

# MATLAB PARALLEL SERVER



# Εισαγωγή

Το MATLAB Parallel Server (MPS) αποτελεί λογισμικό της MathWorks που μας επιτρέπει να κάνουμε scale τον MATLAB κώδικα σε clusters. Κάποια πλεονεκτήματα είναι:

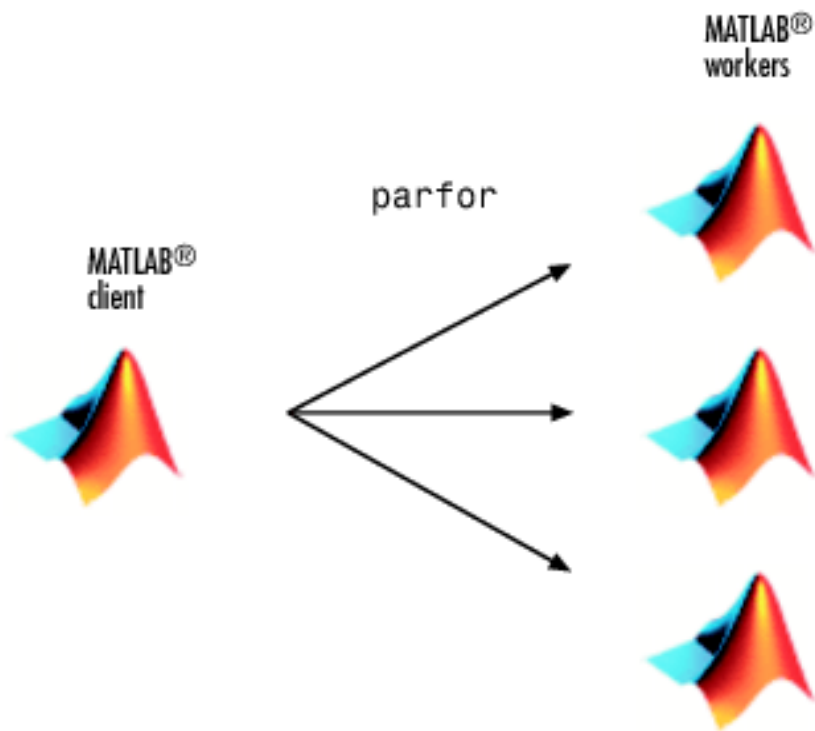
1. Εύκολη υποβολή και διαχείριση εργασιών στο cluster.
2. Αυτοματοποίηση ορισμένων σχετικά επίπονων εργασιών.
3. Δυνατότητα υποβολής από έναν υπολογιστή με MATLAB στο HPC cluster με πολύ απλά βήματα.
4. Με τη χρήση του Parallel toolbox καθίσταται ιδιαίτερα εύκολη η παραλληλοποίηση του κώδικα χωρίς να απαιτούνται πολλές γνώσεις πάνω στον παράλληλο προγραμματισμό.

# MATLAB Parallel Computing Toolbox

Για να μπορέσουμε να χρησιμοποιήσουμε τον MPS είναι απαραίτητη η εγκατάσταση του Parallel Computing Toolbox τοπικά στο Desktop μας. Επίσης θα πρέπει να εγκαταστήσουμε και μια σειρά από plugin scripts για Slurm τα οποία χρειάζονται ώστε να υποβάλλονται σωστά οι εργασίες μας στο cluster και να καθίσταται δυνατή η αμφίδρομη επικοινωνία του Matlab με το cluster.

