For example, we have [94\_Part\_ZincDB\_.csv](https://drive.google.com/file/d/164Xrp7ZBFnCfrQM0EJx85LmAZaYNk5QQ/view?usp=drive_link) (~***500MB SMILE File***)

Steps Followed

1. Download the Entire pipeline from GitHub: [Parallel\_Prcessing\_Ligand\_Files.git](https://github.com/SahilMoreIIIT/Parallel_Prcessing_Ligand_Files.git) or Drive [Final Pipeline](https://drive.google.com/drive/folders/1yKQLt7kSoWxaVtWKXmDWowCyd5xtUyFW?usp=drive_link)
2. Now copy your smile.csv into this directory (ex: [94\_Part\_ZincDB\_.csv](https://drive.google.com/file/d/164Xrp7ZBFnCfrQM0EJx85LmAZaYNk5QQ/view?usp=drive_link)).
3. Server copy this directory on the server. (ex: Server allotted to me)

scp -r \* sahil23079@192.168.3.19:/home/ip\_arul/sahil23079

1. Run Script [\_Split\_Single\_File.sh](https://drive.google.com/file/d/1K1rtKF1Px4P2DNaeg9wy-lFupe608cCe/view?usp=drive_link) on this file, which takes CPU Core Counts and breaks the single file into counts Line by Line (So no data is corrupted)

chmod 777 \_Split\_Single\_File.sh

./\_Split\_Single\_File.sh

1. Run script [multiprocessing.py](https://drive.google.com/file/d/1IqWnp143hQWnYjneDl6C3bgo5WUJhibH/view?usp=drive_link) (already present in the directory) to parallelly convert those separately .smi files into separate .sdf formats.

python3 multiprocessing.py

nohup python3 multiprocessing.py & //TO RUN BackGround

1. Finally, merge the separate .sdf files[\_Merge\_Multiple\_Files.](https://drive.google.com/file/d/1m2HcY0tC08Ak73SX5gCEgRZ6IcHTOj1J/view?usp=drive_link) In the same order.

chmod 777 \_Merge\_Multiple\_Files

./\_Merge\_Multiple\_Files

1. Output: [94\_Part\_ZincDB\_.sdf](https://drive.google.com/file/d/1hgyCtcrF_Z7SSof9MLvHam71hOrLDVRj/view?usp=drive_link) (***30GB***)The Output File can be downloaded from Serve
2. *Server Download cmd (password required):*

scp -r sahil23079@192.168.3.19:/home/ip\_arul/sahil23079/94\_Part\_ZincDB\_.sdf