Appendix: Artifact Description/Artifact Evaluation

Artifact Description (AD)

1 OVERVIEW OF CONTRIBUTIONS AND ARTIFACTS

1.1 Paper's Main Contributions

C_1 A novel energy grid alignment strategy enables parallel processing of multiple receptors by fully exploiting SIMD (Single Instruction Multiple Data) capabilities and maximizing DMA (Direct Memory Access) communication bandwidth. This significantly reduces the I/O and pose matching costs in ensemble docking while maintaining consistent results compared to separate single-receptor docking.

C_2 A novel trilinear SIMD interpolation algorithm and a performance portable layer to maintain the compatibility and efficiency of SWDOCK across multiple platforms beyond Sunway supercomputers.

C_3 A reorganization of the DB2 data structure for storing ligand conformations by sorting and merging the basic conformational components. This approach nearly reduces iteration times by 50% during the scoring phase of docking.

C_4 An "early bump" mechanism to identifies and dismisses clashing atoms, effectively eliminating unnecessary computations for conformations containing unfavorable high-energy components.

C_5 A cache scheme that stores the scores of commonly-used conformational components in the fast-access local data memory (LDM) to accelerate the scoring phase.

C_6 AthreadS framework is implemented, which is useful for porting Sunway-optimized programs to homogeneous platforms, bridging architectural gaps while preserving performance efficiency.

1.2 Computational Artifacts

A_1 https://github.com/hnlab/SWDOCKP2_example

Artifact ID	Contributions Supported	Related Paper Elements
A_1	All	Figure 3-6 12

2 ARTIFACT IDENTIFICATION

2.1 Computational Artifact A_1

Relation To Contributions

This artifact encompasses two x86 compilations of SWDOCKP^2, designed to support a maximum of 4 and 8 targets respectively for parallel processing. Accompanying these two compilations is a computational case study involving the docking of 296 db2 hierarchies against four conformations of the SARS-CoV-2 Main Protease.

To comprehensively showcase the performance of the eight-target compilation, we replicated the target-related files as its input. These compilations are based on a specific version of the source code that incorporates all available code-level optimization strategies.

Two variants of the ligand database are supplied: the unprocessed (raw) version and the optimized version, which has undergone conformation sorting and merging as is mentioned in the paper. This artifact effectively demonstrates the enhanced efficiency and performance portability of SWDOCKP^2.

Expected Results

Both compilations are expected to exhibit faster execution on the optimized db2 database compared to the raw version. More precisely, the four-target compilation is anticipated to achieve a marginally higher processing speed and significantly lower memory consumption when contrasted with the eight-target compilation.

Expected Reproduction Time (in Minutes)

On an AMD EPYC 9654 processor, with a single process utilizing 16 threads, the four-target compilation required 0.08 minutes and 0.11 minutes to complete docking the optimized database and the raw database, respectively. In comparison, the eight-target compilation took 0.12 minutes and 0.14 minutes for the same tasks, respectively.

Artifact Setup (incl. Inputs)

Hardware. A CPU with x86 architecture that supports multi-threading is necessary for execution.

Software. The system environment must have glibc version 2.28 or higher, and mpich version 4.2.0 or higher to successfully run this artifact.

Datasets / Inputs. The provided ligand .db2 datasets are originally sourced from ZINC library [1, 2]. Subsequently, they are transposed and compressed into the .xz format, as detailed in reference [3]. The optimized version of the datasets is created through the process of sorting and merging conformations, as described in the accompanying paper. For proper execution, the working directory should contain an input parameter file named INDOCK and the paths to all input ligand .db2 files. For more comprehensive details regarding the input structure and file requirements, please refer to the GitHub repository mentioned in section 1.2. All relevant files are included within this repository.

Installation and Deployment. The GitHub repository contains all the necessary non - glibc dynamic libraries. The linkage paths for these libraries have been pre-configured via the LD_LIBRARY_PATH environment variable within the provided execution scripts.

Artifact Execution

Users can directly execute the provided scripts to run the artifact without installation.

Artifact Analysis (incl. Outputs)

The artifact will generate the following output files:

- *x*OUTDOCK: Docking scores generated by process *x*.
- *x*mpro_rec_*y*.mol2.gz: Gzip-compressed docking poses in .mol2 format for target *y*, produced by process *x*. The order of these poses corresponds to that in *x*OUTDOCK.

The processing time is computed by subtracting the start time from the end time, where both times are retrieved using the *gettimeofday()* function from the *<sys/time.h>* library. The resulting time value, expressed in seconds, is then outputted to both the *x*OUTDOCK file and the standard output (stdout).

Upon the utilization of multiple MPI processes, the time expenditure of each process may differ due to the task allocation strategy implemented in SWDOCKP[2]. The specific wall time for process x is presented at the end of the *x*OUTDOCK file, succeeded by the total elapsed time for all processes to termicate.

Analogously, the durations for ligand loading, minimization operations, and the entire search procedure across all processes are calculated. These time metrics are printed to stdout in the sequence corresponding to the termination order of the processes.

REFERENCES

[1] J. J. Irwin *et al.*, "ZINC: a free tool to discover chemistry for biology," *J. Chem. Inf. Model.*, vol. 52, no. 7, pp. 1757–1768, Jul. 2012.

[2] T. Sterling and J. J. Irwin, "ZINC 15–ligand discovery for everyone," *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015.

[3] K. Xu *et al.*, "Redesigning and optimizing ucsf dock3.7 on sunway taihulight," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, p. 4458–4471, Dec. 2022. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2022.3194916