Exploiting cloud computing for algorithm development

Neil S. O'Brien, Steven J. Johnston, Elizabeth E. Hart,
Kamal Djidjeli, Simon J. Cox
Faculty of Engineering and the Environment
University of Southampton, UK.
Email: nsob104@soton.ac.uk

September 13, 2011

Abstract

We consider the application of cloud computing to the process of algorithm development. We introduce a case study focusing on the development of a novel algorithm in computational electromagnetics, illustrating several challenging areas for algorithm developers where cloud-based architectures can deliver enhanced productivity and potentially save costs. The development, verification and tuning of our algorithm have all been assisted by cloud-based technologies. Our preliminary results both demonstrate the potential of the algorithm to solve the problems accurately, and of cloud-based architectures to accelerate the development and verification process. We propose that cloud-based architectures will in the future play a greater role in the development of algorithms; saving costs by improving hardware utilisation, and reducing turnaround time.

1 Introduction

The development of new algorithms can be both time- and computationally-intensive, and often involves large volumes of data. Recent developments in cloud computing provide cost-effective, on-demand access to massive amounts of computational resources, services, software and data; making cloud-based architectures potentially very useful for algorithm developers. Here we demonstrate the applicability of cloud computing for algorithm development, using the development and tuning of a novel algorithm for computational electromagnetics as a case study. The results of our novel meshless method are found to be in good agreement with the well known plane wave expansion method (PWEM) [1], with an average relative error better than 1%. By augmenting the development cycle with cloud-based resources we achieved significant development time savings.

The paper is structured as follows: we provide some background to cloud computing, highlighting its strengths in four key areas of computational science in section 2. In section 3 we discuss the applicability of cloud computing to the challenges of developing novel algorithms. We present our case study in section 4, first giving the background to the application area in section 4.1, introducing our novel algorithm in section 4.2, and illustrate the results obtained with cloud computing in section 4.3. Section 5 provides a summary of the work.

2 Cloud computing

Cloud computing is a technology in which customers are able to access computational resources, software, data and services via the Internet as and when they are required. There are a variety of commercial cloud providers including Amazon, Google and Microsoft [2]. It is possible to provision resources from a cloud provider rapidly, on the order of minutes, and the minimum rental period is very short (currently an hour). This is often called a utility pricing model, as it is similar to the way we purchase other utilities such as electricity. Within the constraints of available funds, it is possible to provision large quantities of resources for the desired time. Cloud providers offer customers a variety of hardware specifications, providing capability appropriate to the task at hand. Applications utilising cloud-based architectures can be scaled both up (by using larger machines) and out (by using more machines), all without the capital expenditure associated with purchasing hardware [3].

Cloud computing providers are able to benefit from economies of scale both by bulk-buying hardware, and by optimising their administration practises. Combining this with flexible pricing models that allow customers on-demand access to resources means that the cloud provider is able to achieve higher data centre utilisation than in house solutions. Customers do not pay for under-utilised machines but rather return them to the provider when not in use [4]. This is in stark contrast to running a local data centre where hardware, once purchased, is not usually returnable, and at times of low demand often sits under-utilised and consuming energy.

Cloud-based architectures provide benefits in four key areas, as detailed below [2]:

- **Algorithm development**
  Using cloud-based architectures in algorithm development benefits from the on-demand procurement of appropriate types of hardware. It is possible for the developer to rent a large, highly-resourced computer in order to be able to carry out a high-resolution simulation that would not be

practical on the desktop; or to rent a large number of computers to check the performance of the algorithm with increasing parallelism. Where test or validation datasets are sufficiently large, cloud-based architectures can reduce algorithm development time by providing large quantities of compute resources located close to the data. Utilising cloud-based architectures encourages a modularity in design, so that alternative algorithms may be swapped in for comparison purposes.

