D3.3 Report on Instantiating Computing Patterns and performance measurements and prediction of HPMC Application

Due Date	Month 36
Delivery	Month 36
Lead Partner	UCL
Dissemination Level	Public
Status	Final
Approved	Internal Review
Version	1.0

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 671564.

DOCUMENT INFO

Date and version number	Author	Comments
27.09.2018 v1.0	Robin Richardson	

CONTRIBUTORS

Contributor	Role
Robin Richardson	Editor
Alfons Hoekstra	Internal reviewer
Hugh Martin	Internal reviewer
Saad Alowayyed	Contributed data
Olly Perks	Contributed data
David Wright	Contributed data
Olivier Hoenen	Contributed data
Arjen van Elteren	Contributed data

TABLE OF CONTENTS

1 Executive summary

This report presents the performance studies of the exemplar COMPAT applications, as carried out on multiple different HPC platforms, including those forming the Experimental Execution Environment (EEE). We consider various performance metrics, as may be most appropriate for the given application e.g. wall clock time, file sizes, scalability (strong and weak), and energy consumption (where available). In one weak-scaling application we also consider scalability of the middleware itself.

To further explore and assess the potential impact of extreme parallelism (believed to become important in the approaching exascale era of supercomputing), we also carried out performance studies on two applications beyond the EEE, using even larger supercomputers. To simplify the assessment, we note that the multiscale applications in COMPAT can be abstracted as sections of replica computing steps (either static in number or, in the case of the HMC pattern, dynamic) and large monolithic applications (such as, often, the most expensive model in the Extreme Scaling pattern). For this reason, our predictions as pertain to exascale resource usage largely come from detailed studies of application performance for a **Replica-based** exemplar (the Binding Affinity Calculator) and for an exemplar containing a large **monolithic** application (HemeLB).

Furthermore, a detailed mathematical model was developed to predict the time and length scales attainable by a lattice-Boltzmann solver (such as Palabos or HemeLB) in a fixed time on computers with e.g. 1 billion cores (exascale).

With respect to the exascale, a major conclusion of this work was that, even for those applications exhibiting excellent strong scaling characteristics, the trade-off between resolving time or physical length scales in the system will frequently render such simulations inefficient on enormous core counts when compared to the weak scaling (replica) case. We therefore expect that the actual impact of exascale resources on future science applications will be to encourage the use of uncertainty quantification (techniques that often require multiple runs) in a field where researchers too often only run large simulations once.

With respect to the instantiation of Multiscale Computing Patterns (MCPs) [1], we see how energy measurements (where available) can guide as to the best choice of supercomputer to send a given job to. Combined with the ComPat formalisms and performance models (reported in earlier deliverables) we now have a concrete path to using that data. As we move towards extreme parallelism, we argue that such cost efficient approaches will become essential.

2 Report

2.1 Largest scale performance tests

In this section, we detail the performance studies carried out on ComPat applications on the largest resources. These were typically tested on resources larger than that available under the Experimental Execution Environment. A weak scaling and a strong scaling application are considered, and some exascale predictions are formulated.

2.1.1 Binding Affinity Calculator (replica-based representative)

For multiscale applications following the replica computing pattern, it makes more sense to think in terms not of a single simulation, but rather of simulation campaigns. It is the orchestration of these campaigns that is key here, and therefore the choice and performance of the middleware is highly important, particularly as regards the efficient use of putative exascale resources.

In the first year of this project, the BAC was on the fast track and, with the giant, full-SuperMUC run, we demonstrated the feasibility of such enormous RC runs. In the second year, BAC was the application with which we developed and built the RC pattern, and we demonstrated that with RC we can aid in running BAC on non-trivial distributed environments. Finally, this year we want to demonstrate how BAC behaves on a single resource, with systematic studies of weak scaling. By adding up these three parts, we get the full picture whereby all ingredients are ready to go into production, using RC, the pilot jobs, and so on.

Our selected multiscale application demonstrating replica computing is the High Throughput Binding Affinity Calculator (HTBAC), which builds upon the RADICAL Cybertools (a middleware component of the COMPAT stack), as the framework solution to support the coordination of the required scale of computations, allowing the exploitation of thousands of cores at a time.

To determine the performance of HTBAC, particularly as regards the extension to extreme parallelism, a number of performance studies were carried out. The main resource used was NCSA Blue Waters, with additional runs on LRZ SuperMUC and ORNL Titan.

2.1.1.1 Scalability and resource usage

We explored the performance of HTBAC on NCSA Blue Waters with two different protocols:

1. ESMACS (Enhanced sampling of molecular dynamics with approximation of continuum solvent), consisting of 25 replicas, i.e. 25 pipelines
2. TIES (Thermodynamic integration with enhanced sampling) consisting of 13 lambda windows and 5 replicas, i.e. 65 pipelines

Both protocols run for a total of 6 ns simulation durations. ESMACS produces 3.5 GB/system (24 MB/ns) while TIES produces 10 GB/system (24 MB/ns). Each simulation step in TIES and ESMACS requires 32 cores. Protocols run approximately 10-12 hours, depending on the physical system and the number of timesteps provided by the user.

When considering an application following the replica computing (RC) pattern, the most pertinent performance property is that of weak scaling. This is also the most scientifically relevant property, as it demonstrates the ability of HTBAC to solve large number of drug candidates in essentially the same amount of time (as the resources increase).

To this end, in our first study we investigated the weak scaling behaviour when screening sixteen drug candidates concurrently using thousands of multi-stage pipelines on more than 32,000 cores on NCSA Blue Waters (we observed similar scaling on other platforms such as ORNL Titan for different protocols).