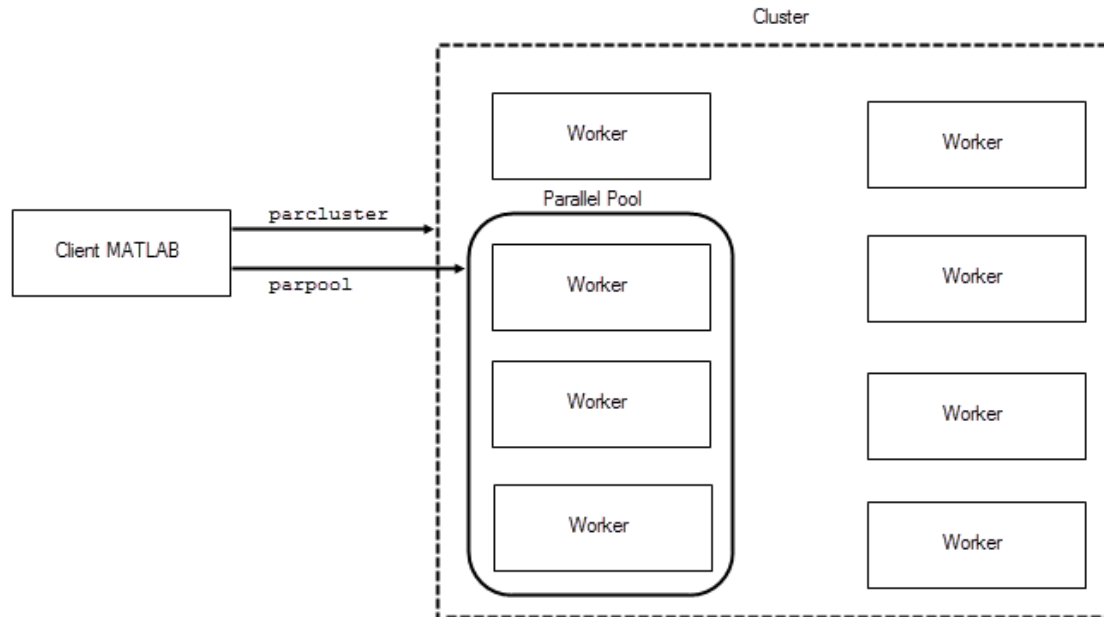
1. Parallel Computing Toolbox [<https://www.mathworks.com/help/parallel-computing/>]
2. Matlab Documentation <https://hpc.it.auth.gr/applications/matlab/>

# Βασικές εντολές (1/2)



- **parpool**: με αυτή την εντολή μπορούμε να δηλώσουμε πόσες cpu θέλουμε να δεσμεύσουμε, τις οποίες θα μπορούμε να χρησιμοποιούμε interactively. Η χρήση του περιορίζεται μόνο μέσω του ondemand partition. Οι περισσότερες εντολές ανοίγουν αυτόματα parpool.
- **parfor**: η εντολή αυτή μας δίνει τη δυνατότητα να τρέξουμε παράλληλα for loops, χρησιμοποιεί by default όσους workers βλέπει. Πρέπει να προσέξουμε να είναι ανεξάρτητες μεταξύ τους οι επαναλήψεις ώστε να μπορούν να τρέξουν παράλληλα.
- **spmd**: η χρήση του προτείνεται όταν θέλουμε να τρέξουμε εντολές οι οποίες απαιτούν επικοινωνία μεταξύ των workers.

# Βασικές εντολές (2/2)



- **distributed**: με την εντολή αυτή μπορούμε να δημιουργήσουμε ένα distributed array από ένα αντικείμενο datastore, οι workers λειτουργούν μόνο στο κομμάτι του array που τους αντιστοιχεί.
- **gcp**: η εντολή αυτή επιστρέφει το τρέχον parallel pool αντικείμενο.

Υπάρχουν πολλές ακόμα εντολές που έχουν διαφορετική χρήση, παραπάνω πληροφορίες μπορούμε να βρούμε <https://www.mathworks.com/help/parallel-computing/run-code-on-parallel-pools.html>.

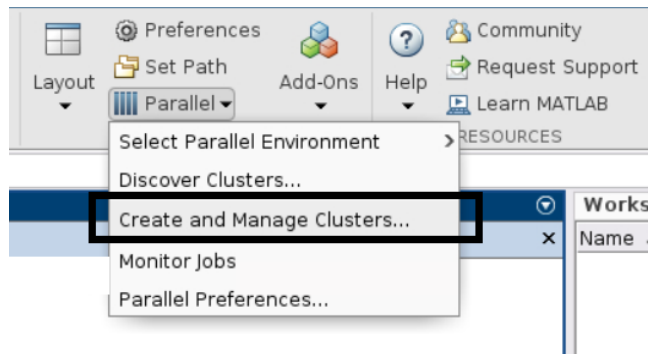
# Χρήση MPS μέσω του ondemand

Για να χρησιμοποιήσουμε το MPS μπορούμε να ανοίξουμε ένα interactive session στο ondemand Partition και από το περιβάλλον του να χρησιμοποιήσουμε το Matlab.