- **Data dissemination**
  With traditional computational science, sharing large data sets can involve the slow and expensive process of copying large volumes of data, sometimes to multiple locations. Cloud computing services, however, are highly available, globally accessible, and benefit from large bandwidths. To share scientific data with collaborators in the cloud can be as simple as setting appropriate access permissions. The data can also be exposed to third-party cloud-based software, allowing them to perform independent analysis at their own expense. In the future it would not be unreasonable for publications to be required to make their underlying data sources public [5], providing readers with the ability to re-calculate results and confirm findings. This kind of analysis could also be supported by cloud-based solutions, as demonstrated by the emergence of data marketplaces [6].

- **Burst capability**
  A traditional data centre can be sized to cope with predictable maximum demands, but unpredictable peak demands can leave an application under-resourced, adversely impacting the user experience. Scaling a data centre to cope even with predictable peak demands can lead to large amounts of hardware spending much of its time idle. Cloud-based solutions can be scaled quickly, either to increase resource at peak times, or to remove under-utilised resources during times of low demand. With traditionally purchased hardware, the former is slower and the latter is difficult to achieve at all.

- **Super-scalability**
  It is difficult to estimate the demands that will be placed on an application and therefore the resources it will require. Cloud-based applications can be scaled beyond the size that a typical data centre may successfully host, whilst also delivering the advantage that customers only pay for the resources needed at a given moment. Cloud-based applications can scale beyond alternative solutions, whilst potentially saving costs.

There has been considerable work carried out on various aspects of cloud computing for scientific applications. One paper focused on the use of a particular astronomy application, Montage [7], and concluded that cloud environments could provide good compute-time performance, but when the runtime is short, delays in resource scheduling and wide-area communication can become significant. Others have included cost-effectiveness in their analyses, e.g. [8], and concluded that a cloud computing service provides a feasible and cost-effective model in many scientific application areas. On the issue of cost-effectiveness, it must be noted that the comparison depends heavily upon the specifics of the application. However, other investigators concluded that the cloud is not yet mature enough for traditional HPC-type computations which require MPI, due to the relatively slow interconnect (commercial cloud computing providers do not generally support low-latency networking such as Infiniband or Myrinet) [9]. Consideration has also been given to utilising the cloud as an extension of a local cluster, concluding that scheduling strategies are key to achieving good value for money [10]. In this case, the local cluster may "overflow" work to the cloud when demand is high, reducing queue times and increasing quality of service. Such a hybrid approach, with appropriate job scheduling techniques, could be superior to an approach exclusively based on one or the other technology, since there are cases (e.g. jobs which work on sensitive data that must be kept locally) which are not well-suited to cloud-based architectures [11].

Hazelhurst [8] compares an Amazon EC2 "c1.medium" cloud compute instance with a local, dedicated cluster. Accounting for the purchase price of a machine for the local cluster, and comparing it to Amazon's pricing, he estimates that if the utilisation of a local machine is above 10-50% it is more cost effective to have the local machine. In our case, there would be large idle periods during times when the algorithm is being changed, implying a cloud-based solution may be more effective. Additionally, the comparison does not account for the electricity, space, air conditioning, UPS systems, operating system licensing and support, and administration requirements for a local cluster – which would, if considered, push the balance in favour of the cloud-based solution (whose pricing includes these overheads).

However, there is currently only limited literature [2] on the applicability of cloud-based architectures to algorithm development, testing and characterisation with representative case studies, and this is the focus of our paper. In Section 3 we describe how the cloud-based architecture fits into the algorithm development process. This application has a particular demand profile, which involves potentially long periods when the developers are writing and improving code – during which a dedicated cluster would be a wasted resource – followed by the need to test the revised algorithm, sometimes against large input data sets or for many values of the tunable parameters. This makes the cloud an ideal technology, appealing to its burst capability as detailed above.