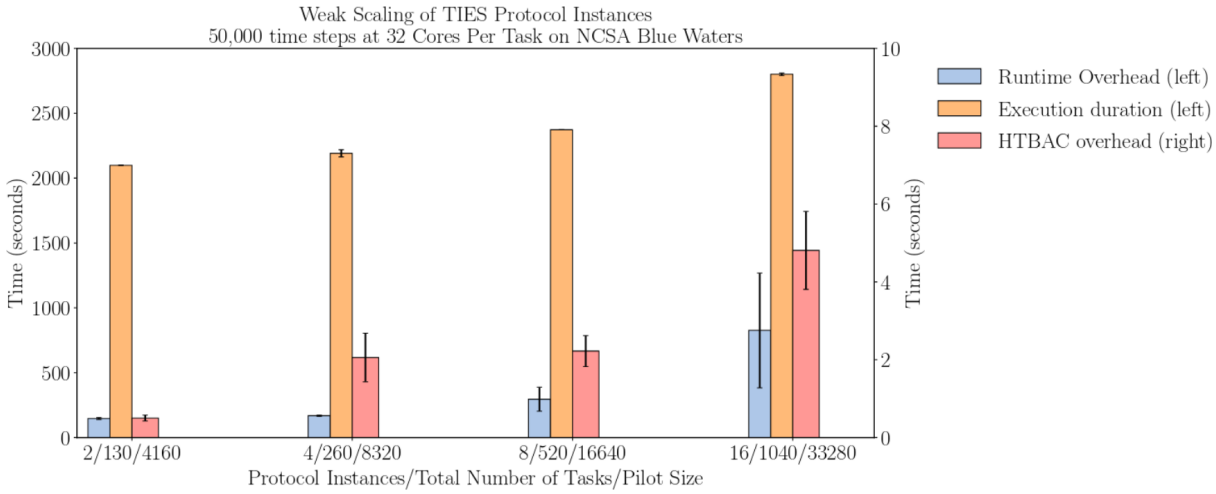

Figure 1: Weak scaling properties of HTBAC. We investigate the weak scaling of HTBAC as the ratio of the number of protocol instances to resources is kept constant. Overheads of HTBAC framework (right), and RCT overhead (left) and total execution time TTX (left) for experimental configurations investigating the weak scaling of TIES. We ran two trials for each protocol instance configuration. Error bars in TTX in 2 and 8-protocol runs are insignificant.

A detailed representation of the weak scaling performance of HTBAC for the TIES protocol is presented in Figure 1, demonstrating almost perfect scaling to hundreds of concurrent multi-stage pipelines.

In our second set of studies [2] we carried out a number of experiments on Blue Waters using both the ESMACS and TIES protocols. We present here the results of the weak scaling experiments:

ID	Type of Experiment	Physical System(s)	Protocol(s)	No. Protocol(s)	Total Cores
1	Weak scaling	BRD4	ESMACS	(2, 4, 8, 16)	1600, 3200, 6400
2	Weak scaling	BRD4	TIES	(2, 4, 8)	4160, 8320, 16640
3	Weak scaling	BRD4	ESMACS + TIES	(2, 4, 8)	5280, 10560, 21120

In Figure 2 we show (a) the weak scaling of HTBAC with the TIES protocol, (b) with the ESMACS protocol, and (c) with instances of both TIES and ESMACS protocols.

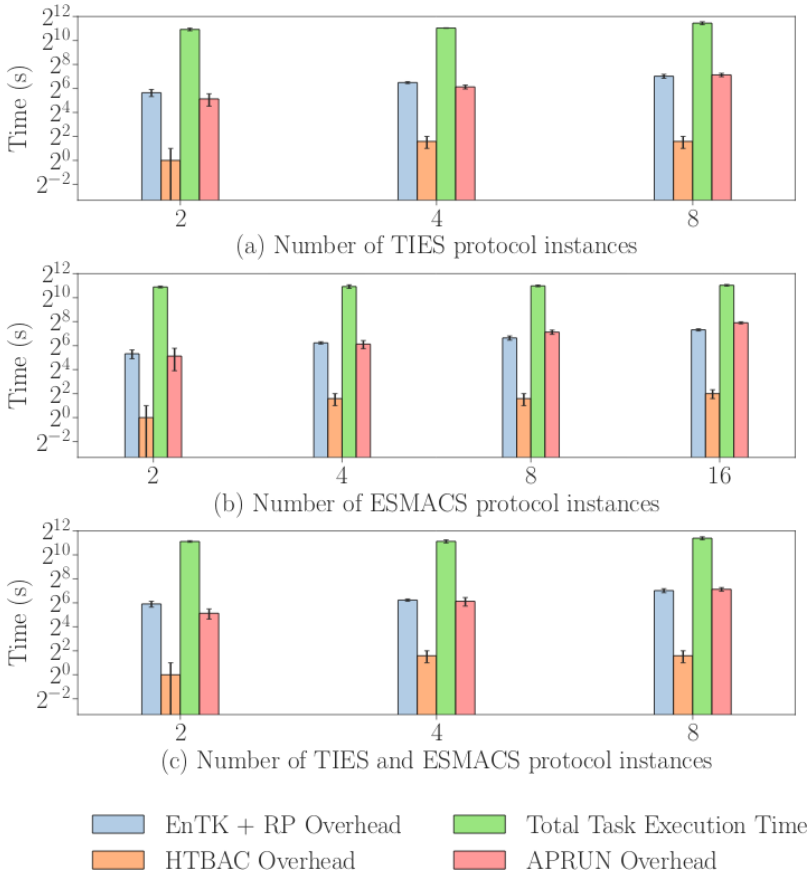

Figure 2. *Weak scaling of HTBAC. The ratio number of protocol instances to resources is constant. Task Execution Time with and HTBAC, EnTK+RP, aprun overheads with (a) TIES (Experiment 1), (b) ESMACS (Experiment 2), and (c) TIES and ESMACS (Experiment 3).*

For all weak scaling experiments (1–3) we used physical systems from the BRD4-GSK library (16 ligands made available for this work by GlaxoSmithKline) with the same number of atoms and similar chemical properties. The uniformity of these physical systems ensures a consistent workload with insignificant variability when characterizing their performance under different conditions.

In all weak scaling experiments (Figure 1 and 2) we observed minimal variation in the task duration as the number of protocol instances increases. We conclude that HTBAC shows near-ideal weak scaling behavior under the conditions tested. The overhead for the TIES results includes the adaptive sampling algorithms. The HTBAC overhead depends mostly on the number of protocol instances that need to be generated for an application. This overhead shows a super linear increase as we grow the number of protocol instances, but the duration of the overhead is negligible when compared to Total Task Execution Time.

This detailed performance data supplements and reinforces our earlier experiences of the excellent weak scaling of the BAC on large supercomputing platforms such as LRZ SuperMUC in 2016, in which both phases (a total of 250,000 cores) were used simultaneously for 37 hours, testing 50 candidate drugs and generating around 5 terabytes of data[1].

