

Optimized Galaxy Bayesian Surface Photometry on a Many-Core Platform

Igor Kolesnikov¹, Celso Mendes¹, Reinaldo de Carvalho², Reinaldo Rosa¹

¹Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brazil

²Universidade Cruzeiro do Sul, São Paulo, SP, Brazil

igor.s.kolesnikov@gmail.com, celso.mendes@inpe.br

rrdecarvalho2008@gmail.com, reinaldo.rosa@inpe.br

Abstract. *Parametric computational modeling of galaxies is a process with a high computational cost. The statistical component of modeling, which may involve model refinements in relation to the source brightness distribution, achieves more satisfactory results when the Bayesian approach is employed. In our research, we use GALaxy PHotometric ATtributes (GALPHAT) as our primary tool for data processing. In the current scenario of cosmology, to be scientifically relevant, this type of modeling must be performed on thousands of galaxies. In this article, we present the study and optimization of solutions based on modern HPC platforms, including a many-core processor, that enable effective processing of that amount of galaxies obtained from Sloan Digital Sky Survey.*

1. Introduction

The parametric computational modeling of galaxies is a process with a high computational cost that may last years of CPU time when several thousands of galaxies have to be processed and especially when the Bayesian approach is used. In a typical scenario currently, this type of modeling can be performed only on a limited set of galaxies, hampering more ambitious and complete astrophysical data analysis. In recent years, the characterization of galactic components such as bulge, disk, central point sources, and others of smaller mass contribution has been shown an essential diagnosis of physical processes that determine their evolution. In particular, they depend on the environment where the galaxy resides.

Many relevant questions still remain open. How does a galaxy form its stars? What determines the total star content of a galaxy? Astronomers have devoted considerable effort to answer such questions in recent decades. Stellar formation starting from a cold gas cloud is an extremely complex problem and one of the most difficult of modern astrophysics. The problem becomes more complicated by the fact that many galaxies reside in larger structures, such as groups and clusters of galaxies, where they interact with neighboring systems. Thus, it becomes imperative to understand separately how internal processes are distinguished from external processes that determine the stellar content of a given galaxy in a certain environment and for a specific redshift. In summary, the results of this research will improve our understanding of these effects by bringing an analysis done with Bayesian inference tools, whereas previously only frequentist tools were used.

This research aims to contribute to the understanding of this matter by processing a Spheroids Panchromatic Investigation in Different Environmental Regions (SPI-DER) [La Barbera et al. 2010] data sample with a Bayesian-based tool. This article presents initial steps of an ongoing, long-term research, which can be divided in three parts: (1) test and optimization of environment; (2) data processing using the Sérsic model [Sérsic 1963]; (3) data processing using the Sérsic+Exponential model (S+E) [Andredakis and Sanders 1994] and consequential analysis. The main contribution of this article is to discuss the first and second parts, i.e., available hardware and software solutions, its usage and the results of processing a portion of our data. This processing is done with GALPHAT¹ [Weinberg et al. 2013], which is a front-end application of the Bayesian Inference Engine (BIE), a parallel Markov Chain Monte Carlo (MCMC) package.

The BIE package uses standard MPI and POSIX threads and, therefore, it can run in a broad spectrum of parallel or scalar environments and can be easily ported to high-performance hardware for production analysis. The basic functioning of GALPHAT consists of spawning a given number of MCMC's instances using a Metropolis–Hastings algorithm [Chib and Greenberg 1995], which will generate and check the values of each parameter and store them in a file at the end. The number of MCMC's instances can vary depending on the number of available threads and used models [Yoon et al. 2011]. After a set of tests, we concluded that 20 MCMC's would be optimal for the Sersic model and 72 MCMC's for the Sersic+Exponential. Exceeding the maximum limits, it is possible to cause the appearance of varied errors due to the saturation of the active chains. From experimental findings, we saw that one cannot keep increasing MCMC's to theoretically increase the performance, because there is an upper limit after which performance deteriorates or processing stalls. This is the main reason why GALPHAT needs another layer of submission and distribution of galaxies, as discussed here.

The remainder of this article is organized as follows. In Section 2, we discuss our main goal and challenges, along with the dataset used in this research. Section 3 describes tests to understand the differences in available hardware and software options, following the methods that were applied to improve performance. In Section 4, we summarize the results and experimental achievements of the work. Section 5 lists examples of related work and how they compare to our research. Finally, Section 6 concludes the work by responding to the initial questions and pointing to future planned developments.