3 Applicability of cloud computing to algorithm development

We applied Microsoft Windows Azure to this case study, although the principles that we demonstrate are applicable across other cloud providers. Workers are the building blocks of an Azure-based solution; each consumes messages from a queue, completes the work described in the message, and outputs results to

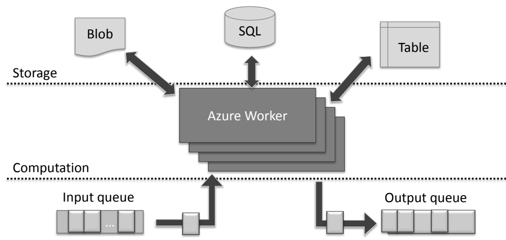

Figure 1: Cloud worker architecture pattern: a worker consumes jobs from a queue and writes intermediate output and final results to blob storage, allowing dynamic numbers of workers to process messages. Messages are also written to the output queue if required, facilitating further processing by other workers.

storage or to a different queue, as shown in Figure 1. Windows Azure workers provide a Windows operating system and basic libraries such as the .Net runtime to run user applications. The process of provisioning an Azure worker involves building a virtual machine, allocating hardware, booting the operating system and starting the user-defined application code; this is managed by the Azure fabric. In our experience it takes 15–30 minutes to provision an Azure worker. At times when new revisions of the algorithm were ready for testing and characterisation it was not necessary to re-provision the worker. Instead we could halt the code, load the new revision, and start it executing, which typically took under five minutes. Upgrading workers is also managed by the Azure fabric.

Azure also provides data storage in the form of blob storage. This is highly scalable and designed to be robust. It supports key-based access, so that it is easy to provide multiple users with access to intermediate results stored here during the verification and tuning process, or to final results once the algorithm is known to be working correctly. In cases where the development of the algorithm, or the provision of input data sets, or analysis of output data, is a highly collaborative activity requiring input from teams around the world, this is a clear advantage for cloud-based technologies. Their high bandwidth and availability obviates the need to copy large amounts of data between institutions whilst providing large amounts of compute capability near to the data.

Our cloud-based architecture fits into the development cycle whenever we wanted to check that the current revision of the algorithm could produce accurate results, or assess the effect of various parameters on the accuracy and runtime of the algorithm. In order to achieve this, we ran a parameter sweep with Azure (details of the parameters for our particular algorithm are given in Section 4.2).

We set up a scalable cloud-based architecture, in which we placed each combination of parameters of interest in a message which was submitted to the input job queue. We provisioned workers which take messages off the queue and run the algorithm with the specified parameters, storing the output (the results from the algorithm, accompanied by logging and performance data) in blob storage. The software running on the workers consisted of the Windows Server 2008R2 operating environment that Azure provides, with a custom worker process to access the job queue and call the algorithm via a command-line interface, capturing its output and writing that to blob storage. The algorithm itself was a Windows executable with supporting libraries such as a vendor-optimised LAPACK implementation. This modularity of the worker design facilitated easy updates of the algorithm when these were required. Moreover, treating each worker as an independent piece of computation in this way reduces the need for inter-worker communication, thus helping to avoid some of the communications latencies encountered with dependent nodes [7].

The architecture we used is illustrated in Figure 1. As shown, all the Azure workers have access to read job messages from the input queue and can write output to blob storage and, if required, an output queue to facilitate further processing by additional workers. The workers also have access to SQL database storage and can use this for logging success and failure messages as well as performance data. Our workers for all but the very highest resolution simulations were "small" instances, providing a 1.6GHz CPU and 1.75GB RAM (our highest resolution simulations required more RAM; to avoid excessive paging we used larger instances with 3.5GB RAM).

In our case study we applied four workers initially, and at times supplemented those with 20 additional workers, on a separate Windows Azure account. With appropriate permission, the data in cloud storage is globally accessible by any Azure worker. In this way, collaborators could provide resources financed from different budgets or institutions, all of which would independently consume messages from the same queue and write results to blob storage. The cloud provides the ability to analyse the effects of changes to the algorithm rapidly, obviating the need to either wait for many hours in a cluster's job queue, or to have expensive hardware sitting idle whilst development takes place, and has the potential to significantly reduce the development cycle time. In the case study, we were able to run 24 instances of our algorithm simultaneously, achieving significant wall-clock time savings compared to the alternatives. With more workers, the process could have been further accelerated, so that in this architecture, the overall wall-clock time for a parameter sweep is bounded below by the sum of times taken to run the longest individual simulation and to provision the worker on which it runs.

