

ARCHER eCSE01-001 Technical Report

Zacros Software Package Development: Pushing the Frontiers of Kinetic Monte Carlo Simulation in Catalysis

Jens Nielsen,¹ James Hetherington,¹ and Michail Stamatakis²

¹ Research Software Development Team, Research IT Services, University College London, Torrington Place, London WC1E 6BT, United Kingdom

² Department of Chemical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom

Abstract

The project set out to deliver spatial parallelism for Zacros, a code for stochastic modelling of chemical processes on a surface. In the project we have successfully parallelized the code with both an exact and an inexact algorithm. The exact algorithm does not provide any speed up but using the inexact algorithm we were able to gain a speed up of up to 23 times on 24 ARCHER nodes for a relevant benchmark example on a moderate sized lattice.

1 Introduction

In this project we have added spatial parallelism for Zacros, a code for stochastic modelling of chemical processes on a surface, based on a graph theoretical kinetic Monte Carlo (KMC) approach. The general KMC framework used in computational catalysis applications is shown in Figure 1 below.¹

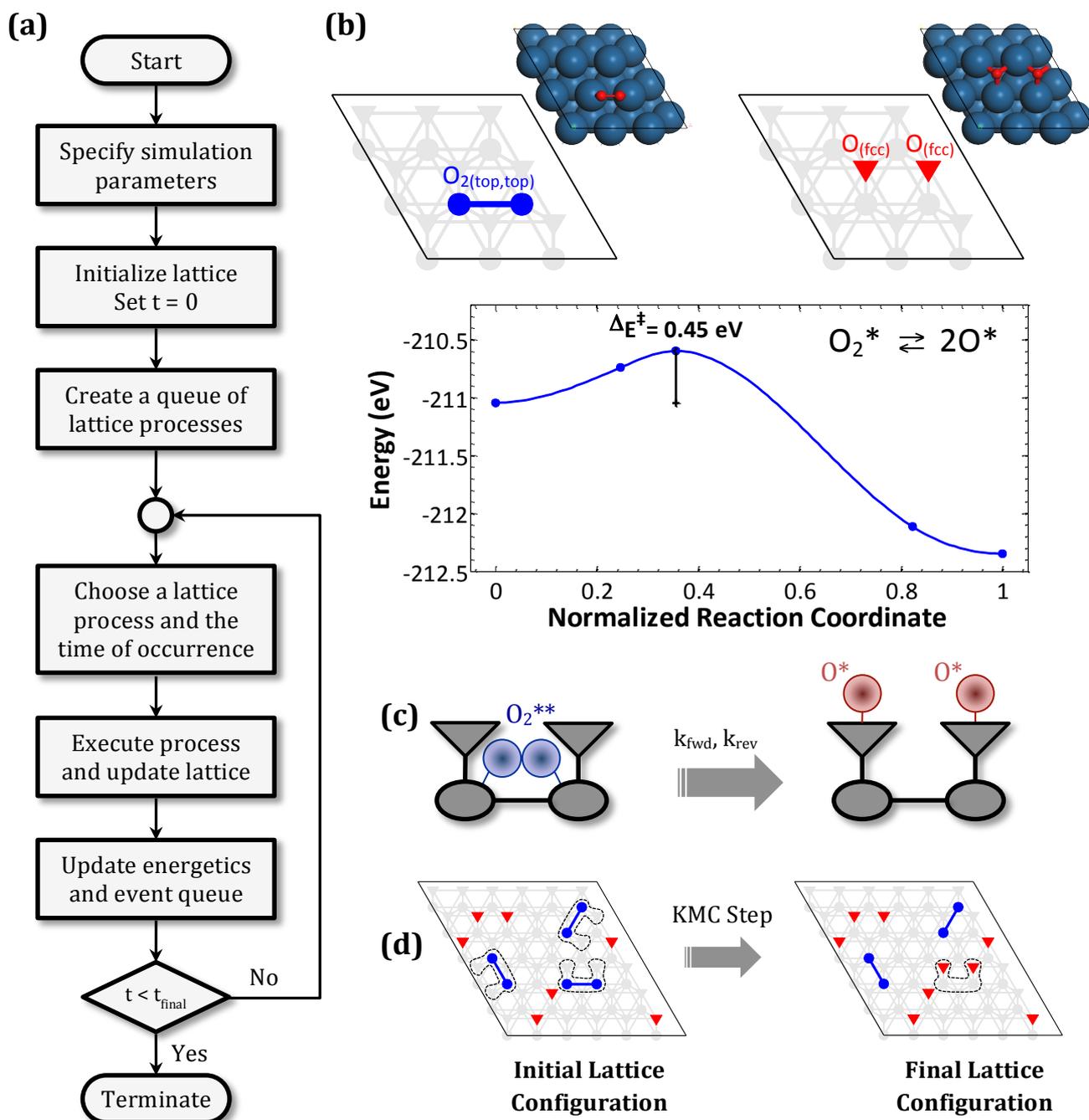


Figure 1: (a) Generic algorithm for a lattice KMC simulation. Initialization entails the definition of a reaction network and the specification of operating conditions. (b, c) Each elementary step can be parameterised through quantum chemistry codes. An example for the O_2 dissociation on Pt is shown: The top pictures (b) show the unit cell and the configurations used in a DFT code, along with the lattice

representation of these configurations, and an NEB calculation showing the MEP whose maximum is the TS. The scheme below (c) portrays a reaction pattern in the KMC framework. The reaction's forward and reverse kinetic constants k_{fwd} and k_{rev} can be calculated using transition state theory. (d) During a KMC simulation the reaction pattern is being searched for on the lattice, and each occurrence is listed as a possible lattice process. At the next KMC step, one of these possible processes is executed, and the lattice state is updated.