2. Main Goal and Challenges

In this research, our focus is to vastly extend a previous preliminary study [Stalder et al. 2017], processing over 40 times more data. Our first goal is to process the data with the Sérsic model and then, at a later stage, apply the much more complex and sophisticated (and more costly in a computational sense) Sersic+Exponential model to the same data. Our sample consists of 39,993 Early-Type Galaxies (ETG) taken from the Sloan Digital Sky Survey (SDSS) Data Release 7 (DR7) [Abazajian et al. 2009]. The total number of galaxies in this catalog is approximately 1.12 million. Given the Big Data context of large databases like SDSS, used in this work, or LSST, to be potentially used in future research, it is necessary to explore software and hardware solutions to increase the effectiveness of GALPHAT.

¹<https://bitbucket.org/mdweinberg/bie.git>

The key objective of processing the data with two models (namely, Sérsic and Sérsic+Exponential) is the fact that, with GALPHAT, we can distinguish which one is the most cost-effective modeling method through the Bayes Factor analysis [Kass and Raftery 1995]. Following the completion of processing, we will study the main scaling relationships of these systems. On one hand, Sérsic (employed in this article) produces acceptable results, while, on the other hand, Sérsic+Exponential (to be employed in the near future) provides more refined results but demands significantly longer processing times. Thus, the final product, consisting of a compilation and analysis of the massive amount of galaxies to be processed using the Bayesian Inference tool, with corresponding processing resources required, is of considerable impact. This kind of study can bring a good amount of new information and insights that were unavailable to researchers who previously used frequentist approaches on the same data.

3. Tests for Configuration Exploration

Here we present initial explorations that we conducted to search for the most appropriate tools and platforms that should be used for the processing of GALPHAT. We summarize our findings and point to the final configuration in the next section. Here, we use previously obtained results [Stalder et al. 2017] as a basis for comparison. Like in that previous work, most of our tests use 20 threads to process each galaxy in GALPHAT.

3.1. Hardware Platforms

At our disposal for the processing of the data, we have a local machine (hostname Pollux) based on the Intel Xeon Phi processor, model 7250, with 68 cores (272 threads), at a clock speed of 1.4 GHz (1.6 GHz in TurboBoost), and 192 GB of RAM. This machine is full-time dedicated to this research activity and is used mainly for tests and processing of the Sérsic model. Also, we have access to a dedicated partition on the Santos Dumont supercomputer consisting of 33 Thin nodes² (each with two processors Intel Xeon E5-2695v2 Ivy Bridge, 2.4 GHz and 64 GB of RAM). That machine is fully dedicated to the processing of the Sérsic+Exponential model and related tests. We also used two Linux systems, based on the Intel i5 and AMD Ryzen processors, for pre/post-processing tests.

3.2. Compiler Comparison

Compiler choice was the first factor we addressed with the preliminary tests, to verify two things: (i) influence of compilers on GALPHAT performance and (ii) scaling efficiency if engaging maximum number of threads on our local machine. The compilers available on the machine based on the Intel Xeon Phi were GNU's GCC and Intel's ICC. For an initial evaluation, we used the NAS benchmark³ with the EP (Embarrassingly Parallel) kernel and C class, obtaining the results in Table 1.

The first important result observed is that ICC is superior to GCC with both numbers of threads. In addition, with the usage of a maximum number of threads, the gain obtained with the ICC compiler was even more prominent, meaning that when we are using near the maximum of the machine for galaxy processing, we can obtain even more performance with ICC. As another important detail, Table 1 shows promising capabilities

²<https://sdumont.lncc.br/machine.php?pg=machine>

³<https://www.nas.nasa.gov/publications/npb.html>

of the Xeon Phi processor in embarrassingly parallel tasks, as the speedup obtained from thread parallelism with both compilers was a significant fraction of the increase in the number of threads employed.

NAS benchmark times with 20 and 272 threads.			
	20 threads	272 threads	Parallelism Speedup
GCC	58.01 s	07.73 s	7.50
ICC	30.36 s	03.64 s	8.34
Compiler Gain	1.91	2.12	

Table 1. Benchmark results using 20 and 272 threads on Xeon Phi machine to compare the performance and efficiency of GCC and ICC compilers. As load task we used NAS benchmark with EP (Embarrassingly Parallel) Kernel and C class. The results were obtained by running each test three times and getting mean values.