4 Case study: a novel method in electromagnetics

In this section we introduce our case study, focusing on the development of a new algorithm for computational electromagnetics. We begin by providing a brief introduction to the application area of photonic crystals (PhCs), and give an overview of the new algorithm that we developed with cloud technologies.

We then go on to show a subset of our results, which demonstrate the ability of the algorithm to produce accurate results. We also investigate some of the factors affecting the accuracy of the algorithm. These results were obtained from a parameter sweep carried out on Azure.

4.1 Photonic crystals and the Maxwell equations

A photonic crystal (PhC) is a periodic dielectric structure [12]. Dependent upon the geometry of the crystal, it may possess useful behaviours based upon its ability to block certain wavelengths of electromagnetic radiation. One-dimensional examples were first studied by Lord Rayleigh in 1887 [13], and there has been renewed interest in PhCs since 1987 [14, 15]. Because of the expense of fabricating such structures, it is imperative that we are able to accurately simulate their behaviour.

The Maxwell equations are the equations that govern the behaviour of electromagnetic waves propagating in the structure. In this example, we concentrate on solving the Maxwell equations in 2D. We assume the crystal is periodic along the x and y directions, and homogeneous in the z direction; and we make standard simplifying assumptions [12], under which the Maxwell equations may be reduced to the form in (1) and (2). The first equation is for the TM (transverse magnetic) polarisation, in which the magnetic field is confined to the xy plane, and the second the TE (transverse electric) polarisation, in which the electric field is confined to the xy plane:

$$-\frac{1}{\varepsilon}\Delta\psi = \lambda\psi, \tag{1}$$

$$-\nabla \cdot \frac{1}{\varepsilon}\nabla\psi = \lambda\psi, \tag{2}$$

where λ is the spectral parameter, ε is the dielectric constant, and in two dimensions we have $\Delta = (\partial^2/\partial x^2, \partial^2/\partial y^2)$. The Bloch-Floquet wavefunction is ψ [16].

4.2 A novel meshless algorithm for photonic crystal modelling

There exist several solution approaches for the Maxwell equations, including the PWEM (plane wave expansion method) and the FEM (finite element method) [17]. However, these methods have some potential pitfalls, and meshless methods offer a promising alternative [18].

Existing meshless methods rely on a computationally-intensive weak form to solve TE modes. This is necessary because the dielectric ε is discontinuous, yet is differentiated in (2). For TM modes, no derivative of $1/\varepsilon$ enters (1), and therefore a strong form method, with considerably less computational cost, is used.

The novel method that forms this case study is formulated for TE modes using the weak form method in the vicinity of the discontinuity, where it is necessary; and combining it with a strong form method further from the discontinuity. It is thus a hybrid meshless local weak-strong form method (MLWSFM). This delivers an overall saving in computational cost, because the use of the weak form is reduced.

The algorithm has two distinct stages: it constructs several system matrices and combines them appropriately to form a generalised eigenvalue problem; it then uses a vendor-accelerated LAPACK routine to find the generalised eigenvalues, which are the useful outputs of the algorithm. Dependent chiefly upon the resolution, the code required between several MB to around 2GB of RAM in the resolution range we investigated. The typical run times ranged from seconds at the very low resolutions, to a few hours for the highest resolutions. The output data are very small, consisting of the generalised eigenvalues at each of sixteen points in the reciprocal lattice ($\sim 250\,\mathrm{kB}$ for a typical resolution).

The method has several tunable parameters. It uses a compactly-supported radial basis function (CSRBF) and currently supports a choice of several such functions, including the C2 and C4 functions of Wu [19] and Wendland [20]. Additional tunable parameters are the CSRBF shape parameter c, the extent of the weak-form domain δ, and the numbers of nodes n_{N} and background cells n_{BG}.