2.1.1.2 Node failure rate

The probability of node failures is likely to increase as supercomputers are constructed with ever larger numbers of nodes, and might therefore become significant on some exascale platforms. However, on the resources used for our performance measurements, we observed typically very few node failures, even when under high stress. During a campaign of 64 proteins, 25 replicas each, and 2-4 nodes per replica (executed on Blue Waters), only 2 node failures occurred (even though this campaign was executed twice). It should be noted that these two campaigns were executed shortly after Blue Waters came back online after a shutdown period, so the system may have been in a more stable state than after a long period of continuous usage. Nevertheless, there is little evidence on present systems (even when using hundreds of thousands of cores) that node failures will significantly impact the scalability of HTBAC in the short to medium term, although unforeseen issues might well arise on the e.g. billions of cores a full exascale machine may contain. This remains an active research topic, and in principle we understand, in the context of RC, how to deal with potential node failure in an automatic way. However, given this experiment we have not yet implemented automatic detection and recovery of node failures into the RC pattern. We intend, as larger machines come available, to continue running such huge campaigns to understand the actual node failures, and when needed, to realise fault tolerance and recovery mechanisms into the RC pattern.

[1] http://www.gauss-centre.eu/SharedDocs/Pressemitteilungen/GAUSS-CENTRE/EN/2016-03_SuperMUC_Pers_Med.html?nn=1290050

2.1.1.3 Conclusions and prediction for Exascale

Extrapolating from the promising weak scaling performance analysis presented above, we might expect good scaling of replica based applications at even greater node counts. Our studies have not yet shown any limitations that might preclude efficient use of exascale services. Differences in architecture and hardware may, naturally, affect this, and as the COMPAT stack matures, we will obtain more performance data to further clarify the viability of such applications on exascale machines.

As we demonstrated in deliverable D2.2 and D3.2, replica computing can also very well be executed in a distributed mode, running replicas across a range of supercomputers, with the multiscale computing patterns algorithms and software facilitating the detailed deployment. We continue to explore these capabilities, but this will require a production ready distributed supercomputing environment, such as the ComPat EEE. The European Supercomputing landscape would, in our opinion, very much benefit from such an environment, e.g. operated under the governance of PRACE. We have demonstrated that our middleware (QCG) is production-ready, and that our RC pattern is capable of exploiting such distributed HPC resources in a very efficient way. In combination with the weak scaling performance as reported in this deliverable, this would even allow us to reach the Exascale on a RC application by aggregating the power of sub-exascale machines. To conclude, ComPat has demonstrated that this is a viable option.

2.1.2 HemeLB (monolithic representative)

The Experimental Execution Environment did not have sufficiently large resources for determining large scale monolithic runs. As we have argued in Deliverables D2.1 and D3.1, and demonstrated in D2.2 and D3.2, the primary models in Extreme Scaling patterns are large scale monolithic codes. We have already demonstrated how the multiscale computing patterns algorithms and software can efficiently deploy Extreme Scaling applications on the EEE. The next step is to study in detail if and how the primary models themselves can scale to the largest HPC machines currently available to us.

We therefore used ARCHER (up to 96k cores) and Blue Waters (up to 300k cores) for our largest runs. The ARCHER supercomputer in Edinburgh, UK is a Cray XC30, with dual 12-core Intel Xeon E5-2697v2 (Ivy Bridge) 2.7 GHz processors joined by two QPI links, connected via a proprietary Cray Aries interconnect in a dragonfly topology. The Blue Waters supercomputer in Illinois is a Cray XE6/XK7 system consisting of more than 22,500 XE6 compute nodes (each containing two AMD Interlagos processors, with 8 floating point cores each).

2.1.2.1 Scalability and resource usage

Unlike the Replica Computing case explored earlier in this report, the most scientifically relevant scaling for such a monolithic application was determined to be strong scaling. While weak scaling would allow (physically) larger systems to be simulated in the same time (on more cores), the characteristic time scales of processes of interest typically scale as a power (greater than or equal to 1) of the system size, and thus aiming for constant wall clock time would not yield scientifically useful results.

Instead, we focussed on how a system of fixed size might be simulated faster through the use of more cores (on the same supercomputer). Our test system was the circle of Willis, an important vascular system located at the centre of the brain (and a region in which many aneurysms form). Such a system can already be simulated using (coarser) finite element methods, but we use it here as a useful geometry for benchmarking. On EPCC ARCHER, we benchmarked with a 15 micrometre resolution geometry (777 million fluid sites), and on NCSA Blue Waters we used a 7 micrometre resolution geometry (5.5 billion fluid sites). Note that in both cases the geometries are highly sparse (<< 1% fluid fraction), posing challenges for load decomposition as compared with a dense geometry.

The results of the performance measurements on ARCHER are shown in Figs. 3 and 4 [3]. The profiling of the code was carried out using the parallel performance tool, Scalasca (http://scalasca.org/).

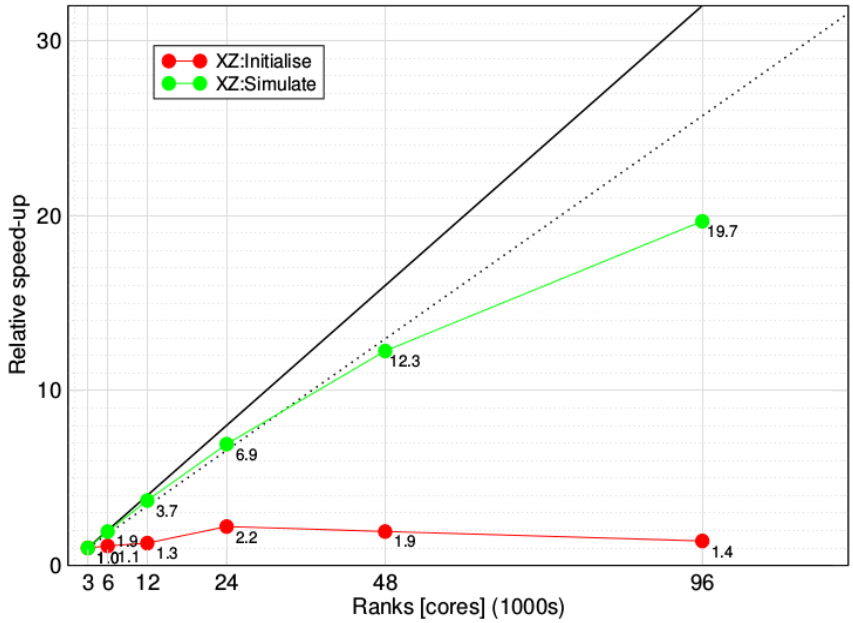

Figure 3: Strong scaling of HemeLB up to 96k cores on EPCC ARCHER, showing both initialisation and simulate phases.