In our graph-theoretical approach the activation energies (and consequently the reaction propensities) of surface reactions depend on the number and location of spectator molecules in the vicinity of the reaction (see Figure 2).

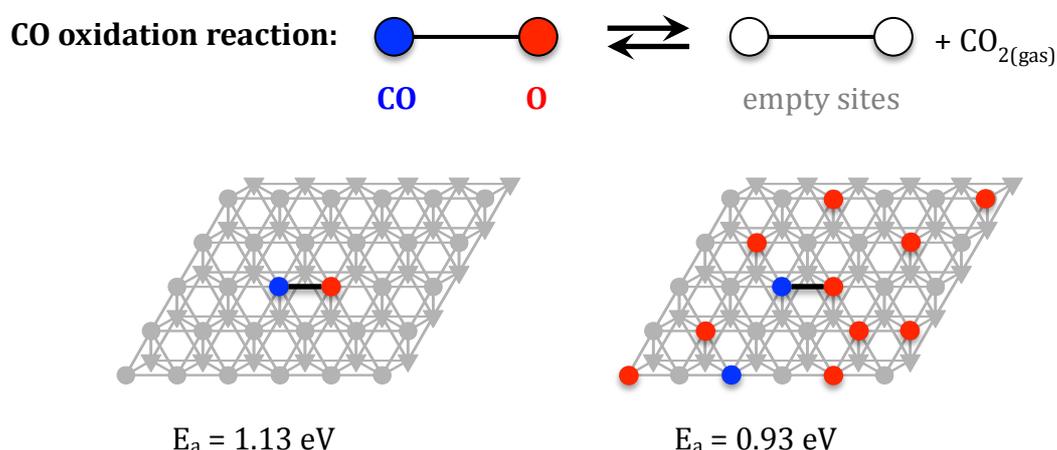


Figure 2: The CO oxidation reaction has an activation energy of 1.13 eV in the absence of spectator molecules (zero coverage limit, left panel). However, in the presence of spectator molecules (right panel) the activation energy drops, due to the repulsive interactions between the reacting molecules and the spectators.²

For such intrinsically stochastic methods, where each run results in a single sample from a distribution of possible simulations, two approaches are possible; *exact parallelisations* that are guaranteed to give equivalent results to a serial simulation, and *approximate parallelisations* those that introduce some further error, but with potentially greater speedup.

2 Scientific area

The importance of heterogeneous catalysis in applications that enhance the quality of life cannot be overstated. Iron-based catalysts for the production of ammonia by the Haber process is a notable example, the importance of which becomes evident by the fact that fertilisers, generated from ammonia thus produced, are responsible for sustaining one-third of the Earth's population.

In such catalytic processes, reactant molecules adsorb on the surface of a material (such as a transition metal), and interact with each other and with the

catalyst in such a way that (i) the overall reaction (e.g. ammonia production) is accelerated, and (ii) the material regenerates in the end of the catalytic cycle. However, finding such catalytically active materials is non-trivial: traditional practice involves trial- and-error experimentation which is time-consuming, costly, and does not guarantee success. On the other hand, simulations can yield vital information and a detailed understanding of how catalysts work and what affects their performance. Our kinetic Monte Carlo methodology and software package, *Zacros*, have already contributed to the unravelling of complex phenomena on catalytic surfaces. This knowledge can be used to improve existing catalysts or design novel materials with optimal activity and selectivity for bespoke applications.

