



בעבודה זו עסקנו חישוב המשוואת ה- semi-empirical binding energy בעזרת הכלי CONDOR.

תחילה בהתאם להוראות שראינו בקובץ ה-PDF השתמשנו בפקודת status\_condor על מנת לראות שסביבת העבודה רצה כראוי וקיבלנו באופן זהה:

```
[hpc-user@hpc pp_4]$ condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@hpc.localdomain	LINUX	X86_64	Unclaimed	Idle	0.000	2533	0+00:00:00
slot2@hpc.localdomain	LINUX	X86_64	Unclaimed	Idle	0.000	2533	0+00:00:32

	Total	Owner	Claimed	Unclaimed	Matched	Preempting	Backfill	Drain
X86_64/LINUX	2	0	0	2	0	0	0	0
Total	2	0	0	2	0	0	0	0

בכדי לבצע מקבול לחישוב של המשוואה ה semi-empirical binding energy לכל הערכים האפשריים לווקטורים  $N = [1...200]$  ו-  $Z = [1...200]$ , החישוב חולק ל-200 חישובים לכל תהליך. מכאן כי לכל תהליך ערך  $N$  קבוע שאיתו מחשבים את המשוואה עבור על ערכי ה  $Z$  הנתונים.

בכדי להשתמש ב- CONDOR ישנו שימוש בקובץ מסוג submit שמריצים אותו בטרמינל עם הפקודה:

condor\_submit sim.submit

```
# Submit Description File:
# file name is sim.submit
# (Lines beginning with # are comments)
# NOTE: the words on the left side are not
# case sensitive, but file names are!

# 1 cluster 200 procces
universe = vanilla
executable = pp.py
Arguments = $(Process)
output = output$(Process).txt
log = sim.log #saves the life story of the jobs
InitialDir = run|
queue 200

#creating a subdirectory for all files
```

בקובץ מוגדר שיהיו 200 jobs המריצים את הקוד pp.py שבו הגרנו את החישוב של הפונקציה עבור ערך  $N$  קבוע בהתאם למספר הjob המתקבל כארגומנט.

על ידי הפקודה הבאה ניתן לראות את תהליך התור בזמן אמת:

condor\_q

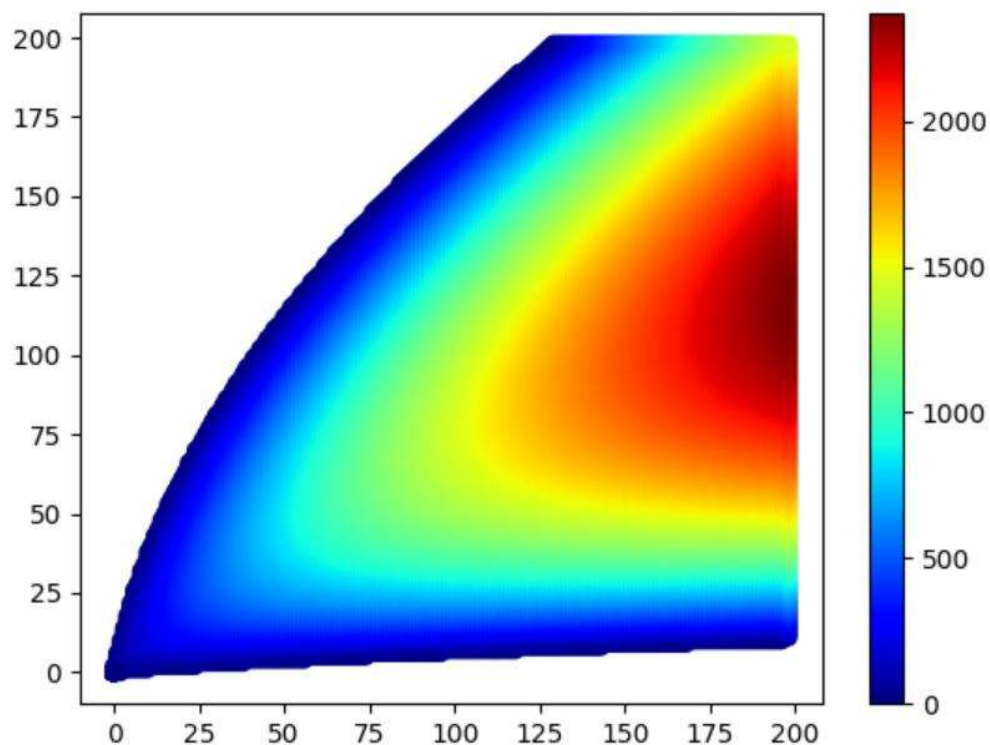
```
-- Schedd: hpc.localdomain : <10.0.2.15:9618?... @ 01/18/23 16:08:54
OWNER   BATCH NAME   SUBMITTED   DONE   RUN    IDLE  TOTAL JOB_IDS
hpc-user ID: 13      1/18 16:08    154     2     44    200 13.154-199

Total for query: 46 jobs; 0 completed, 0 removed, 44 idle, 2 running, 0 held, 0
suspended
Total for hpc-user: 46 jobs; 0 completed, 0 removed, 44 idle, 2 running, 0 held,
0 suspended
Total for all users: 46 jobs; 0 completed, 0 removed, 44 idle, 2 running, 0 held
, 0 suspended
```

ובכדי למחוק את ה jobs במקרה של terminate השתמשנו בפקודה הבאה:

```
condor_rm -all
```

לאחר קבלת ה 200 ערכים מכל job שנשמרו בקובץ טקסט בהתאם למספר ה- job בתוך התיקיה run כפי שהוגדר ב submit file, קבצים אלא נקראו על ידי תוכנית בpython. תוכנית זו לוקחת את הערכים שהתקבלו ובונה מטריצה מכל הערכים. לבסוף ערכים אלא הוסגו בעזרת הספרייה matplotlib, על ידי scatter plot שבו הצבעים של הנקודות מוגדרות בהתאם לערכי המטריצה, כך שמתקבל הגרף הבא:



## מסקנות כלליות מאופן העבודה עם CONDOR

- שימוש ב - condor מאפשר שימוש במשאבים זמניים pool condor על מנת לבצע תהליכים במקביל .
- ניתן להתקין על מגוון רחב של רשתות ומערכות.
- עמיד בפני תקלות.
- מאפשר לבחור מדיניות הפעלת תוכניות, כלומר ניתן תוכניות משפות שונות.
- שימוש ב - universe מסוג vanilla מאפשר ריצה כמעט של כל קוד סריאלי, מעביר I/O כמעט לכל מטלה.