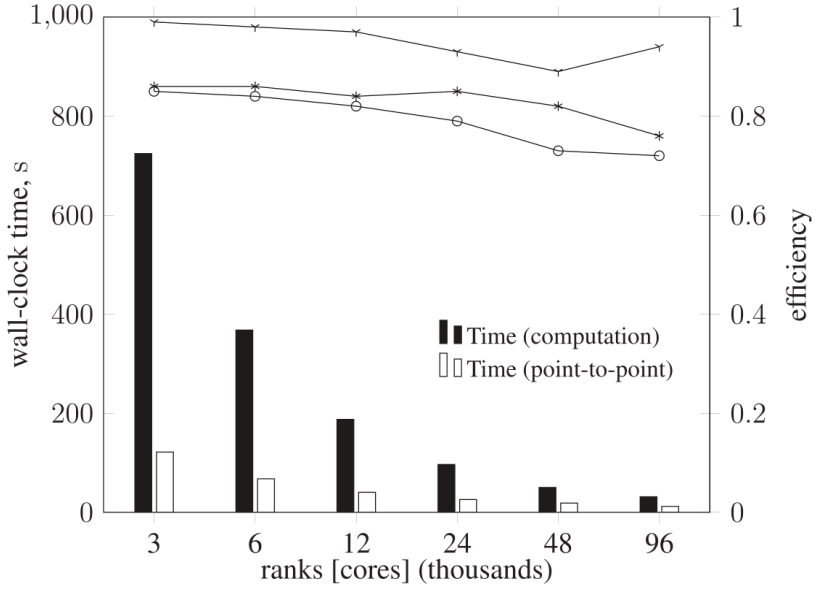

Figure 4: Wall clock time and efficiency metric for strong scaling of HemeLB on EPCC ARCHER, up to 96k cores.

In Figure 3 we see the speed-up of HemeLB from 3000 cores to 96000 cores, while Figure 4 shows the corresponding measured wall-clock time, and measure of parallel efficiency.

There is a negligible amount of MPI collective communication, and the amount of non-blocking point-to-point communication for data exchange decreases in proportion to computation time. Therefore, communication efficiency remains above 0.89. Load balance, however, starts at 0.86 and progressively deteriorates to 0.76, such that the overall parallel efficiency degrades to 0.72 once at 96,000 cores.

In our second study, we performed benchmarking of HemeLB on NCSA Blue Waters up to 300000 cores, using a higher resolution system (with approximately double the number of fluid sites). At such a high number of cores and low fluid site count per core (approximately 5000 sites per core) it was more challenging to avoid overheads from the use of a profiling tool such as Scalasca, so we focussed only on wall clock time per run. The resultant performance data is given in the following table:

Table 1: Data for strong scaling study on Blue Waters plotted in Figure 5.

# cores	# nodes	wall clock time (simulate phase) [s]
16000	1000	3490.868
32000	2000	1799.434
64000	4000	0942.717

128000	8000	0494.497
256000	16000	0376.673
300000	32000	0557.471

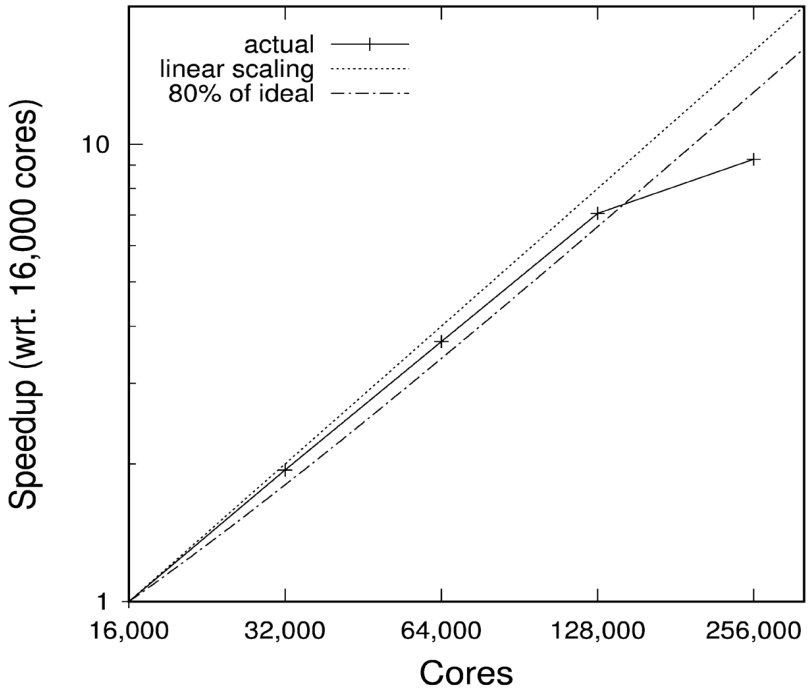

Figure 5: Strong scaling of memory-optimized HemeLB on Blue Waters, up to 256k cores, for a 5.5 billion fluid site circle of Willis geometry.

In Figure 5, we see the results of the strong scaling on of HemeLB on Blue Waters, shown here up to 256k cores. The performance degradation thereafter is attributed to significant load imbalances (due to the difficulty of minimising the communication surface in such a complex, sparse geometry) and the very low computational load per core (5000 sites on average).

It was unfortunately not possible to obtain energy usage information from ARCHER or Blue Waters (they do not make this information available to users).

2.1.2.2 Node failure rate

Node failure rate on BW was low, even at 300k cores, although on exascale machines this is expected to be a significant issue - monolithic applications will be particularly vulnerable to this. Similarly on ARCHER we found a negligible node failure rate under normal conditions - however: when running multiple OOM jobs over the whole system, subsequent jobs appeared to fail on the released nodes.

2.1.2.3 Conclusions and prediction for Exascale

Our monolithic test application used in the above performance analysis shows very good strong scaling for the given system sizes. However, due to the locality of interactions in the lattice-Boltzmann formulation (and hence locality of information transfer) the challenges of efficiently exploiting extreme parallelism will likely lie not so much in the simulation phase - a larger or higher resolution input file may always be used - but rather in the creation and initialisation of such enormous input files, and the *physical* time scales one may reach (given that processor speed increases little). The focus here on strong scaling is precisely due to this practical need for parallelism to increase physical time evolution in the system (rather than merely allow physically larger or higher resolution systems) but a more intelligent performance model must take into account the trade-off between the spatial and temporal scales, and the combinations allowed at different core counts.

To this end, such a performance model has been developed for lattice-Boltzmann simulations (soon to be published by [4]) the results of which are shown in Figure 6.