4.3 Cloud computing applied to meshless algorithm development and verification

Cloud computing assisted in several ways with the development and verification of this algorithm:

- It was first necessary to validate the algorithm against results calculated by the PWEM code. In this respect, Azure provided the ability to rent a large, capable machine with a lot of RAM to run a single instance of the new method, so that the simulation could be run at a high resolution to check that the results match expectations within reasonable tolerance. Initial results were available very soon after they were calculated, so that we could inspect them and verify that the process was proceeding as intended.

- To assess the impact of each of the tunable parameters of the algorithm – such as the resolution, the specific CSRBF used, and the balance of strong form to weak form nodes – we carried out a parameter sweep across a high dimensional space.

- It will be necessary to use the new algorithm to model several different PhC geometries that have been reported in the literature and have been solved with existing methods. This will allow a better characterisation of the overall accuracy of the method.

- Once the algorithm's accuracy is established there will be interest in simulating shapes whose characteristics are not already known; cloud-based

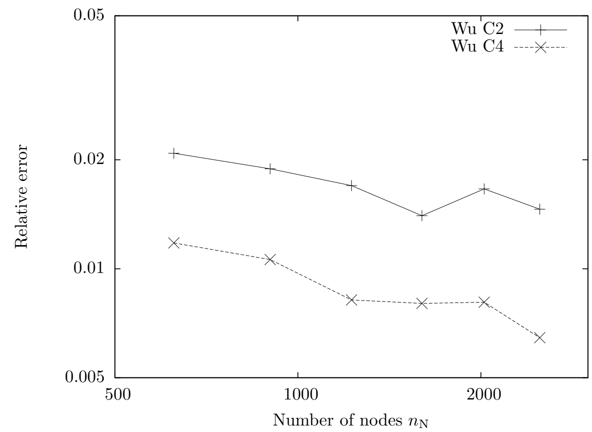

Figure 2: Average relative error of the MLWSFM compared to PWEM as a function of number of nodes and background cells $n_{\mathrm{N}} = n_{\mathrm{BG}}$ for two different CSRBFs (Wu C2 and Wu C4, see text).

architectures will facilitate this. If it is desired to simulate a large variety of shapes at once, as would be the case in a bandgap optimisation exercise [21], we could provision many worker nodes for a relatively short period of time. We note that these nodes would still perform independent calculations, so inter-node communications delays should not be a disadvantage in this application. Were it necessary to perform very high resolution calculations to verify the best-case results, it would also be possible to provision a machine with large memory for just the time taken to run the required simulations.

4.4 Preliminary results

The preliminary results that we present here were calculated using Windows Azure with the architecture detailed in section 3. We simulated one of the classic examples in the domain of PhC modelling, the case of round rods on a square lattice, with radius $r = 0.2a$, where a is the lattice constant of the PhC. The dielectric of the rods was $\varepsilon = 8.9$ and the surround was air ($\varepsilon_{\mathrm{air}} = 1$).

In Figure 2 we show the average relative error for the novel MLWSFM for two different CSRBFs, Wu's C2 and C4 functions [19], compared to a well-known PWEM [1]. The abscissas are values of n_{N}, the number of nodes, and we took the number of background cells $n_{\mathrm{BG}} = n_{\mathrm{N}}$. We fixed values of $c = 0.5$ and $\delta = 0.3$. The trend as expected is for the accuracy to increase with increasing numbers of nodes. We speculate that the cases where there is a deviation from this trend may be caused by the increasingly ill-conditioned nature of the eigenvalue problem as the resolution increases [22]. Accuracies better than 1% were achieved for several resolutions.

