potresni odziv konstrukcije

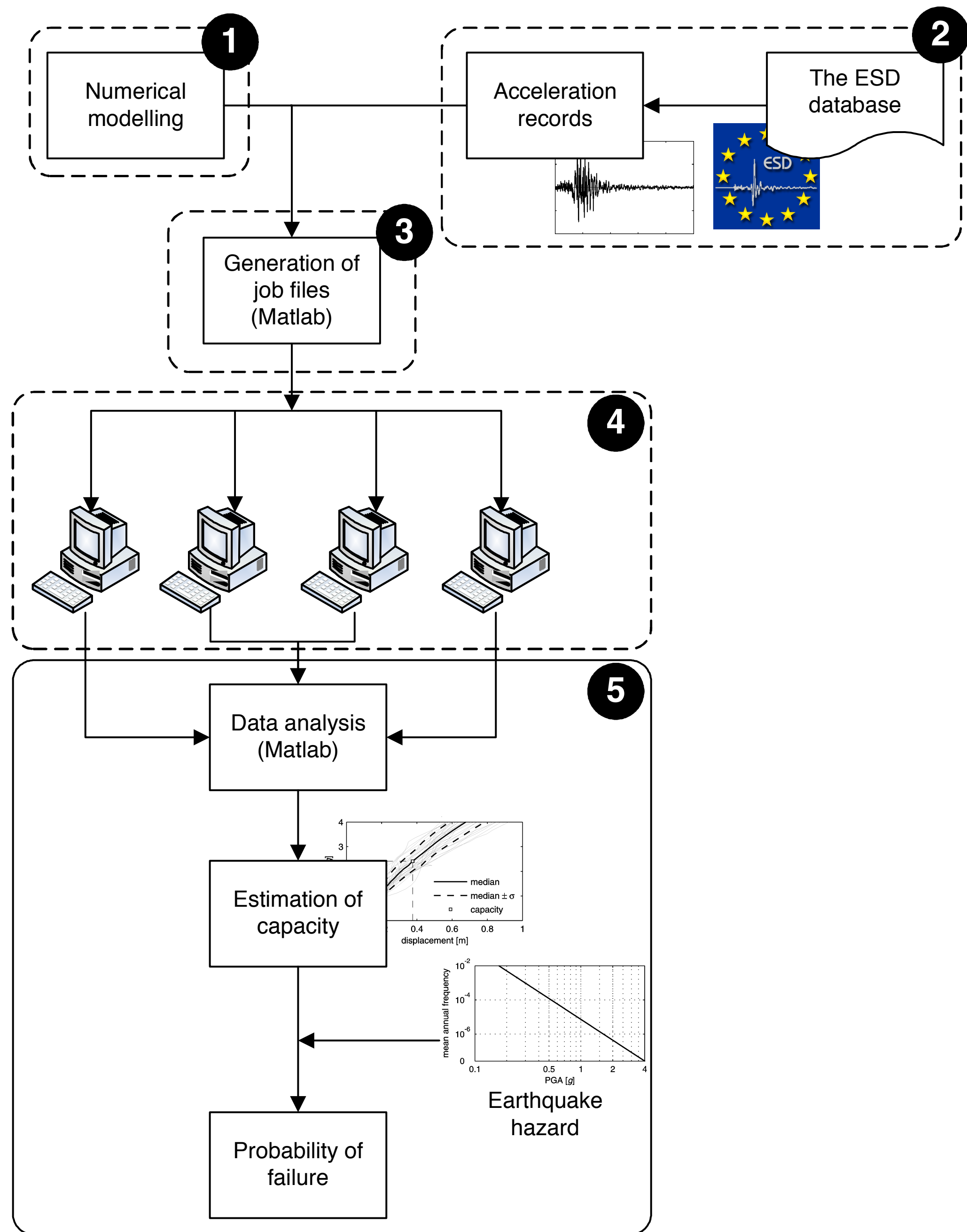
večja kot je skupina akceleroگرامov bolj zanesljivo se lahko določi potresno tveganje

proporcionalno z večanje skupine akceleroگرامov se povečujejo tudi računski časi



# IDA postopek

1. Numerično modeliranje
2. Izbira potresov
3. Priprava opisane naloge
4. Nelinearna analiza
5. Analiza rezultatov





# Ocena učinkovitosti 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  |