Parallelism efficiency would be 100% (ideal case) if we got speedup proportional to the increase of active threads. In this case, we had 20 threads in the first test and 272 threads (total number of available threads) in the second. This represents an increase of 13.6 times in the number of threads. If we had obtained the same speedup, we would be able to say that we can use all available threads without losing any performance. Nevertheless, due to the fact that the Xeon Phi has only 68 real cores, each capable of running four simultaneous threads, that is not possible. In our case, we start to lose performance while using $\sim 55\%$ (for the GCC compiler) and $\sim 61\%$ (for the ICC compiler) of the total number of threads. While we do not get the ideal case of efficiency when employing all threads, we are able to use more than half of the machine for our processing. Later in the article, we will comment more on this matter in the context of GALPHAT.

After thoroughly preparing the environment and main packages to ensure correctness of results, we selected a sample of 15 random-sized galaxies to verify the dependence of GALPHAT’s performance on the compiler. The results of this run on the Pollux system using GCC and ICC compilers clearly showed, again, that we obtain more performance by using ICC, which justifies its adoption for further experiments. Using the 15 galaxies from our list as testing input, we had a mean gain of 1.35 for ICC over GCC, which we can convert to an economy of more than two months when processing our full collection of galaxies.

3.3. Processor Comparison

Our next task was to compare the performance between our local machine’s Xeon Phi processor and the processor employed in previous studies [Stalder et al. 2017], which was an Intel Xeon E5-2660 processor. According to our preliminary tests [Kolesnikov 2020], we estimated that the difference between the two processors (Xeon Phi and Xeon E5 2660) would be a factor of about ~ 2.39 times (namely, 71 minutes vs. 30 minutes in a random galaxy of size 132 pixels) in favor of the Xeon E5-2660, all this while using GCC on both machines with the same galaxy. On the other hand, using the same setups and input galaxy, but changing the compiler to ICC on the Xeon Phi, we obtained the results where we can see a significant decrease in the processing time for the Xeon Phi and consequent change of the Xeon E5-2660 gain, from 2.39 to only 1.69. This shows, once more, a clear

argument for the importance of using the Intel C compiler on the Xeon Phi processor when running GALPHAT.

To obtain a more comprehensive comparison between the two processors, we considered a subset of the SPIDER sample containing 1,200 galaxies, the same subset that had been used in a previous study [Stalder et al. 2017]. We processed each of those galaxies with the Xeon Phi processor, and obtained the data in Figure 1, which also shows the previous results with the Xeon E5-2660 processor. In both cases, 20 threads were used to process each galaxy with GALPHAT. Extrapolating these results, we would get bleak predictions on how long it would take to process our entire SPIDER data collection, containing 39,993 galaxies, with the Xeon Phi processor.

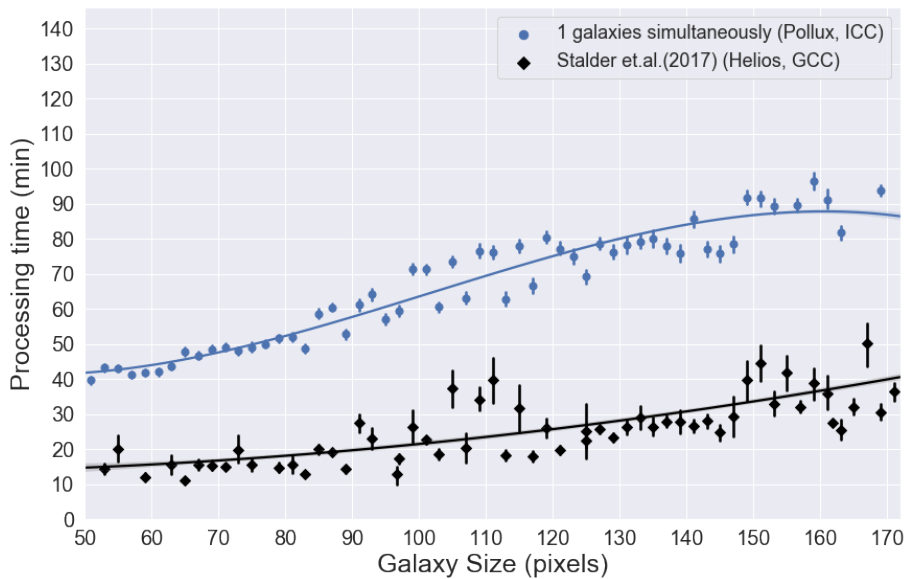


Figure 1. Comparison between Xeon (GCC) and Xeon Phi (ICC) running each galaxy (in a subset of 1,200 galaxies) with the Sérsic model. The data was cleaned from outliers greater than $2 * \text{Standard Deviation(STD)}$ on both X and Y axis. The chart presents the binned results of processing 1200 galaxies (this is better for visual analysis). The X variable was divided into discrete bins and then we estimated the central tendency and a confidence interval.

