

Performance of a distributed three dimensional Coprocessing code for the Klein Gordon equation

M. Quell

Abstract

This study compares the performance of parallel pseudospectral Klein Gordon solver on the VSC2 [1] with data output, without output and using coprocessing. It can be concluded that writing data to disk is time consuming.

Example Simulation

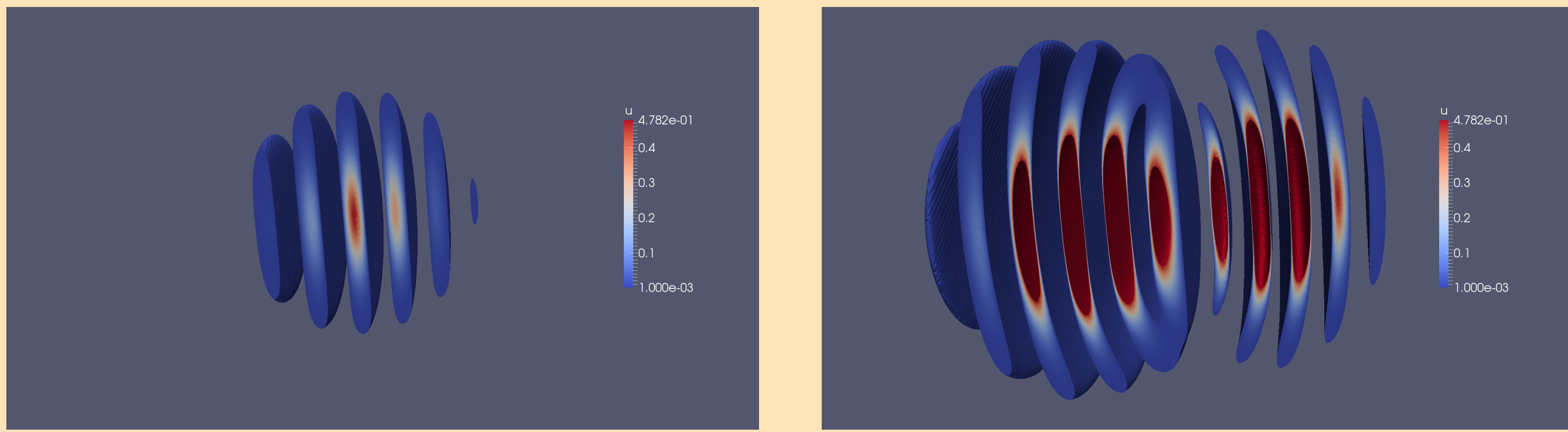


Figure 1: The images show propagating of the initial condition on a $[-3\pi, 3\pi]^3$ domain. Plots are at times $t = 0, 1$. The timestep was 10^{-3} and resolution was 512^3 .

Equation

3D Klein-Gordon Equation

$$\mathbf{u}_{tt} - \Delta \mathbf{u} + \mathbf{u} = |\mathbf{u}|^2 \mathbf{u}$$

where t is the time, $\Delta = \partial_{xx} + \partial_{yy} + \partial_{zz}$ and \mathbf{u} is a scalarfield. The simulation was done with periodic boundary conditions and exponentially decaying initial conditions. A second order semi-implicit Fourier pseudospectral discretization scheme used by Donninger and Schlag [2] and implemented in this study is

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{\delta t^2} - \Delta \frac{u^{n+1} + 2u^n + u^{n-1}}{4} + \frac{u^{n+1} + 2u^n + u^{n-1}}{4} = |u^n|^2 u^n.$$

This scheme requires one FFT per time step. The implementation used allows for the field u to be complex.

Data Distribution and Visualisation

Data distribution and parallel Fourier transforms are done with the library 2Decomp&FFT [3]. ParaView Catalyst [4, 5] embedded within a full ParaView built with Mesa for software-based rendering is used in this study. The Catalyst within the latest ParaView release 4.1.0 was used.

To use ParaView Catalyst for *in situ* processing includes:

- writing an adaptor code to interface the simulation with the Catalyst libraries
- instrumenting the simulation code
- setting up an analysis and visualisation pipeline for co-processing

To set up the visualisation pipeline using ParaView client GUI as a local preprocessing step using a very small grid size simulation, hence building the ParaView client is skipped and only the various servers on VSC2 are built. The visualisation pipeline is generated as a python code by the ParaView client GUI, it would be possible though to create a hardcoded C++ Version. All plugins are disabled in order to streamline the process.

In the simulation code:

- the coprocessor needs to be initialized
- the adaptor subroutine needs to be called during the simulation loop at the end of each time step to transform the simulation grid and field data structures into VTK data structures that Catalyst can process
- lastly the *coprocessorfinalize* function is called to shutdown the coprocessor

