# Massively Scaling Molecular Screening Workloads on EuroHPC Supercomputers

***Sascha Hunold***[a], **Ioannis Vardas**[a], **Gökhan Ibis**[b], **and Thierry Langer**[b,c]

[a] *Research Group for Parallel Computing, TU Wien, Austria*
[b] *Inte:Ligand Software Development and Consulting GmbH, Austria*
[c] *Department of Pharmaceutical Sciences, University of Vienna, Austria*

**LigandScout** is an *advanced molecular modeling and design software suite*, which aims at enhancing early-stage drug discovery efficiency by providing in-silico experiments that help reducing the number of expensive in-vitro and in-vivo assays. An essential method of LigandScout is *virtual screening*, where pharmacophore models are used for identifying hits from large compound databases. Due to the computational requirements of this screening process, LigandScout already provides parallelization strategies, which were primarily *designed for elastic computations* on dedicated HPC or Cloud resources [1].

**Goal:** In this work, we consider the problem of *executing* an adapted, parallel version of *LigandScout* on a *dedicated supercomputer*, in particular, the Vienna Scientific Cluster 5 (VSC-5). Running LigandScout on a multi-user supercomputer is a challenging task, which requires a careful redesign of the parallelization workflow. We address these challenges and propose several solution strategies.

**Challenges:** A fundamental question is how to efficiently perform application-level load balancing. In dedicated HPC or Cloud environments, the task of load balancing can be offloaded to the batch scheduler (e.g., SLURM), i.e., the application sends computational tasks as single jobs to the batch scheduler, which takes care of executing these tasks whenever machines become available. This strategy is less suited for supercomputers such as the VSC-5, where batch jobs typically stay minutes or hours in the queue before being executed.

**Approach:** A possible solution to the load balancing problem at application level is to perform the scheduling within a batch job. To that end, we have ported our traditional parallelization approach to a job-level load balancer that was built on top of a distributed task queue (using the open-source Celery framework).

**Results:** The strong scaling results (fixed input) in Fig. 1 indicate that there is a sweet spot when selecting an effective number of compute nodes. The relatively low parallel efficiency mainly stems from the fact that the frequency of one core of the EPYC 7003 series processor, used in VSC-5, can be immensely boosted. Fig. 2 highlights that a good mix of threads and processes is needed to effectively use the large multi-core compute nodes of VSC-5.

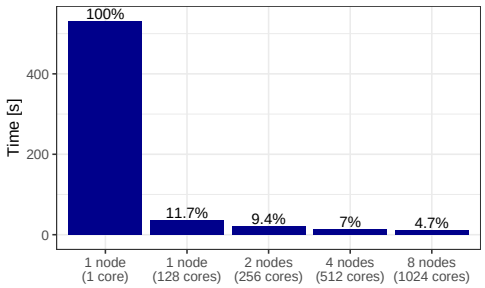


**Fig. 1:** Parallel runtime and efficiency (on top) of LigandScout for an increasing number of compute nodes (cores) on VSC-5.

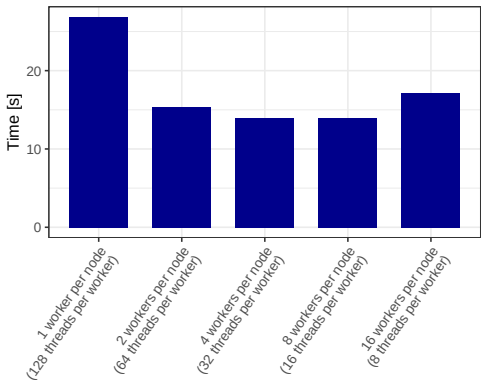


**Fig. 2:** Runtime of LigandScout when varying the number of workers and threads per worker. Notice that a total of 128 threads is always used per compute node.

**Acknowledgements:** This research was funded by the Austrian Research Promotion Agency (FFG).

## References

[1] Thomas Kainrad, Sascha Hunold, Thomas Seidel, Thierry Langer: LigandScout Remote: A New User-Friendly Interface for HPC and Cloud Resources. *J. Chem. Inf. Model.*, 59(1): 31–37, 2019.