The critical point in this comparison is that the Xeon Phi has a much weaker single-core performance. Nevertheless, when we switched to the Intel ICC compiler, we were able to decrease the performance difference between the two types of processors. This considerable boost of single-core performance due to the compiler, along with the much higher number of available threads, opened a path to improvements with the Xeon Phi, which we will discuss in the next section.

3.4. Method of Performance Improvement

Besides raw performance, we also addressed scalability effects. GALPHAT, as we have mentioned earlier, is already parallelized with MPI and is scalable inside a single processing node of conventional server CPUs. However, it lacks scalability on many-core systems like Xeon Phi or clusters of several nodes containing hundreds or thousands of cores. The reason for it is that each model has its optimal range of MCMC's that one

can allocate for a galaxy processing. While allocating more chains in that range, the performance growth is noticeable, but not overwhelming. The problem is when one tries to allocate more MCMC's than GALPHAT can manage for a given model. It can result in stack processing and completely unreliable results. According to [Kolesnikov 2020] and [Stalder Diaz 2017], one will not benefit greatly from having a machine with more than 72 cores for the processing of a single galaxy. While there is cases (especially with S+E model) that require more that this, its rare and speedup gained is not significant. To overcome this limitation, we developed a submission system to distribute galaxies through available cores, such that multiple galaxies can be processed simultaneously. This is the benefit of submission system - being able to schedule and distribute several galaxies instead of just one. After completing this new submission system development, we observed a performance improvement for the Xeon Phi machine, enabling easier use of the resources available, and bringing the Xeon Phi's performance on par and even surpassing the results obtained by previous research. This benchmark comparison can be seen in Figure 2. The only caveat of this new scheme is much higher usage of RAM on the Xeon Phi machine, because processing of each galaxy by Sérsic roughly consumes 20 GB of RAM. Thus, processing six galaxies at the same time will require 120 GB of available RAM.

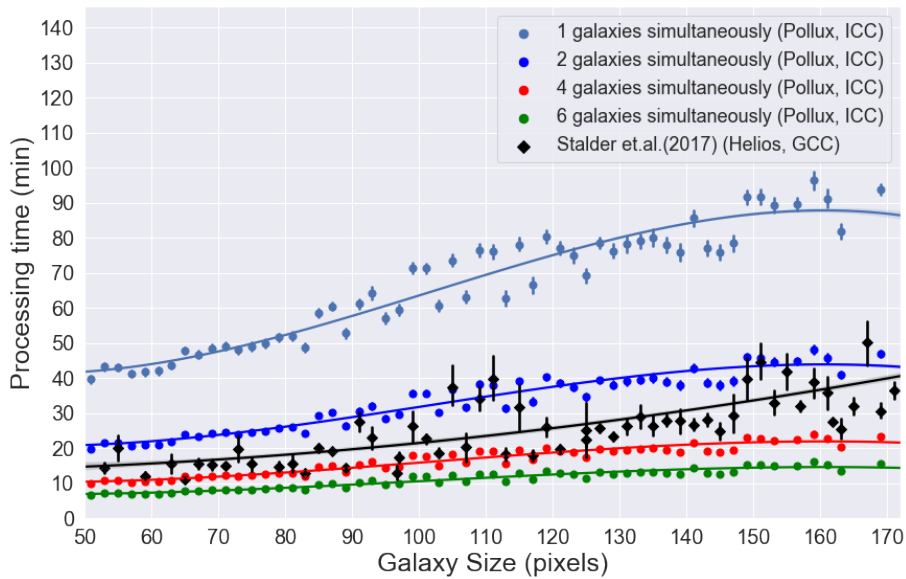


Figure 2. This figure shows the results of processing several galaxies simultaneously on the Xeon Phi using the developed submission system. These results are compared to the performance of previous studies [Stalder et al. 2017], where a machine (Helios) with a Xeon E5-2660 processor had been used. Each line corresponds to a number of galaxies that are executed simultaneously (submitted at once) on Xeon Phi (Pollux), using 20, 40, 80 and 120 GB of RAM and 20, 40, 80 and 120 threads, respectively, under the Sérsic model. The data was cleaned from outliers greater than $2 * \text{STD}$. This shows that when running four simultaneous galaxies, we were able to surpass the results from Stalder's research.