An issue that has limited the accuracy of the predictions and the applicability of KMC is that simulations on large lattices/domains are prohibitively expensive. Such large-scale simulations are however necessary for two main reasons: (i) Improving the accuracy of the simulation. When estimating quantities from stochastic simulations the noise inherent to the approach results in errors. Observables, such as the coverage of an adsorbate or the average activation energy of an elementary event, exhibit fluctuations, which however drop with the square root of the number of sites of the simulated lattice. (ii) Capturing relevant physics. A phenomenon of significant interest to the catalysis community is catalyst reconstruction, which encompasses a large class of dynamic changes in the surface structure of the catalyst. Such changes have been shown to lead to pattern formation, such as spiral waves, first described by Nobel laureate Prof. G. E.rtl and coworkers.⁵ However, the detailed molecular mechanisms that give rise to these patterns are not well understood. A major obstacle is that they evolve over extended length scales, with wavelengths on the order of hundreds of micrometers, whereas current KCM approaches are limited to domain sizes of tens of nanometers.

In this project, we have made it possible for multiple computers to work together to run simulations of large KMC domains, with each computer tackling a different part of the domain, producing results much more quickly. With the probabilistic nature of statistical mechanics, achieving consistent results between these computers requires careful bookkeeping. We have therefore proceeded by first implementing a solution which is known to be consistent, but where the cooperating computers ensure that only one is acting at a time. This doesn't result in faster simulations, but allows us to prove that our approach to sharing out the work is correct. From this baseline, we can then subtly relax the requirements for consistency, producing approximately correct results much more quickly. This has resulted in 23x speedup using 24 ARCHER nodes. The eCSE project will therefore enable transformative research in the field of catalysis, by making it possible to efficiently conduct accurate large-scale simulations of phenomena such as reconstruction and pattern formation on catalytic surfaces.

3 Zacros

Zacros is a Kinetic Monte Carlo software package written in Fortran for simulating molecular phenomena on catalytic surfaces. The tool enables researchers in the areas of Computational Catalysis and Surface Science to perform dynamic modelling of adsorption, desorption, surface diffusion, and reaction processes on heterogeneous catalysts. The rates of these elementary processes are typically computed from *ab initio* simulations, thereby enabling the prediction of catalytic performance metrics (such as activity and selectivity) from first principles, and the detailed validation of hypothesised kinetic mechanisms against experimental data.

The package employs the Graph-Theoretical KMC methodology coupled with cluster expansion Hamiltonians and Brønsted-Evans-Polanyi relations, which can naturally capture:

- steric exclusion effects for species that bind in more than one catalytic sites,
- complex reaction patterns involving adsorbates in specific binding configurations and neighbouring patterns,
- spatial correlations and ordering arising from adsorbate lateral interactions that involve many-body contributions,
- changes in the activation energies of elementary events, influenced by the energetic interactions of reactants with neighbouring spectator species.

The software was commercialised by UCL Business in Nov-2013, and has a dedicated website (<http://zacros.org>) with documentation, tutorials, list of publications and links to e-Lucid, the UCL online licensing portal where users can obtain Zacros from. The code can be obtained free of charge by academics, and since inception it has been distributed to 68 users in 23 countries worldwide.

While not available as a module the software can be simply installed on ARCHER using the CMAKE based installation tools we have developed and maintain. Request for access to the Zacros repository on Github are reviewed and granted on an individual basis.

4 Parallelisation work

Since this is the first such code to be spatially parallelised, significant new issues must be addressed to achieve this. Hence, for this project, our objective has been first to achieve a correct spatial decomposition, independent of core count, and secondly, to achieve speedup on a small number of nodes.

In the project proposal, we described an exact parallelisation extending an approach by Lubachevsky,³ and an approximate parallelisation based on an approach by Martinez.⁴ However, early in the work, it became clear that the Lubachevsky approach itself would be approximate for this system, due to possible diffusive processes occurring much faster than adsorption/desorption and reaction events, thereby creating non-nearest neighbour interactions

between decomposition domains. Such situations can lead to causality paradoxes as shown in Figure 3 (below).