1. Πηγαίνουμε στην επιλογή Parallel



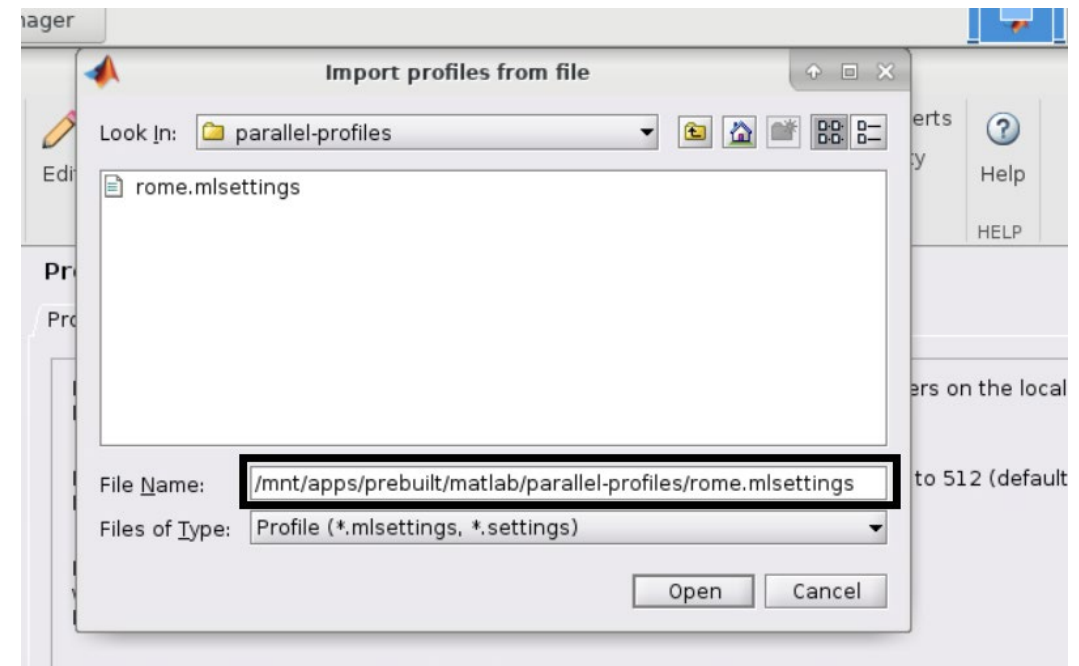
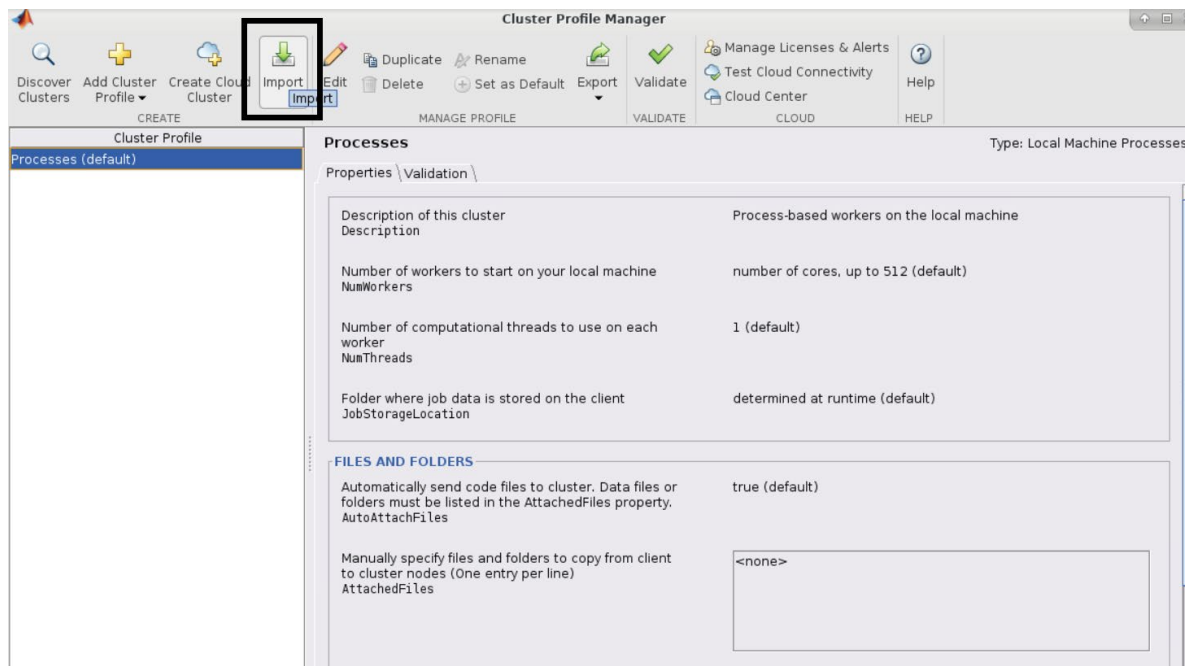
2. Επιλέγουμε Create Manage Clusters