While one could view as a good idea to run, for example, ten galaxies at the same time, using 200 threads and 200 GB of memory, we cannot forget that the Xeon Phi has 68 real cores, each supporting up to four threads, resulting in 272 threads that could run simultaneously. However, it does not mean that these threads have the same performance as

the real cores. It is merely due to CPU co-scheduling, which enables us to schedule more than one task on the same core. Intel calls this technology HyperThreading [Stokes 2002]. During the tests, we saw that when one uses more than 120 active threads, performance starts to suffer penalties. The time of processing starts to grow, which is detrimental to the gains. This result corroborates our previous findings during testing with the NAS benchmark, in section 3.2. We empirically found that an optimal number of galaxies to be run simultaneously on this machine is six, using 120 active threads throughout the testing.

3.5. Sérsic+Exponential Performance Remarks

Our initial tests with S+E on Xeon Phi have shown that it is entirely unfeasible to use this machine to process this sophisticated model [Kolesnikov 2020]. Due to the relatively poor single-core performance, in some cases the difference in time reached tenfold or more to process the same galaxy with S+E model compared to the Xeon E5-2695v2 processor. This sometimes led to nearly one day of processing time for a single galaxy.

As a consequence, we moved the processing of the S+E model to the Santos Dumont system, where, at the time, we received access to a dedicated queue, which helped and continues to help greatly our research. Although we are at a fairly early stages of processing the S+E model, we expect that the better single-core performance of the Xeon processors on Santos Dumont will provide better throughput to our production runs.

3.6. Optimization of Workflow Pipeline

Another problem linked with performance is the time consumed for the preparation of executions and analysis of results. The entire workflow pipeline of our cosmological framework [Kolesnikov 2020, Stalder Diaz 2017] roughly consists of three main steps: pre-processing, model processing, and post-processing. To improve performance of steps that were not parallelized, our choice of parallelization tool was the *joblib*⁴ library. By default *joblib.Parallel* uses the *loky* back-end module to start separating Python worker processes to execute tasks concurrently on separate CPU cores. This is a reasonable default for generic Python programs but can induce a significant overhead as the input and output data need to be serialized in a queue for communication with the worker processes [Varoquaux and Grisel 2009]. This library enables simple and logical parallelization of embarrassingly parallel tasks. We have also rewritten our implementations of the converter routine that processes output data, including the plot generator of produced files.

The clock speed of 1.4 GHz (1.6 GHz in Intel TurboBoost) is the weakness of single-core performance on the Xeon Phi, even in comparison with the 8th generation of energy-efficient U line of processors for notebooks (Intel i5-8350u, 4 cores/8 threads@1.8GHz; 3.6GHz in Turbo-Boost) or AMD’s 2nd generation desktop processor Ryzen 2700X (8 cores/16 threads@3.7GHz; 4.3GHz in AMD TurboCore). However, when we switch to parallel versions of the script and run it with a high enough core count, the Xeon Phi still comes out as a clear winner in regard to speedup. The results of our testing of the updated routines can be seen on Figures 3 and 4. Here, again, we need to note that, during this operation, the Xeon Phi was consuming considerably more RAM (proportional to the increase of threads used), at a much higher level than its counterparts.

⁴<https://joblib.readthedocs.io/en/latest/>

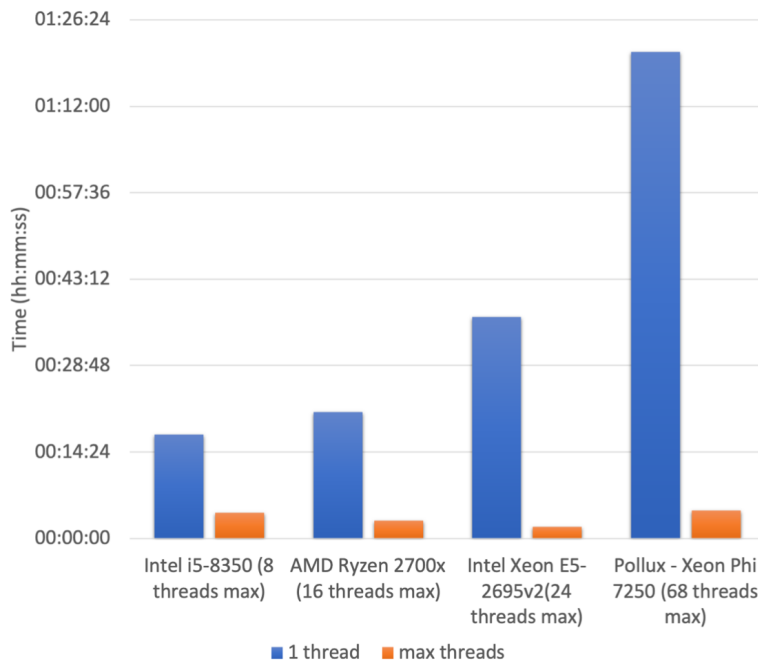


Figure 3. Performance gains of compression routine parallelization on various platforms. The comparison is between usage of one thread and the optimal amount on each processor. Even though the gains are observed for all platforms, the speedup on the Xeon Phi machine is the most prominent, showing the importance of using as many threads as possible on this processor family, to make it comparable to the other processors.