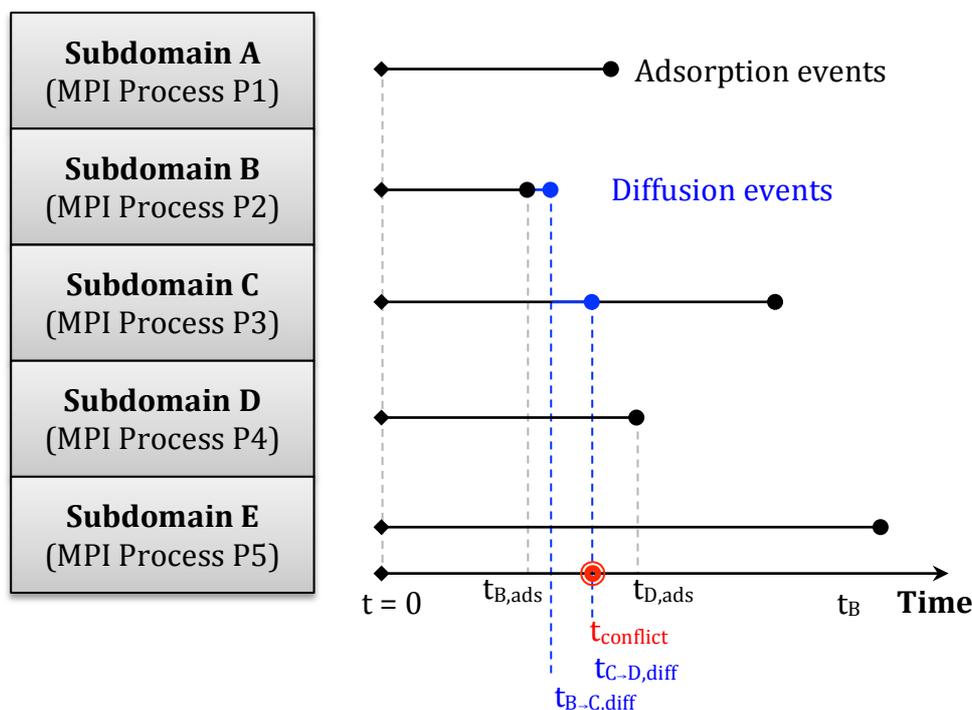


Figure 3: An example of a causality paradox. For details see text.

In this figure, five lattice domains are shown which are initially empty (no particles), and in which adsorption and diffusion events may occur. The waiting times for these events are depicted in black and blue, respectively, and it is assumed that diffusion is much faster than adsorption (as is usually the case in catalysis). In the Lubachevsky approach, each MPI process is permitted to execute a KMC process in its domain only if the occurrence time is lower than the occurrence times of the neighbours. By this criterion, MPI processes P2 and P4 will simulate one adsorption event each, on domains B and D at times $t_{B,ads}$ and $t_{D,ads}$, respectively. Now, the particle on domain B can diffuse into domain C at time $t_{B-C,diff}$, and into domain D at time $t_{C-D,diff}$. Since diffusion is fast, it is very probable that $t_{C-D,diff} < t_{D,ads}$ which would lead to a causality paradox, because the adsorption event on domain D has already been simulated assuming a completely empty domain D.

We have therefore completed the exact parallelisation work using an approach which forces only the domain on which the globally next occurring reaction to advance. While clearly, this cannot offer any parallel speedup, successful implementation of this is a major achievement for a complex code. The graph theoretical approach needed for modelling reactions with spatially extended molecules requires the sharing of many complex buffers with large haloes representing potential changes to the state of the lattice. Thus, very precise book-keeping is required to achieve this essential baseline for further simulation. We are pleased to have reached this milestone despite the

unexpected mathematical difficulties; a correct distributed-memory code with no speedup. Benchmarking confirms that this simulation runs slight slower than the serial implementation. On a relevant simulation the serial simulation performs 15.5 KMC events per second and the single process updating MPI simulation performs 15.1 KMC events per second.