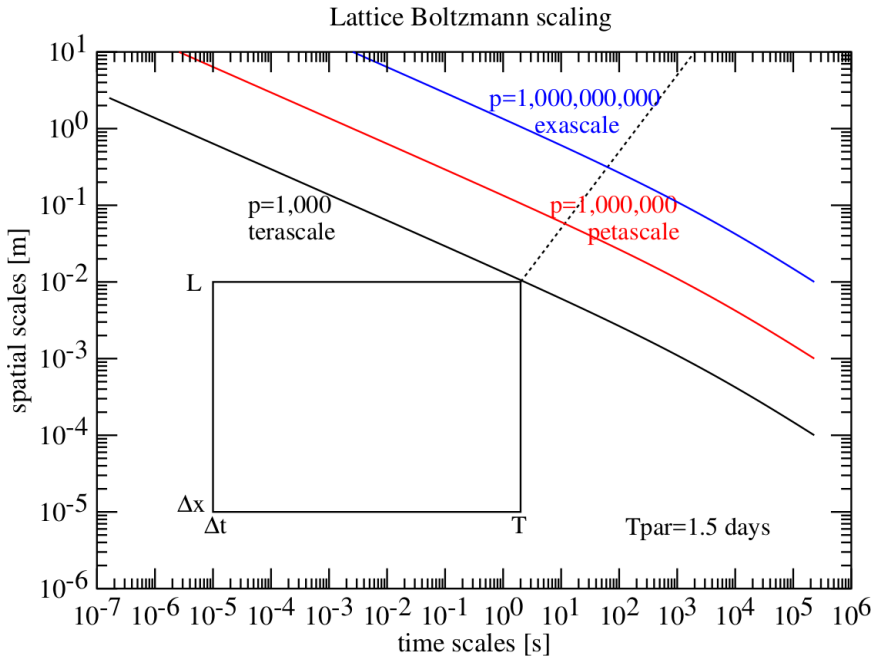

Figure 6: Reachable spatial and temporal scales for a lattice-Boltzmann simulation at 10 micrometre resolution, given a fixed 1.5 days of calculation time, for cores ranging from 1000 (terascale) up to 1 billion (exascale).

In Figure 6, we see the performance prediction using typical lattice-Boltzmann model parameters (in this case for Palabos, but equally applicable to HemeLB), showing the achievable time and spatial scales (at 10um resolution) achievable on 1k, 1M and 1G cores (the latter representing exascale).

The above has so far considered only the simulate phase of the lattice-Boltzmann application. However, as system sizes increase, the initialisation time (during which the load decomposition of the system occurs, and ranks read the relevant parts of the geometry into their memory) will also increase. The initialisation phase in the Blue Waters benchmark simulations varied (approximately) from 15 to 40 minutes, with more cores corresponding to longer initialisation.

2.2 Performance of multiscale applications on the Experimental Execution Environment

We now detail performance studies on the Fusion and In-stent restenosis (2D and 3D) applications, which were run on the experimental execution environment (EEE). These studies have been run on comparatively smaller numbers of processors than the applications listed in the previous section, and are unlikely to act as predictors of these applications' performance at the exascale. Nevertheless, the data collected is valuable for informing expectations on petascale machines, as well as (especially in the case of Fusion) showing the potential issues with energy consumption at higher core counts.

2.2.1 Fusion

The strong scaling of the Fusion application across 4 different systems (SuperMUC thin and fat nodes, Eagle and Neale) is presented in Figure 7. Measurements of energy consumption on 2 systems (SuperMUC and Eagle) are also provided in Figure 8.

The Fusion application is composed of 4 submodels, three of which (A,B,D) are serial and the last of which is parallel (C). The submodels are running synchronously (in this version), and we loop over this chain to iterate in time at the macro level i.e. (A->B->C->D)->(A...). Only a few iterations are executed in this benchmark. The parallel submodel C is composed of 8 flux tubes (or annulus) covering a sublayer of the 3D physical space, and each flux tube is discretized with a medium sized grid (128x128x32). Submodel C as its own internal time evolution (micro) for which results are averaged before being returned to macro level (D,A). Compared to its execution standalone, submodel C requires additional collective operation (MPI_Bcast) to send updated input data from process 0 (receiving from MUSCLE) to others. This is sub-optimal, but we currently want to treat the submodel as a black-box. The performance shown in these figures concerns submodel C (the parallel submodel) only.

For the system considered, the strong scaling levels off around 2k cores. Interestingly, the energy consumption seems to increase significantly at this core count, having previously remained relatively

steady up to 1k cores. This may be a result of excessive communication costs for the problem size, or may be indicative of significant increases to come at higher core counts.

From Figure 8 we can conclude that (in this case) Eagle is providing the same run time at a lower energy cost. Our execution plans in the MCP software [1] might therefore use this information to decide to run on Eagle for subsequent jobs of this nature.

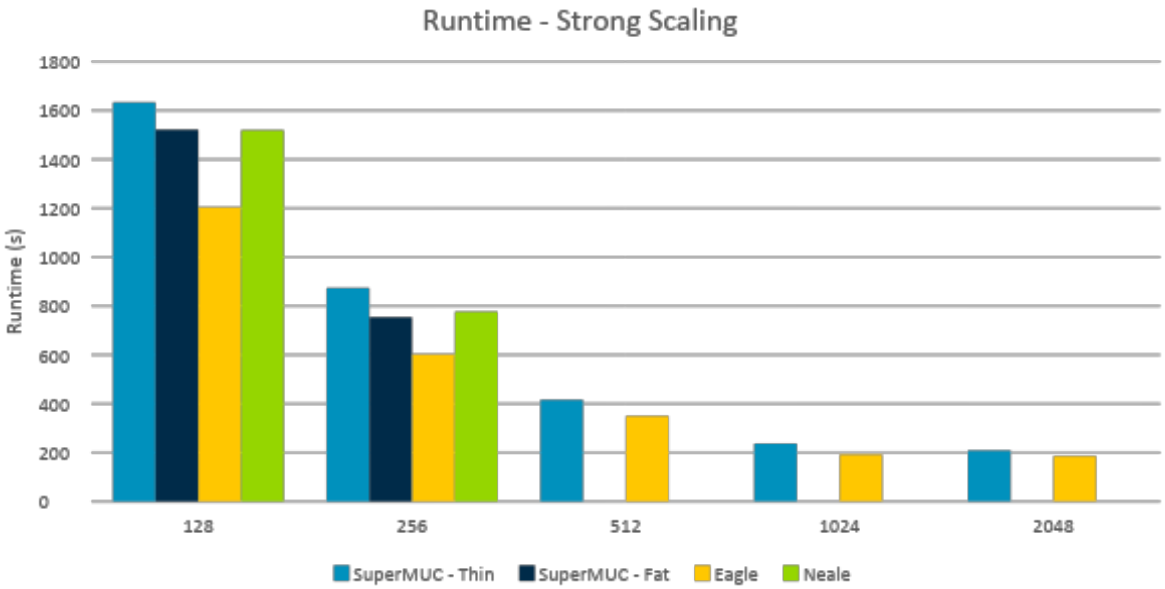

Figure 7: Wall-clock time to completion for Fusion application on 4 different systems: LRZ SuperMUC (thin and fat nodes), PSNC Eagle and STFC Neale