In summary, parallelizing these two phases decreased the times of pre- and post-processing steps significantly. As a pleasant bonus, the new codebase of post-processing is highly scalable, so if it becomes necessary to integrate new functionality (such as a new type of chart), it could be done effortlessly and already in a parallel manner.

4. Environment for Production Experiments

With the work described above, we were able to improve performance and usability of GALPHAT's pipeline significantly, making it more automated and faster on the Xeon Phi processor. We used this improved pipeline to analyze the entire SPIDER sample with the Sérsic model, processing nearly 40,000 galaxies in about a year. Also, we extended the pipeline with support for the S+E model and modularized it to facilitate future extensions.

To obtain good performance with GALPHAT, it is possible to use either a small number of high-frequency, powerful cores, or many low-frequency, less powerful cores like in the Xeon Phi. In this second case, it is necessary to have higher amounts of RAM to compensate the low frequency, and run several galaxies at the same time. For the amount of RAM, it is easy to follow a rule of one Monte Carlo Chain (1 thread) with 1 GB of RAM. This means that the processing of a galaxy by the Sérsic model, which uses 20 chains, would require 20 threads and 20 GB of available RAM.

5. Related Work

The galaxy modeling field is dominated by solutions based on the frequentist approach. These solutions tend to be faster in processing the data, but this speed comes with a cost.

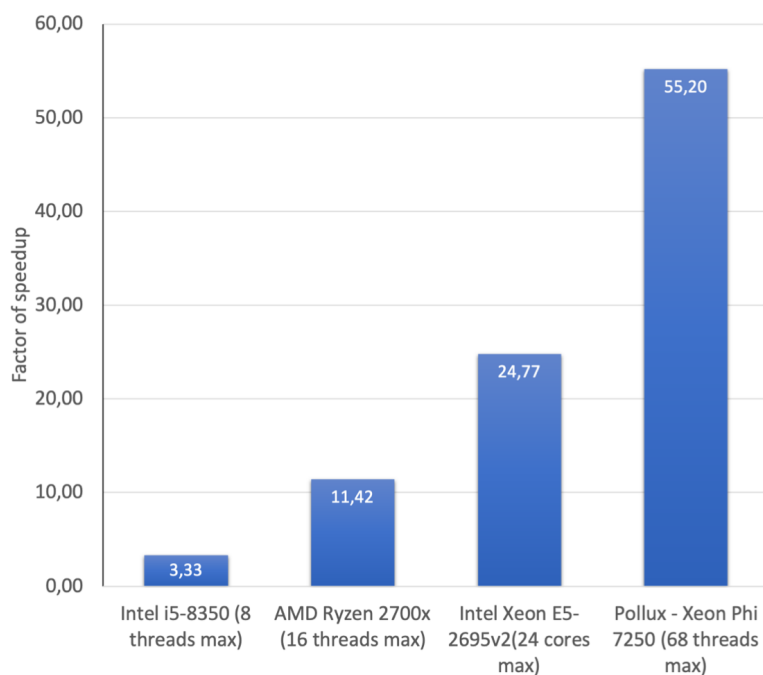


Figure 4. Performance gains on various platforms when comparing sequential and parallel versions of the code for generation of the posterior plot. We can see from the chart that, once again, the Xeon Phi machine had a more significant improvement than any other. This data was generated with the following equation: $factor = sequentialTime/parallelTime$. Where $sequentialTime$ is the best sequential result and $parallelTime$ is the best parallel time.

Commonly, the multi-modal signature is weak when analyzed by maximum likelihood methods (ML) such as GALFIT [Peng et al. 2002]. Meanwhile, Bayesian inferences with elicited priors, such as used by GALPHAT, were found to be less biased, especially for high-concentration profiles: GALPHAT's Sérsic index n , re , and MAG deviate from their actual values by 6%, 7.6%, and -0.03 mag, respectively, while GALFIT deviates by 15%, 22%, and -0.09 mag, respectively [Stalder et al. 2017].