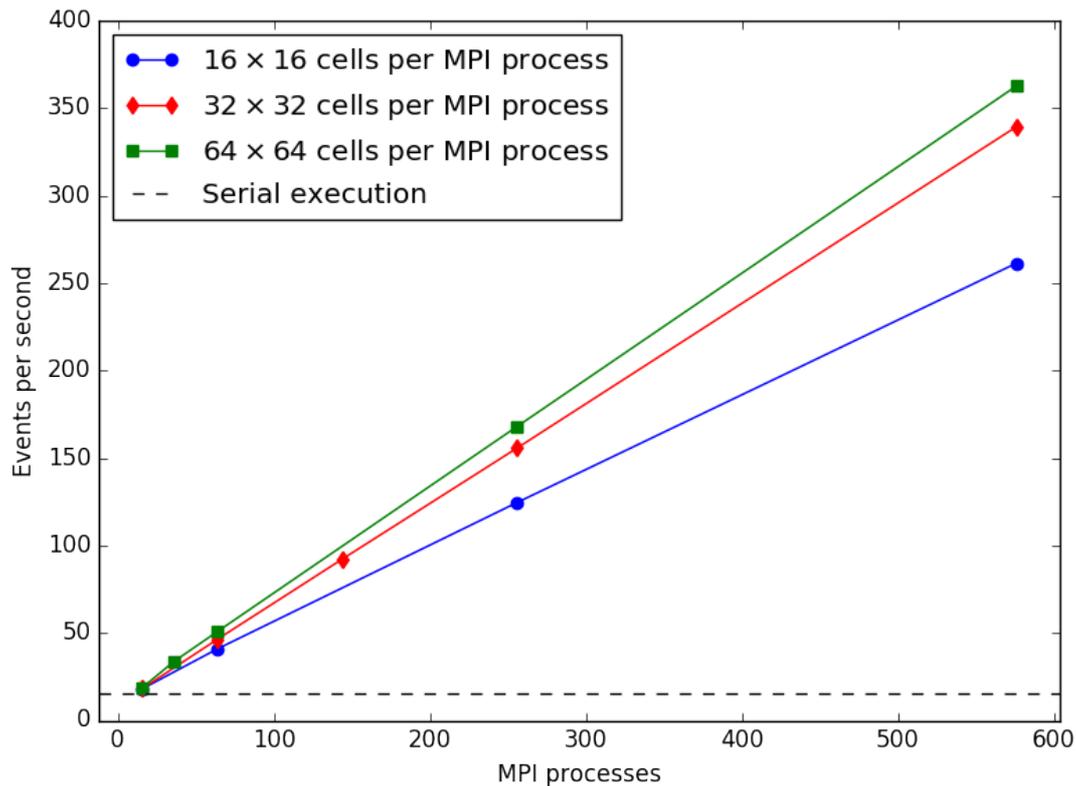


Figure 4 Weak scaling of the MPI code keeping the number of unit cells per MPI process constant. Shown is the number of KMC events simulated per second of computational time with respect to the number of MPI processes.

Proceeding from this stable base, we have then implemented an approximate method, where spatial decomposition is allowed to result in inconsistent changes between simulation domains. Using this approach, we have achieved parallel speed up of 23x on 24 ARCHER nodes (see figure 4). Thus, while the algorithms originally envisaged (Lubachevsky, Martinez) have not been implemented as they were found to be inappropriate, new solutions have been found, achieving the principle objectives by other means. Figure 4 shows the measured performance of the non-exact MPI code as a function of the number of MPI processes used. In this benchmark the number of unit cells per MPI process is kept constant. This is investigated for 3 different lattice sizes per MPI process. The 3 cases show similar trends with a near linear scaling with number of processes but with different slopes depending on the lattice size. The simulation is also compared to the serial case which is known to be independent on the lattice size (see report 1). As expected, a larger lattice with a smaller fraction of halo sites, and thus fewer halo interchanges, results in a better performance and

scalability. The lowest number of MPI processes investigated is 16 i.e. 4 x 4 nodes. Assuming a square lattice with a square distribution of lattice sites to MPI processes this is the smallest number of cores where we can hope to see any performance gain since all domains will be neighbours of all other domains in a 3 x 3 and 2 x 2 MPI processes configurations. In the case of 4 x 4 MPI processes there is a very modest speed gain from 15.5 KMC events per second to approximately 18 events per second. The speed gain is as expected to be very modest since at most 3 processes can do KMC events in parallel for a 4 x 4 configuration. As the number of processes increases, more and more processes can perform updates in parallel but the scaling is inherently limited by the ratio of processes that can perform updates to the total number of processes.