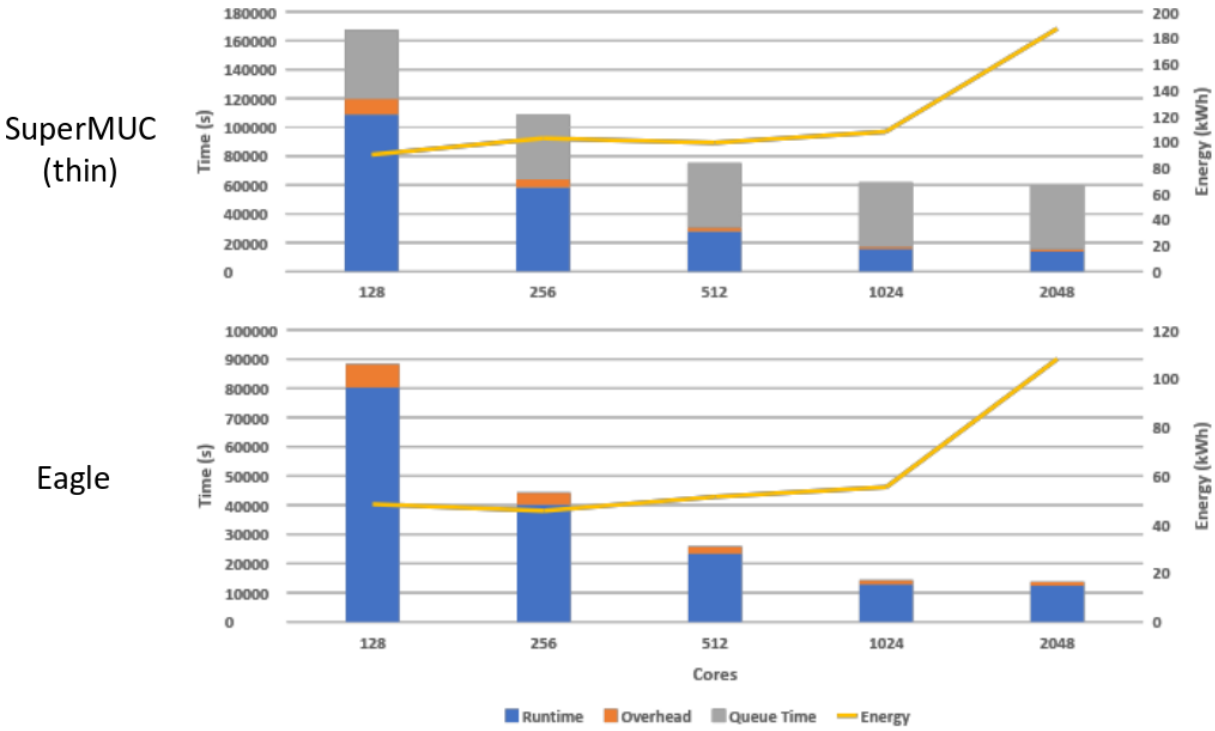

Figure 8: Fusion application energy to solution, strong scaled on 2 different systems (LRZ SuperMUC and PSNC Eagle).

2.2.2 ISR 2D and 3D

Performance data was collected for strong scaling of the 2D and 3D In-stent restenosis applications (ISR2D and ISR3D, respectively) on the LRZ SuperMUC thin nodes. In both cases, a vessel of length 1.5 mm and width 1.2 mm, and 1400 cells was simulated. The performance and energy consumption of the 2D application is shown in Figs. 9 and 10 respectively. The performance of the 3D ISR application is shown in Figure 11. The interpretation of these results is currently still ongoing.

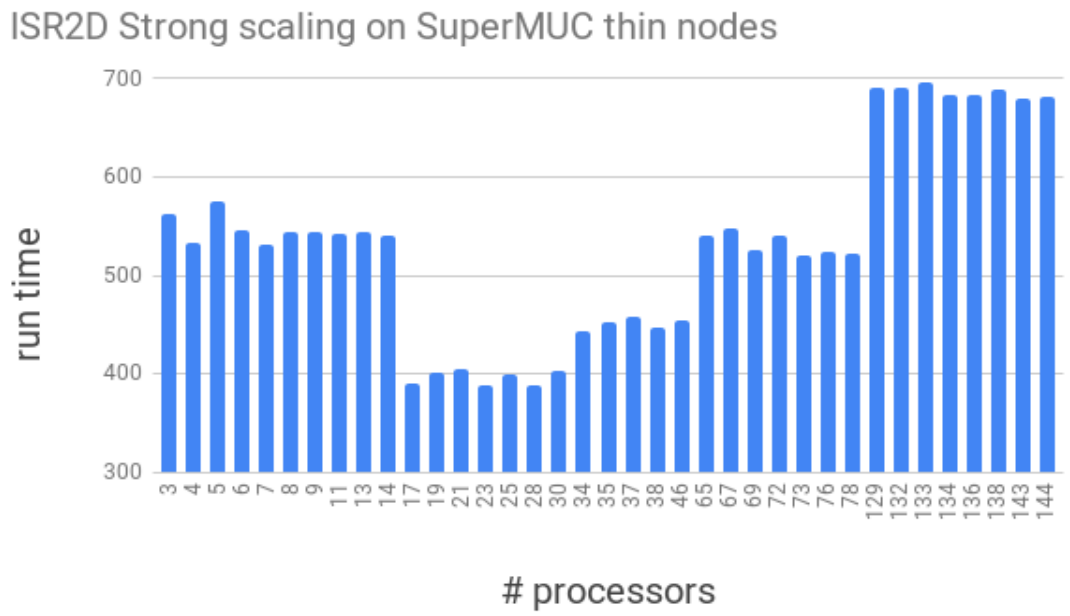

Figure 9: Strong scaling run time measurements for ISR2D on LRZ SuperMUC's thin nodes

