Efficient and Portable MPI Support for Approximate Bayesian Computation

Lorenzo Fabbri[a,†], Avinash Ummadisingu[a,†], Ritabrata Dutta[b], Radim Janalik[a], Antonietta Mira[b], Olaf Schenk[a], Marcel Schoengens[c]

[a] Institute of Computational Science, Faculty of Informatics, Università della Svizzera italiana,
[b] InterDisciplinary Institute of Data Science, Università della Svizzera italiana,
[c] Swiss National Supercomputing Centre,
[†] These authors contributed equally to this work

Approximate Bayesian Computation

ABC

ABC. Approximate Bayesian Computation (ABC) refers to a class of methods used in statistical inference the aim of which is to obtain the posterior distribution $P(\theta|D)$, with θ being the model's parameters and D the observed dataset, for models whose likelihood function is intractable. In its most simple version, a parameter θ_n is sampled from the prior distribution $P(\theta)$ and a dataset $D(\theta_n)$ is simulated using a statistical model M. The sampled parameter is accepted or rejected based on the value of a distance measure $\rho(D, D(\theta^*))$. The iteration of the sampling process results in a posterior distribution of the parameters θ.
Figure 1 shows the workflow for an ABC method based on the Rejection algorithm.

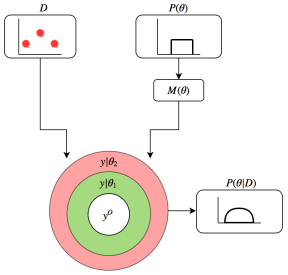


Figure 1: Basic workflow for ABC methods.

ABCpy

ABCpy. ABCpy is a modular, scientific library for ABC written in Python. The design of the library follows ABC methods' required components, which are implemented in abstract classes, and it allows domain scientists to integrate their own models into the framework. The library is currently parallelized using the map-reduce model and uses Apache Spark as a backend.

Contribution. The contribution of this project is the development of a parallel backend for ABCpy based on MPI. The analysis of scaling properties on the CSCS supercomputer Piz Daint is also provided.

MPI Backend

MPI Backend

Motivation. Depending on the model complexity, ABC has usually high computational resource requirements. HPC centers are often available at convenience to academic researchers. Since MPI is a *de facto* standard in such facilities, an MPI backend improves easy of use for such users.
Figure 2 shows an extract of the most important classes and functions within the MPI backend.

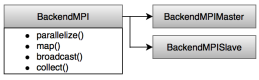


Figure 2: Class diagram.

Performance Evaluation

Performance evaluation of the MPI backend has been conducted on the CSCS supercomputer Piz Daint using the multi-core nodes, each equipped with two Intel Broadwell processors with 36 cores and 128G RAM in total.

Figure 3: CSCS supercomputer Piz Daint.

Methodology

One master node has been used, while scaling the worker nodes from 2 to 32 in powers of two. The study of scalability has been performed measuring two quantities:

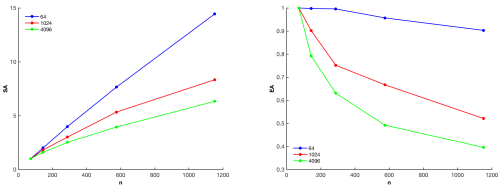
- **Speedup** The ratio of an algorithm running time on m cores (baseline, $m = 72$) and the running time on n cores;
- **Efficiency** The ratio of the observed and expected speedup.

Two sampling schemes, PMCABC and SABC, and the non-linear dynamics model of Lorenz have been used, with the following parameters: **Prior distribution** Uniform, **Perturbation kernel** Multivariate t-distribution, **Samples from the posterior distribution** 10000, **Distance function** Euclidean distance, **Summary statistics** `HakkarainenLorenz`.

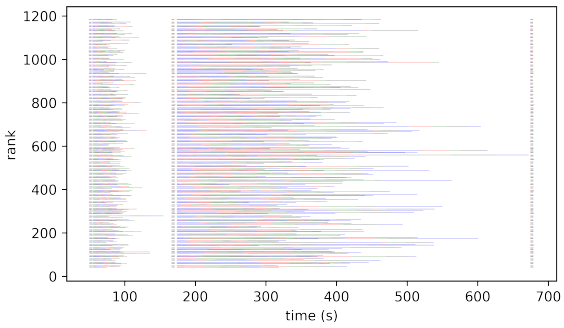
Results & Discussion

PMCABC algorithm

Results. Speedup (left), $S_A(n)$, and efficiency (right), $E_A(n)$, of PMCABC algorithm for Lorenz model parallelized on different numbers of cores n, from 72 to 1152 in powers of two, and for different values of NTT (number of time-steps between $[0, 4]$), 64, 1024 and 4096.

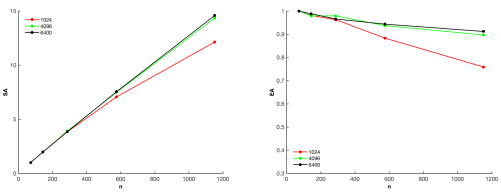


Discussion. Wall times in seconds taken to execute `map()` instructions for all the ranks on a data item for 32 nodes and $NTT = 1024$.

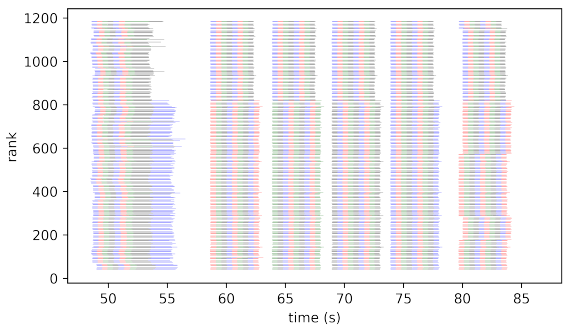


SABC algorithm

Results. Speedup (left), $S_A(n)$, and efficiency (right), $E_A(n)$, of SABC algorithm for Lorenz model parallelized on different numbers of cores n, from 72 to 1152 in powers of two, and for different values of NTT (number of time-steps between $[0, 4]$), 1024, 4096 and 6400.



Discussion. Wall times in seconds taken to execute `map()` instructions for all the ranks on a data item for 32 nodes and $NTT = 1024$.



Conclusion

- The MPI backend shows near linear scaling for small values of NTT for both PMCABC and SABC algorithms
- A performance degrade is observed for some inference schemes such as PMCABC due to a computational load imbalance
- Future work will investigate the implementation of a load balance system through dynamic allocation of data chunks to worker ranks

The presented project is part of the Master course *Software Atelier: Simulation, Data Science and Supercomputing* given by Prof. Dr. Olaf Schenk at USI.

References

[1] Ritabrata Dutta, Marcel Schoengens, Jukka-Pekka Onnela, and Antonietta Mira. *ABCpy: A user-friendly, extensible, and parallel library for Approximate Bayesian Computation.* Proceedings of The Platform for Advanced Scientific Computing Conference, Lugano, Switzerland, June 2017 (PASC2017).

[2] Jarno Lintusaari, Michael U. Gutmann, Ritabrata Dutta, Samuel Kaski and Jukka Corander. *Fundamentals and Recent Developments in Approximate Bayesian Computation.* Systematic Biology, September 2016.