Another benefit, not shown in figure 4, is a reduction in simulation start-up time (the start-up times would dominate in the relatively short simulations performed for benchmarks, and have thus been excluded in the above plots so as not to give an unfair advantage to the parallel simulations). At the start of the simulation, the entire lattice must be passed in an operation with a cost proportional to the number of lattice sites. This process was never parallelized in the existing OpenMP parallelisation of the code and was completely serial prior to this work. However, in the MPI parallelized code each process only needs to load a much smaller lattice **reducing the setup time from 400 seconds to as low as 1.1 second when performing a simulation on 1024 cores.**

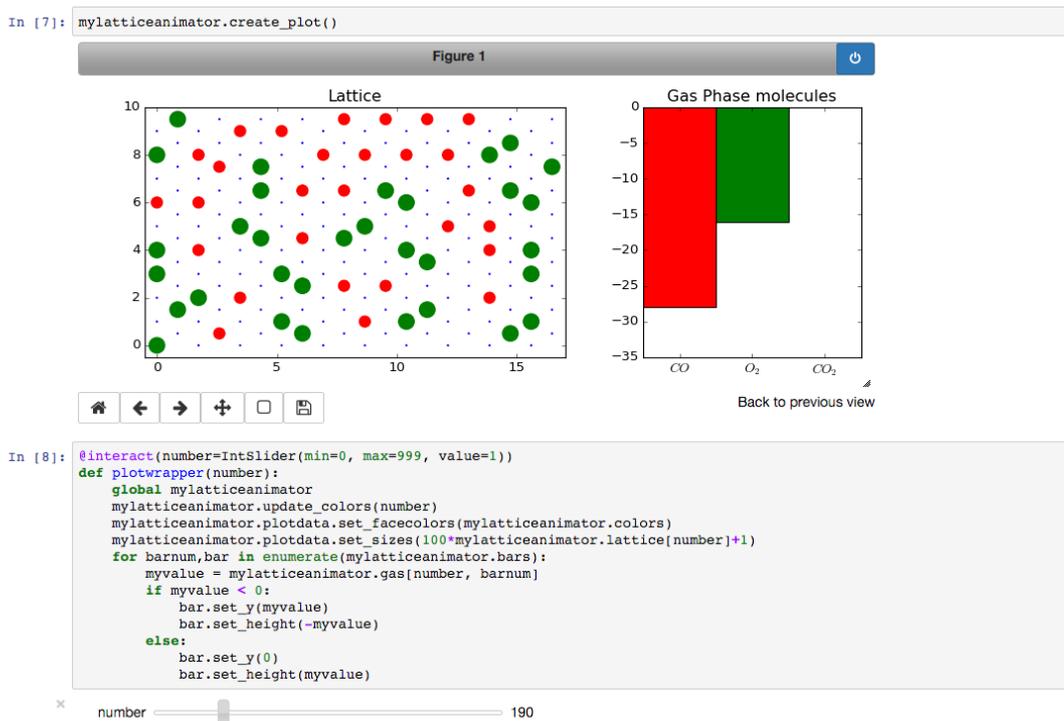


Figure 5 Example of animation of lattice in Jupyter notebook.

In addition to this work, we have made significant improvements to the usability of the code for work on compute clusters, particularly in the area of setup and configuration of the simulation, and reporting. It is now possible to fully

configure a simulation through easy-to-use command-line switches, which are included in results for traceability. The command-line switches allow us to easily override settings of the input files. We have developed Jupyter code for easy post-processing and graphical visualisation of results, allowing interactive step-through of simulations (see figure 5) which significantly facilitates the verification of the work performed, especially in relation to halo communication in the case of MPI simulations.

We do not have improvements in the serial performance to report; the objective of this proposal was to achieve correct spatial decomposition for the first time. The exact parallelisation does not provide any parallelism, and has been established to verify correctness of the parallel bookkeeping. The additional overheads of the necessary communication result, as expected, in a parallel slow-down as shown in figure 2.