The key to testing galaxy formation theories and evolution is to make full use of the information in galaxy image data. Algorithmic approaches for describing two-dimensional surface photometry profiles - e.g., SExtractor [Bertin and Arnouts 1996], GIM2D [Simard 1998], GALFIT [Peng et al. 2002], 2DPHOT [La Barbera et al. 2008], GALAPAGOS [Barden et al. 2012] - are based on maximum likelihood estimation (MLE), or more generally, optimizing an objective function that differentiates between two distributions.

The approach above, however, has significant limitations. Firstly, the estimated structural parameters are affected by random and systematic errors. For example, pixel integration, rotation, and convolution techniques used to generate model predictions, as well as the background noise, contamination by nearby objects, initial guesses, the form of the objective function and the models themselves, minimization algorithms, and image sizes [Häussler et al. 2007, Vikram et al. 2010, Guo et al. 2009, Simard et al. 2011, Mendel et al. 2013, Bernardi et al. 2017] all may cause deviations

from the correct model. Secondly, a simple change in the parameterization for the data values can change the likelihood function's shape and affect the inferred confidence regions, as can be seen by applying the derivative chain rule on the likelihood function. Therefore, inferred galaxy properties using ML fitting tools can be affected significantly [Bernardi et al. 2003, Hyde and Bernardi 2009] by seemingly innocuous changes in the problem definition. Thirdly, the ML approach cannot easily select between various models for spheroids, bulges, discs, and/or point sources given a particular galaxy image. In other words, ML provides no relative measure of how well the model explains the data except in the special case of nested models. This is one of the crucial points for the achievement of our goal, in the long run, and justification for the attempts to optimize and improve performance in order to make GALPHAT a more useful tool.

6. Conclusion

In conclusion, we have shown in this article that it is indeed possible/viable to process a large amount of data locally with a Bayesian Inference tool. We achieved this with the Sérsic model processing the SPIDER sample, containing thousands of galaxies. The execution time for the entire sample was on the order of about a year on a system with a many-core processor (Intel Xeon Phi). However, for an increased number of galaxies, or for a more complex model (such as Sérsic+Exponential), we would need a more powerful system to complete the task in a similar time range.

The work conducted so far in this research resulted in the updating and optimization of the submission pipeline and its automation to facilitate the usage on multi-core systems. This pipeline is built upon the refactored, optimized, and extended code-base that was inherited from previous research. At the same time, the large volume of the resulting data requires organization and a tool for its visualization, so we implemented the first version of diagnostic and monitoring systems that evolved into visualization and analysis tools for our data [Kolesnikov 2020]. In the long run, we aim to integrate and unify related tools to facilitate and optimize the process of galactic modeling and analysis.

6.1. Future Work

As future work, we have the goal of finishing the processing of the Sérsic+Exponential model, possibly using GPGPU technology to speedup even more the processing routine. The next step would be gathering and organizing all the data, and starting the analysis of the results, focusing on the comparison of Bayes Factor of the two models. However, at the same time, one ought to continue implementing improvements of the processing pipeline. We hope that with data produced by the pipeline we will be able to get key insights on galaxy formation and evolution inside the clusters. The tools that were developed during this research should greatly facilitate the process of analysis. The next phase will consist in a detailed analysis of cosmological data and extension of our astronomical knowledge with results from that analysis.

7. Acknowledgments

IK acknowledges financial support from CAPES scholarships. The authors had support from FAPESP via grant 111564/15. Much of this work was made possible with help from LNCC (Santos Dumont Supercomputer) and from MCTI/INPE.