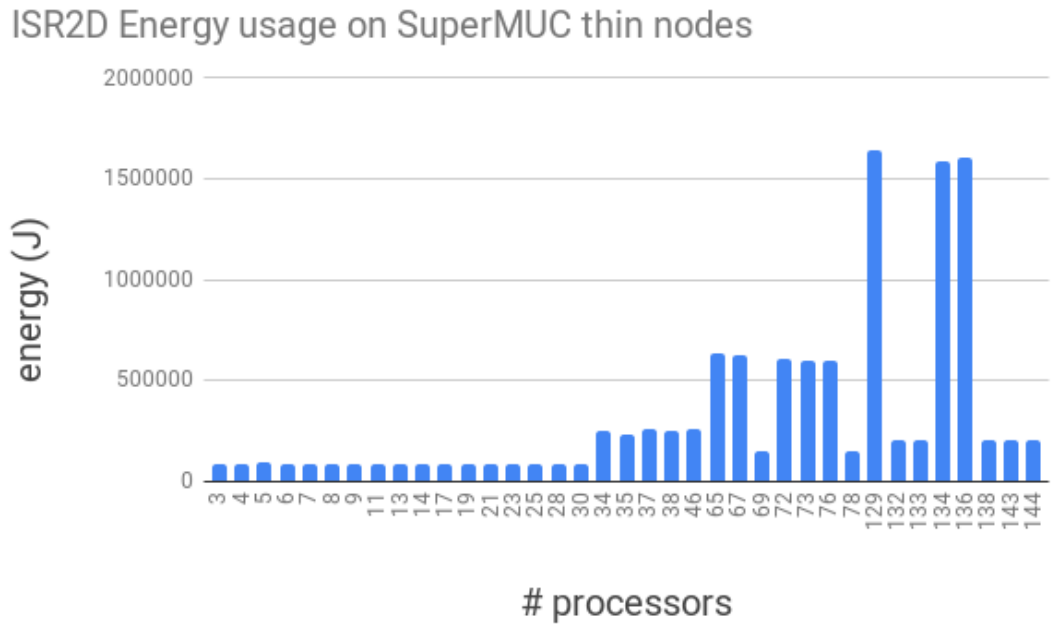

Figure 10: Energy measurements on LRZ SuperMUC's thin nodes for ISR2D runs shown in Figure 9.

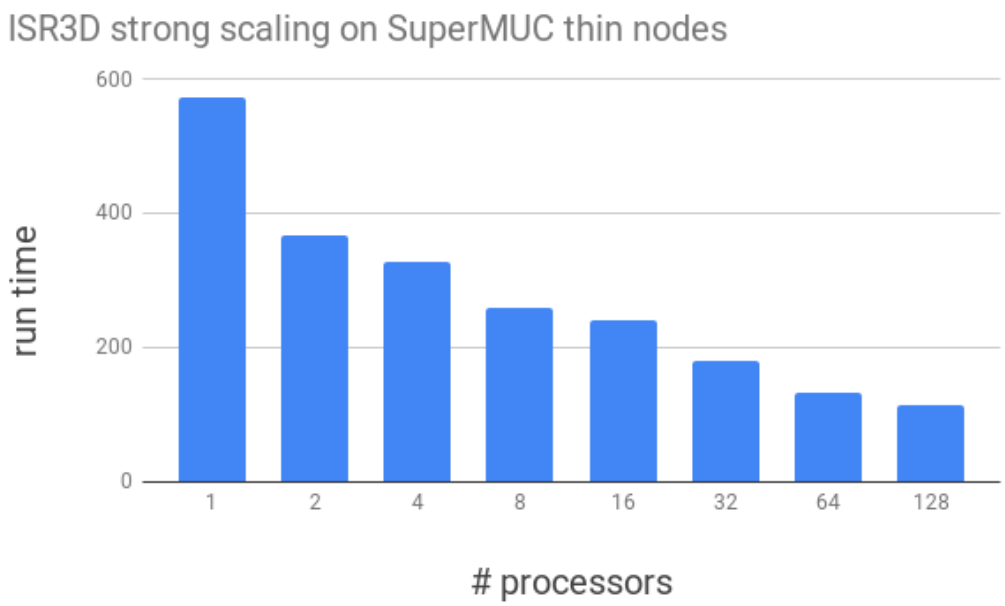

Figure 11: Strong scaling run time measurements for ISR3D on LRZ SuperMUC's thin nodes.

2.2.3 Astrophysical application

The performance of the hierarchical astrophysical model (which follows the HMC pattern) was explored through the simulation of the evolution of planetary systems in their birth cluster. The cluster consisted of 10,000 or 20,000 stars. A simple strong scaling test showed a speedup of 232.1 on 500 nodes with respect to 2 nodes (185.62 s and 43082.94 s wall-clock time respectively). The results of a weak scaling study on the same system are presented in Figure 12**.** For the weak scaling, a different number of planetary systems (each containing 20 planets) orbit the stars so that each node was fully employed.

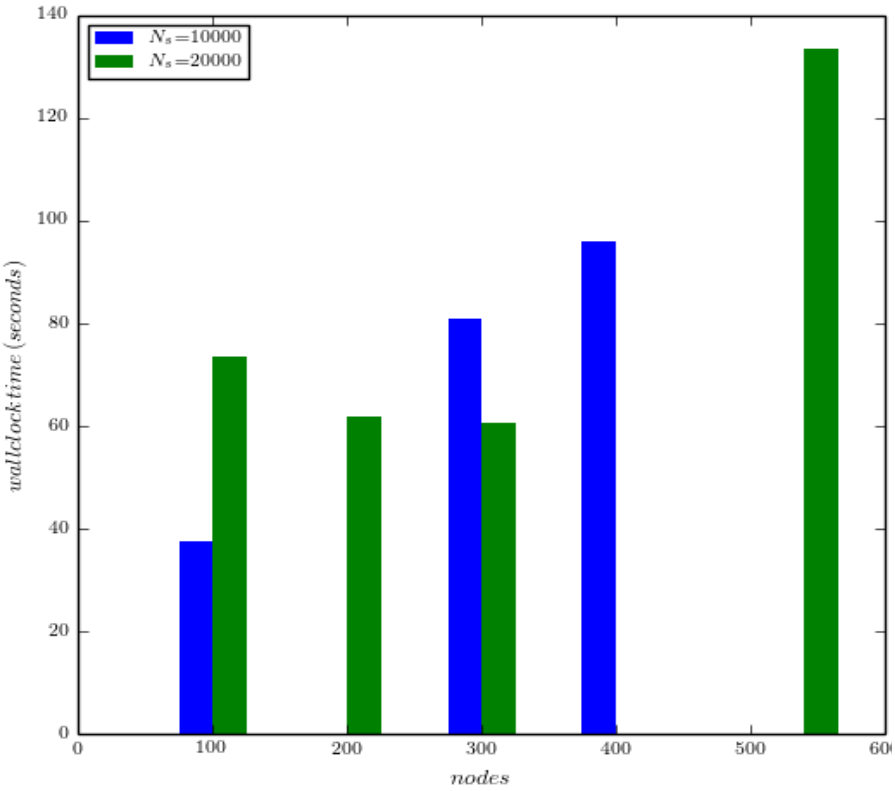

Figure 12: Weak scaling run time measurements for the hierarchical astrophysical model on Cartesius, for clusters of 10000 and 20000 stars.