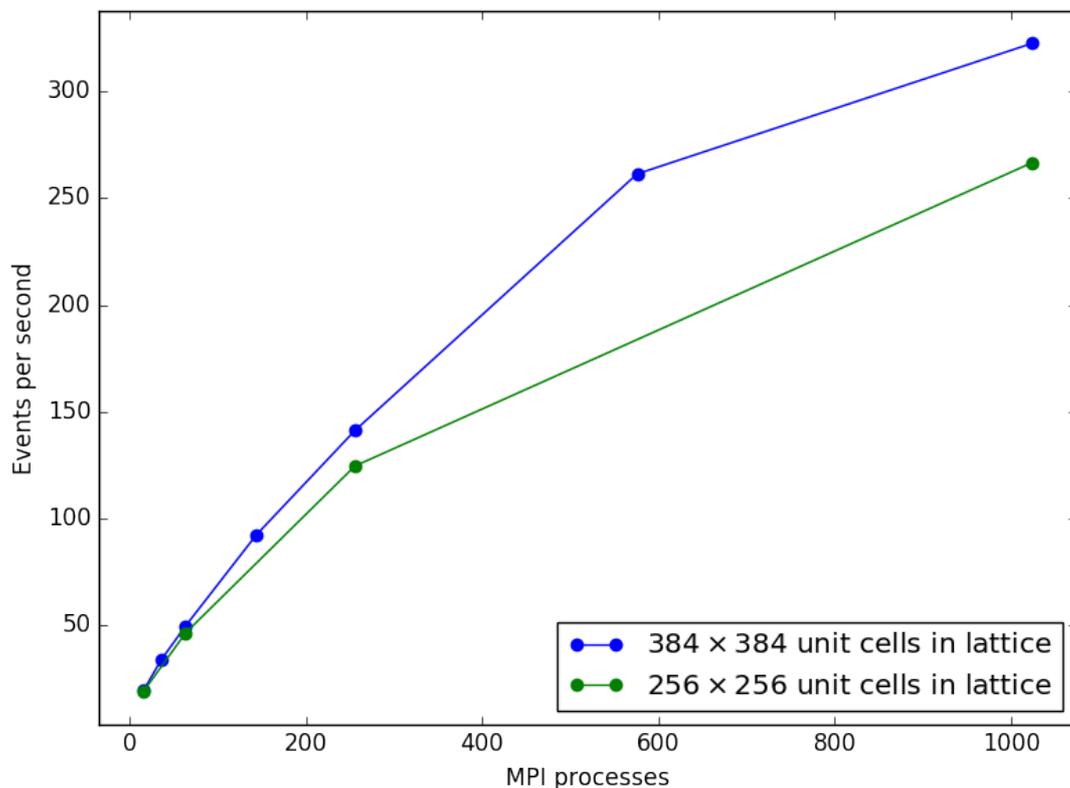


Figure 6 Strong scaling of the MPI code keeping the number of unit cells constant.

As shown in Figure 5, wall-clock time improvements for the approximate parallelisation are obtained up to 24 nodes for a moderate simulation domain. We anticipate further weak scaling, though we have not been able to quantify this beyond the range shown. In this particular simulation the sites occupancies of a halo of 4 unit cells belonging to the each of the neighbours must be known to correctly calculate and store the energy reactions on the lattice. This means that in the case of a 1024 MPI processes on a 256 x 256 unit cell grid (32 x 32 unit cells per MPI node) we must store 40 x 40 unit cells in each MPI node and the halos are $(2 \cdot 4 / (32 + 2 \cdot 4)) \cdot 100 = 20\%$ of the total stored lattice. Since the neighbouring MPI nodes must also know about a halo stretching into this this

domain any KMC event that involves 40 % of the lattice sites must involve communication of halo sites with the neighbouring domains making it clear why strong scaling breaks down. Figure 6 provides an alternative view of this, showing how the number of events per second increase as the lattice size per node grows.