# Χρήση MPS μέσω του ONDEMAND

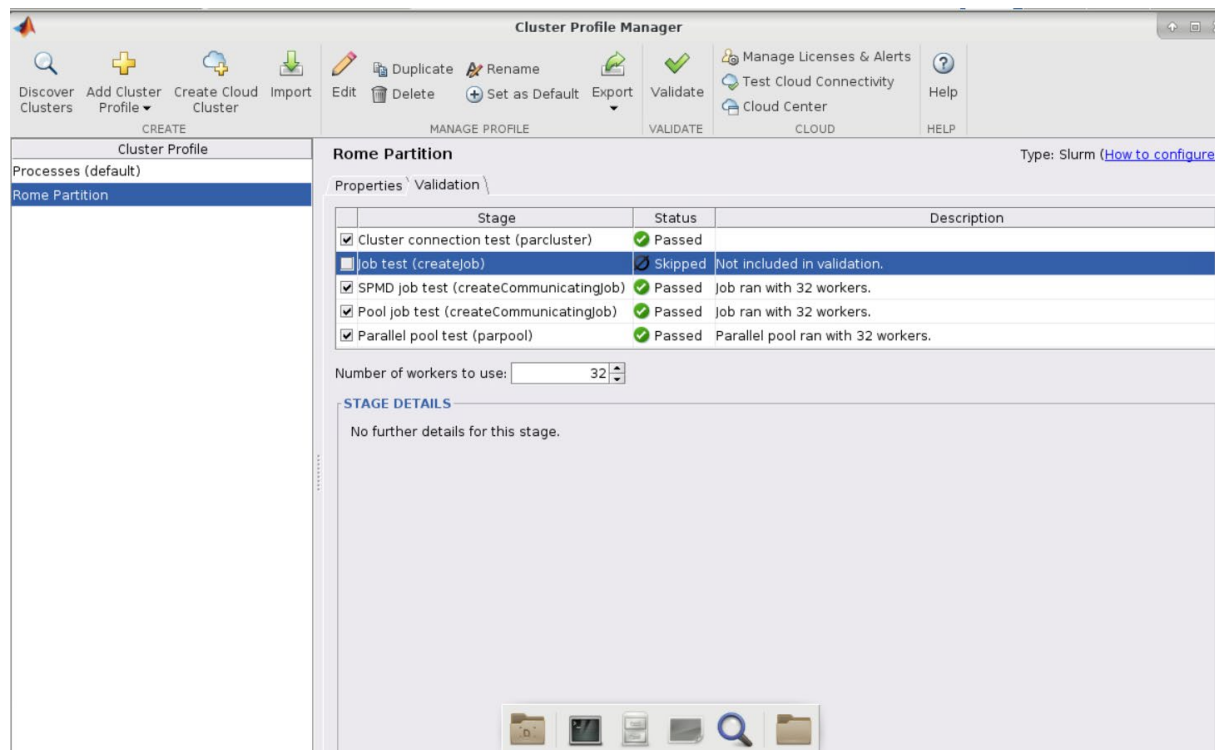
3. Μπορούμε είτε να δημιουργήσουμε το δικό μας cluster profile είτε να κάνουμε import κάποιο profile και να κάνουμε τις αλλαγές που θέλουμε, το οποίο και προτείνεται. Κάνουμε import από το path

`"/mnt/apps/prebuilt/matlab/parallel-profiles/rome.mlsettings"`



# Χρήση MPS μέσω του ONDEMAND

4. Στη συνέχεια μπορούμε να προβούμε σε οποιαδήποτε αλλαγή θέλουμε στο profile ώστε να πληροί τις απαιτήσεις του job που θέλουμε να τρέξουμε. Μπορούμε να κάνουμε validation στο profile επιλέγοντας τις κατάλληλες εντολές.