References

- Abazajian, K. N., Adelman-McCarthy, J. K., Agüeros, M. A., Allam, S. S., Prieto, C. A., An, D., Anderson, K. S., Anderson, S. F., Annis, J., and Bahcall, N. A. (2009). The seventh data release of the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 182(2):543.
- Andredakis, Y. and Sanders, R. (1994). Exponential bulges in late-type spirals: an improved description of the light distribution. *Monthly Notices of the Royal Astronomical Society*, 267(2):283–296.
- Barden, M., Häußler, B., Peng, C. Y., McIntosh, D. H., and Guo, Y. (2012). Galapagos: from pixels to parameters. *Monthly Notices of the Royal Astronomical Society*, 422(1):449–468.
- Bernardi, M., Fischer, J.-L., Sheth, R., Meert, A., Huertas-Company, M., Shankar, F., and Vikram, V. (2017). Comparing pymorph and sdss photometry–ii. the differences are more than semantics and are not dominated by intracluster light. *Monthly Notices of the Royal Astronomical Society*, 468(3):2569–2581.
- Bernardi, M., Sheth, R. K., Annis, J., Burles, S., Eisenstein, D. J., Finkbeiner, D. P., Hogg, D. W., Lupton, R. H., Schlegel, D. J., SubbaRao, M., et al. (2003). Early-type galaxies in the sloan digital sky survey. iii. the fundamental plane. *The Astronomical Journal*, 125(4):1866.
- Bertin, E. and Arnouts, S. (1996). SExtractor: software for source extraction. *Astronomy and Astrophysics Supplement Series*, 117(2):393–404.
- Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335.
- Guo, Y., McIntosh, D. H., Mo, H., Katz, N., Van Den Bosch, F. C., Weinberg, M., Weinmann, S. M., Pasquali, A., and Yang, X. (2009). Structural properties of central galaxies in groups and clusters. *Monthly Notices of the Royal Astronomical Society*, 398(3):1129–1149.
- Häußler, B., McIntosh, D. H., Barden, M., Bell, E. F., Rix, H.-W., Borch, A., Beckwith, S. V., Caldwell, J. A., Heymans, C., Jahnke, K., et al. (2007). Gems: galaxy fitting catalogs and testing parametric galaxy fitting codes: Galfit and gim2d. *The Astrophysical Journal Supplement Series*, 172(2):615.
- Hyde, J. B. and Bernardi, M. (2009). Curvature in the scaling relations of early-type galaxies. *Monthly Notices of the Royal Astronomical Society*, 394(4):1978–1990.
- Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.
- Kolesnikov, I. (2020). Study and optimization for high performance processing with galphat. Master’s thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos.
- La Barbera, F., De Carvalho, R., de La Rosa, I., Lopes, P., Kohl-Moreira, J., and Capelato, H. (2010). Spider–i. sample and galaxy parameters in the grizyjhk wavebands. *Monthly Notices of the Royal Astronomical Society*, 408(3):1313–1334.

- La Barbera, F., De Carvalho, R., Kohl-Moreira, J., Gal, R., Soares-Santos, M., Capaccioli, M., Santos, R., and Sant'Anna, N. (2008). 2dphot: a multi-purpose environment for the two-dimensional analysis of wide-field images. *Publications of the Astronomical Society of the Pacific*, 120(868):681.
- Mendel, J. T., Simard, L., Palmer, M., Ellison, S. L., and Patton, D. R. (2013). A catalog of bulge, disk, and total stellar mass estimates for the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 210(1):3.
- Peng, C. Y., Ho, L. C., Impey, C. D., and Rix, H.-W. (2002). Detailed structural decomposition of galaxy images. *The Astronomical Journal*, 124(1):266.
- Sérsic, J. (1963). Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletín de la Asociación Argentina de Astronomía La Plata Argentina*, 6:41.
- Simard, L. (1998). Gim2d: an iraf package for the quantitative morphology analysis of distant galaxies. In *Astronomical Data Analysis Software and Systems VII*, volume 145, page 108.
- Simard, L., Mendel, J. T., Patton, D. R., Ellison, S. L., and McConnachie, A. W. (2011). A catalog of bulge+ disk decompositions and updated photometry for 1.12 million galaxies in the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 196(1):11.
- Stalder, D., de Carvalho, R. R., Weinberg, M. D., Rembold, S. B., Moura, T. C., Rosa, R. R., and Katz, N. (2017). Bayesian surface photometry analysis for early-type galaxies. *arXiv preprint arXiv:1711.02188*.
- Stalder Diaz, D. H. (2017). *Applied computing to study structural and environmental properties of SDSS's galaxies / Computação aplicada ao estudo das propriedades estruturais e ambientais de galáxias do SDSS*. PhD thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos.
- Stokes, J. (2002). Introduction to multithreading, superthreading and hyperthreading. *ArsTechnica.com*.
- Varoquaux, G. and Grisel, O. (2009). Joblib: running python function as pipeline jobs. 2009. Available from <https://pypi.python.org/pypi/joblib#downloads>.
- Vikram, V., Wadadekar, Y., Kembhavi, A. K., and Vijayagovindan, G. (2010). Pymorph: automated galaxy structural parameter estimation using python. *Monthly Notices of the Royal Astronomical Society*, 409(4):1379–1392.
- Weinberg, M. D., Yoon, I., and Katz, N. (2013). A remarkably simple and accurate method for computing the bayes factor from a Markov Chain Monte Carlo simulation of the posterior distribution in high dimension. *arXiv preprint arXiv:1301.3156*.
- Yoon, I., Weinberg, M. D., and Katz, N. (2011). New insights into galaxy structure from galphot-i. motivation, methodology and benchmarks for sérsic models. *Monthly Notices of the Royal Astronomical Society*, 414(2):1625–1655.