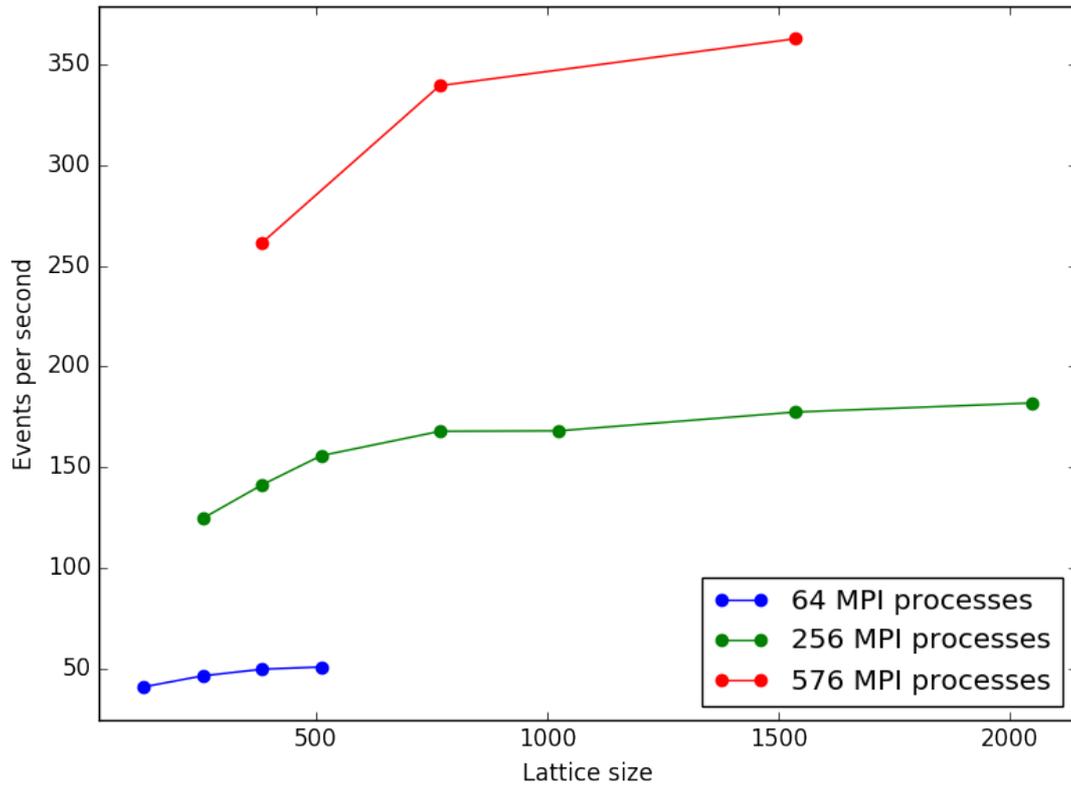


Figure 7 KMC events per second as a function of the lattice size for different number of MPI processes.

The principal achievement of this project is the successful delivery of a correct exact parallel implementation. As shown in Figure 7, consistent simulation results can now be obtained in a spatially decomposed domain.

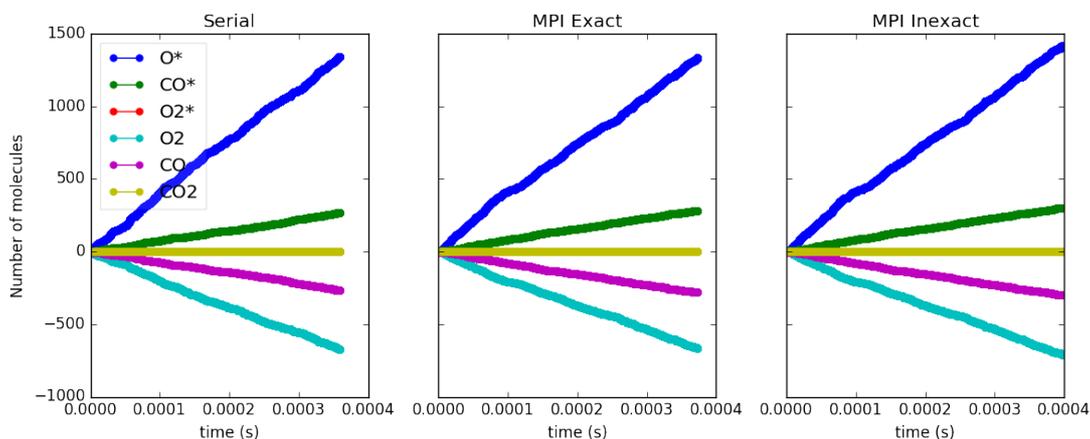


Figure 8 Comparison of output from simulations in serial and parallel. Negative numbers of molecules mean that a gas species reactant is being consumed.

5 Conclusion

Our eCSE project implemented MPI parallelization to address the challenge of the required long computational times when simulating large domains in catalysis- and surface-science-related KMC. We have implemented an exact and inexact algorithm for spatial parallelism. The exact algorithm does not provide any speed up but using the inexact algorithm we were able to gain a speed up of up to 23 times on 24 ARCHER nodes for a relevant benchmark example on a moderate sized lattice.

References

- (1) Stamatakis, M.; Vlachos, D. G. *ACS Catal.* **2012**, *2*, 2648-2663.
- (2) Piccinin, S.; Stamatakis, M. *ACS Catal.* **2014**, *4*, 2143-2152.
- (3) Lubachevsky, B. D. *J. Comput. Phys.* **1988**, *75*, 103-122.
- (4) Martínez, E.; Marian, J.; Kalos, M. H.; Perlado, J. M. *J. Comput. Phys.* **2008**, *227*, 3804-3823.
- (5) Ertl, G. *Angew. Chem. Int. Edit.* **2008**, *47*, 3524-3535.

Acknowledgment

This work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service (<http://www.archer.ac.uk>)