The screenshot shows the 'Cluster Profile Manager' interface. The main window displays the 'Rome Partition' configuration. The 'Validation' tab is active, showing a table of validation stages. The 'Number of workers to use' is set to 32. The 'STAGE DETAILS' section is currently empty.

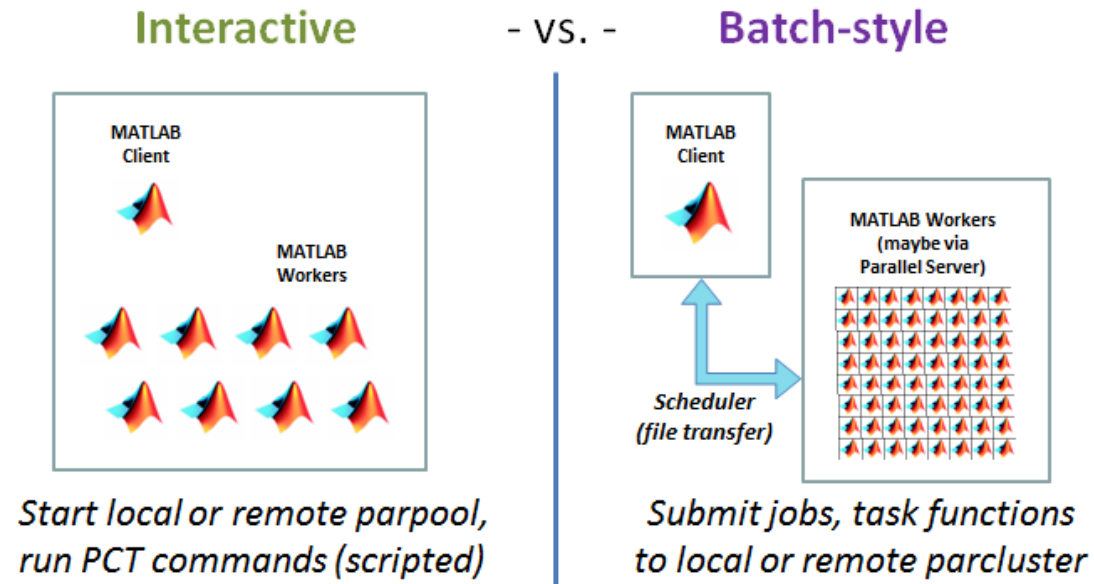
| Stage  | Status  | Description                        |
|--|---------|------------------------------------|
| <input checked="" type="checkbox"/> Cluster connection test (parcluster)   | Passed  |                                    |
| <input checked="" type="checkbox"/> job test (createJob)                   | Skipped | Not included in validation.        |
| <input checked="" type="checkbox"/> SPMD job test (createCommunicatingJob) | Passed  | Job ran with 32 workers.           |
| <input checked="" type="checkbox"/> Pool job test (createCommunicatingJob) | Passed  | Job ran with 32 workers.           |
| <input checked="" type="checkbox"/> Parallel pool test (parpool)           | Passed  | Parallel pool ran with 32 workers. |

Number of workers to use:

**STAGE DETAILS**  
No further details for this stage.

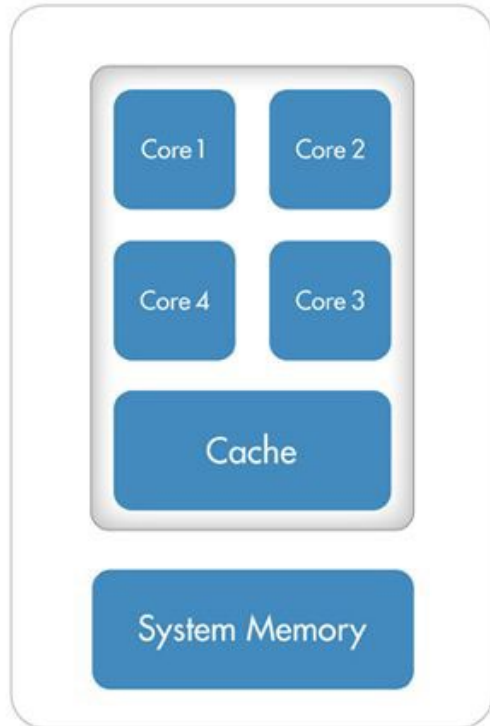


# Χρήση MPS μέσω του ONDEMAND

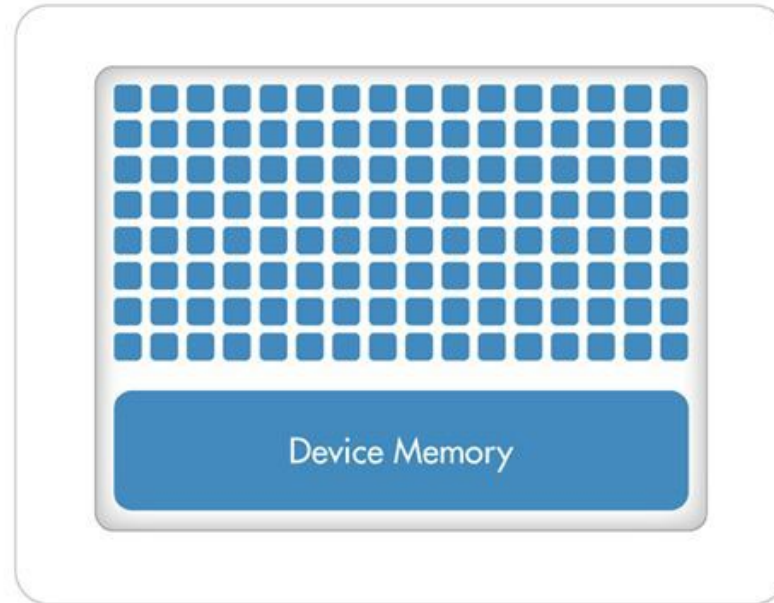


5. Ακολουθώντας αυτή τη διαδικασία μπορούμε να τρέξουμε εργασίες είτε μέσω της εντολής `batch` είτε ανοίγοντας `parpool`. Το `parpool` μας δίνει τη δυνατότητα να δεσμεύσουμε πυρήνες και να τους χρησιμοποιούμε *interactively* το οποίο όμως δεν προτείνεται για μεγάλο πλήθος πυρήνων.

### CPU (Multiple Cores)



### GPU (Hundreds of Cores)



# Χρήση GPU στο MPS

Στο Matlab μπορούμε εύκολα να χρησιμοποιήσουμε GPU για ορισμένους υπολογισμούς που θέλουμε να κάνουμε.

Δημιουργούμε `gpuArray` objects και τα χειριζόμαστε όπως και ένα απλό `array`. Το Matlab αυτόματα δεσμεύει τη GPU του συστήματος μας, χωρίς περαιτέρω απαιτούμενες ενέργειες

[<https://www.mathworks.com/company/newsletters/articles/gpu-programming-in-matlab.html>]

# Παράδειγμα Υποβολής Εργασίας

Με το script αυτό,  
δημιουργούμε cluster  
object για το partition  
ampere και υποβάλλουμε  
την εργασία  
simple\_gpu\_benchmark.

```
delete(gcp('nocreate'))  
c = parcluster("ampere");  
Create a job and submit it to the cluster.  
job = batch( ...  
c, ... % Cluster object created using parcluster  
@simple_gpu_benchmark, ... % Function or script to run  
1);  
wait(job);  
results = fetchOutputs(job)
```

# Επίλογος

Ο MPS καθιστά αρκετά εύκολη την υποβολή παράλληλων εργασιών στο cluster, όπου μας επιτρέπει να κάνουμε scale και σε περισσότερα

- Οδηγίες για τη χρήση του MPS στη συστοιχία μπορείτε να βρείτε

<https://hpc.it.auth.gr/applications/matlab/>

- Οδηγίες και παραδείγματα για το MPS μπορείτε να βρείτε και στο site της Mathworks

<https://www.mathworks.com/products/matlab-parallel-server.html>