Benchmark Results

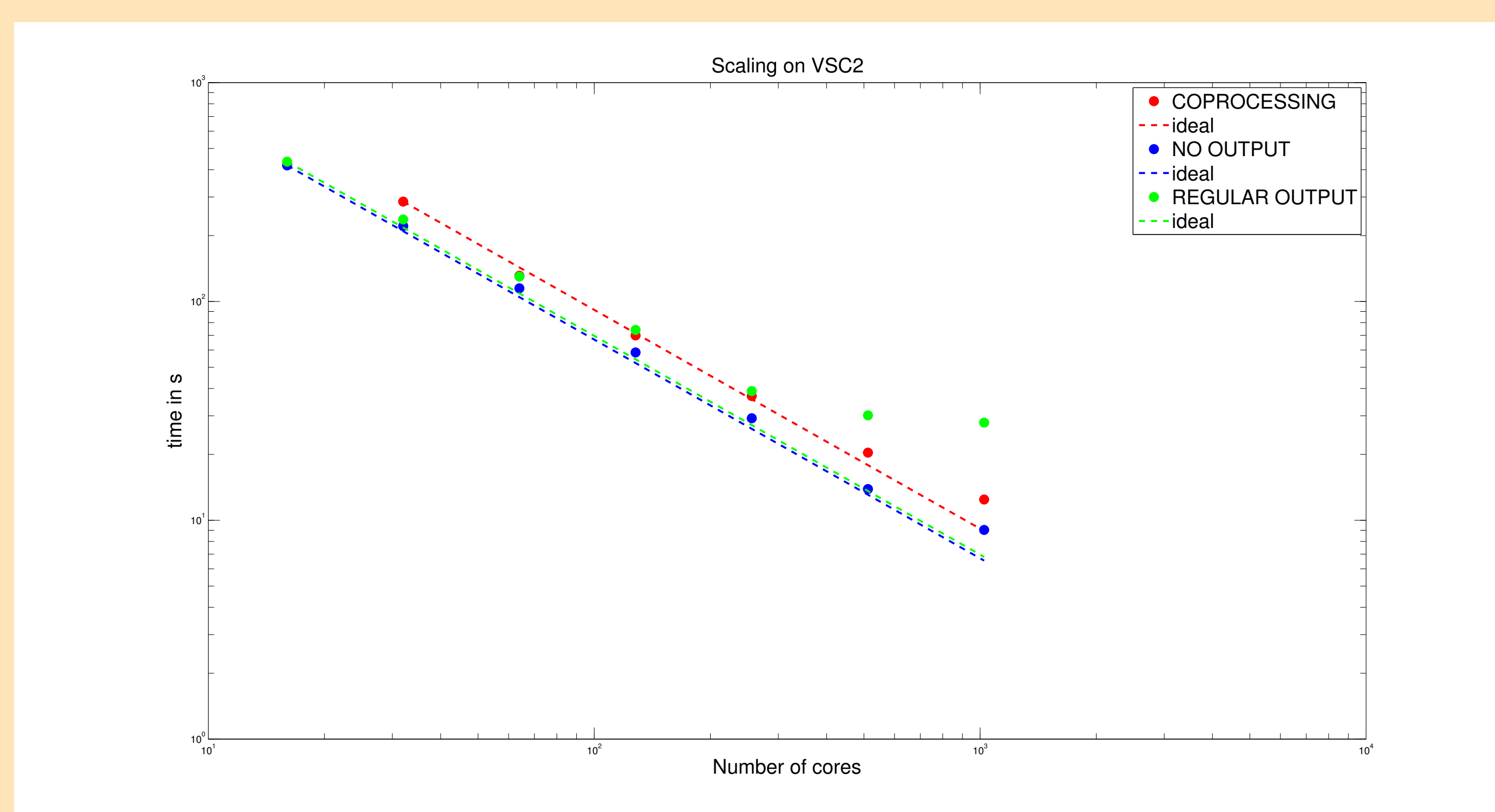


Figure 2: Computation time in seconds for 30 timesteps on a 512^3 grid, output every 10th step on VSC2. The same pipeline as in the example simulation was used, *IsoVolume* which needs extra memory, because of the unstructured grid.

Conclusion

Coprocessing is a very fast method of visualisation your data, because writing data to disk can be very time consuming and the scaling is better than for the data output. But on the negative side for coprocessing is that you are blind during the process, because if you did not setup the pipeline adapted to your data you need to run the whole simulation again.

Acknowledgments

This research used resources of the VSC2. I also like to thank W. Auzinger, O. Koch and B. K. Muite without them this would not have been possible.



Contact

e-mail: michael.quell@yahoo.de

Homepage: <http://web.student.tuwien.ac.at/~e1226394/>

Youtube: <https://www.youtube.com/channel/UCSG6bca26nybgK39n5vnE6w>

References

- [1] Vienna Scientific Cluster 2, <http://vsc.ac.at/>
- [2] Donninger, R. and Schlag, W., "Numerical study of the blowup/global existence dichotomy for the focusing cubic nonlinear Klein-Gordon equation", *Nonlinearity* 24, 2547-2562, (2011).
- [3] Li, N. and Laizet, S., "2DECOMP&FFT - A highly scalable 2D decomposition library and FFT interface", *Proc. Cray User Group 2010 Conference*
- [4] Fabian, N. D., Moreland, K. D., Thompson, D., Bauer, A. C., Marion, P., Geveci, B., Rasquin, M., Jansen, K. E., The ParaView coprocessing library: a scalable, general purpose in-situ visualization library. *LDAV 2011 IEEE Symposium on Large-Scale Data Analysis and Visualization*, (Oct. 2011), 89-96. DOI=<http://doi.acm.org/10.1109/LDAV.2011.6092322>.
- [5] Bauer, A. C. and Geveci, B. and Schroeder, W., "The ParaView Catalyst User's Guide v1.0", <http://paraview.org/Wiki/images/4/48/CatalystUsersGuide.pdf>. Kitware, (2013).
- [6] Chen, G., Cloutier, B., Li, N., Van Moer, M., Muite, B.K. and Rigge, P. https://en.wikibooks.org/wiki/Parallel_Spectral_Numerical_Methods/
- [7] Aseeri, S., Batrashev, O., Icardi, M., Leu, B., Li, N., Liu, A., Muite, B.K., Palen, B., Quell, M., Servat, H., Sheth, P., Speck, R., Van Moer, M. and Vienne, J. "Solving the Klein Gordon equation using Fourier spectral methods: A benchmark test for computer performance" (2014)