2.2.4 Towards hybrids of Replica Computing and Extreme Scaling

These two applications are examples of Extreme Scaling, and if we would then need to execute say an Uncertainty Quantification on top of these, where we would easily be running some thousand copies of the ES application (so, resulting in a Replica Computing of an Extreme Scaling application, or a hybrid pattern) this would strongly amplify the need for computational resources. Note that for the two-dimensional version of the In-Stent Restenosis application this has already been demonstrated [5]. Already for this relatively small ISR2D application we needed a substantial amount of resources in order to perform the Uncertainty Quantification (order days, running on 128 processors). We are now in the process of scaling this up to the 3D version of the model. This work is going to be carried out in the recently started FET-HPC VECMA project, leveraging on and using ComPat technology.

3 Conclusions

3.1 Discussion

The performance work on the Binding Affinity Calculator provides a good indication of how Replica Computing (RC) pattern applications are likely to scale under extreme parallelism.
This includes, to a large extent, the Heterogeneous Multiscale Computing (HMC) pattern, for which the greatest computational cost generally lies in the execution of replicas. However, as we have elaborated in deliverable D2.2, the dynamic nature of HMC, depending on the quality of the surrogate model to capture accurately enough the microscale dynamics, makes this less obvious then for the pure RC patterns. I might well be that exascale performance is required in phase 1of HMC (the initial training of the surrogate) after which HMC applications could resort to reduced resources. This remains a topic of research. In the materials HMC application (UCL), for example, the similarity between microsimulations is determined in parallel by the primary model (which is mostly a Finite Element Solver), but the costs are dwarfed by those of running the many submodels (replicas). In the less common case of an HMC pattern with a very expensive macroscale (primary) model, the performance on exascale will likely more closely follow that of the Extreme Scaling pattern. It is our opinion that both RC and HMC type of applications are viable candidates for exascale computing, as we have demonstrated on several occasions in ComPat. However, we have only been able to 'scratch the surface', and more research and demonstrators are required to substantiate these conclusions.

In the case of applications following the Extreme Scaling (ES) pattern, the performance of the primary model will be of most interest at levels of extreme parallelism. The strong scaling performance studies of HemeLB were carried out to test this aspect. Prediction work carried out by Chopard *et al.* [4] indicated (for a general lattice-Boltzmann application) the expected attainability of physical and temporal scales with putative exascale systems (1 billion cores) of order 100 s for 10 cm (or 10 s and 100cm) after 1.5 days of wall clock time. Depending on the problem size, it may be more efficient to run at lower resolutions, while using several replicas, in order to derive uncertainties from the simulations. Again, we believe that ComPat has demonstrated that ES can scale to the exascale, if the primary model is capable of extreme scaling *and* if the auxiliary models that may serialize the execution, are deployed in an efficient way, maybe even by staging two independent ES runs. This was discussed and demonstrated in deliverable D2.2 and D3.2.

While more restricted in terms of system sizes considered, the performance of the Fusion application (strong scaling levelling off at 2k cores) would appear to support the thesis that uncertainty

quantification (via many replica simulations) will likely be a more efficient use of exascale resources. Performance studies on the ISR2D and ISR3D applications also point to the use of UQ as a good way forward. As pointed out above, in the recently started VECMA project we are going to proceed to create hybrids of ES applications (such as Fusion or ISR) and RC, to facilitate Uncertainty Quantification. In VECMA we will also explore more advanced 'semi-intrusive' Uncertainty Quantification algorithms, which can be mapped to the HMC pattern. In other words, in VECMA we will also be exploring hybrids between ES and HMC, when implementing the semi-intrusive Uncertainty Quantification algorithms for ES multiscale applications.

3.2 Impact of exascale resources on future science applications

The major conclusion of this work is the excellent performance of replica based calculations to extreme parallelism supercomputing. While this is directly applicable to RC or HMC applications in phase 1 of the performance cycle, the efficient use of ES applications may need more careful treatment (depending on the physical and temporal scales of the process of interest). One way will likely be to respond to the growing desire for uncertainty quantification (UQ) in computational science, leveraging the efficiency of replica computing on exascale systems by running several ES applications in the same allocation. This will have the benefit of rendering the simulation output "actionable", in the sense that UQ will give users more faith in the applicability of the results and thus greater ability to make decisions thereon. Additionally, Replica Computing is typically more resistant to node failures.

We therefore expect that the actual impact of exascale resources on multiscale computing is likely to be to encourage the use of replica based computing patterns (RC or HMC), and quantifying uncertainties in larger simulations (such as the primary model of the ES pattern) which is not feasible with present day petascale facilities.

4 References

[1] Alowayyed, S., T. Piontek, J. L. Suter, O. Hoenen, D. Groen, O. Luk, B. Bosak et al. "Patterns for high performance multiscale computing." *Future Generation Computer Systems* (2018). (https://doi.org/10.1016/j.future.2018.08.045)

[2] "Concurrent and Adaptive Extreme Scale Binding Free Energy Calculations" Dakka et al., 2018 (https://arxiv.org/abs/1801.01174)

[3] Patronis, A., Richardson, R. A., Schmieschek, S., Wylie, B. J., Nash, R. W., & Coveney, P. V. (2018). Modelling Patient-Specific Magnetic Drug Targeting within the Intracranial Vasculature. *Frontiers in physiology*, 9, 331.

[4] (I*n press*) Chopard, Hoekstra and Coveney, 2018, *Phil Trans A*

[5] Nikishova A, Veen L, Zun P, Hoekstra AG. 2018 Uncertainty Quantification of a Multiscale Model for In-Stent Restenosis. *Cardiovasc. Eng. Technol.* 1–14. (doi:10.1007/s13239-018-00372-4)