In Figure 3 we have illustrated the effect of two of the other parameters, c

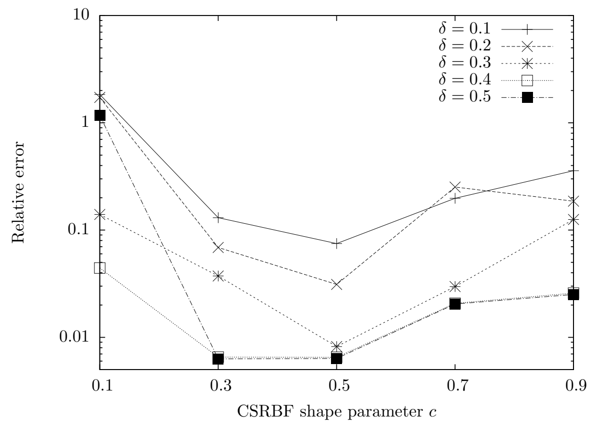

Figure 3: Average relative error of the MLWSFM compared to PWEM as a function of CSRBF shape parameter c and weak-form extent δ.

and δ, for a fixed number of nodes and background cells $n_{\mathrm{N}} = n_{\mathrm{BG}} = 1225$. The shape parameter giving best overall accuracy was $c = 0.5$ but for $\delta \geq 0.4$, $c = 0.3$ also gives rise to very small errors. Accuracies of better than 1% were achieved with $c = 0.5$ and $\delta \geq 0.3$, and also for $c = 0.3$ with $\delta \geq 0.4$. These results are in good agreement with previous work on the role of the shape parameter in meshless methods on a periodic domain, which found $c = 0.5$ to be the optimum [23]. The larger values of delta correspond to using the weak form method for a larger proportion of the domain. Weak form methods usually have better accuracy than the strong form [24] so this is also expected behaviour.

In Figure 4 we compare band diagrams [12] generated by the MLWSFM and the PWEM. The band diagram shows the frequencies of the four lowest-frequency modes that may propagate through the crystal. The agreement between the MLWSFM and the PWEM can be seen to be good. Taking $\delta = 0.3$ and $c = 0.5$, we found the MLWSFM to be faster than the previous meshless method [18] that uses the weak form alone, by around 11% for $n = n_{\mathrm{BG}} = n_{\mathrm{N}} = 400$, and 8% for $n = n_{\mathrm{BG}} = n_{\mathrm{N}} = 900$ on account of the reduced amounts of computationally-intensive numerical integration required by the new method.

5 Summary

In this paper we have outlined the benefits of cloud computing for algorithm development, and have illustrated the involvement of cloud-based architectures with a case study in photonic crystal modelling. The development of algorithms is typically a serial task but when it is necessary to verify a new algorithm by running simulations at high resolution or with large input or output data sets, the cloud can provide short term rental of appropriate hardware which can

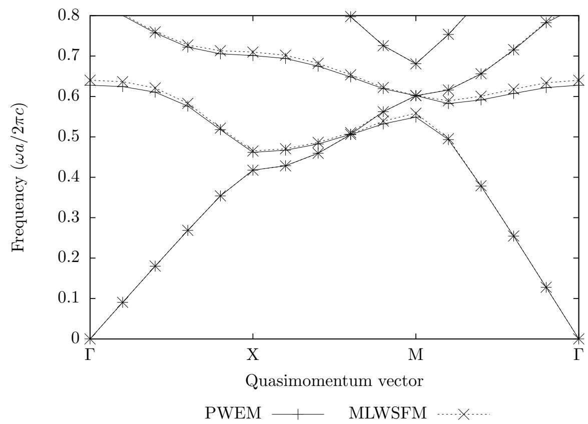

Figure 4: Comparison of band diagrams calculated by the PWEM and ML-WSFM.

significantly accelerate the verification process. When the dependence of an algorithm's accuracy upon multiple parameters is to be analysed, renting many cloud workers for a short period can provide considerable speedup at a moderate cost.

Although it is difficult to accurately compare the TCO of cloud-based architectures to traditional clusters, the utilisation of a dedicated cluster would be low while changes are being made to the algorithm; with a cloud-based solution and utility pricing, we avoided paying for idle computers, which gives the cloud the potential to deliver a better cost-benefit ratio than a dedicated cluster.

Our results demonstrate that our algorithm can accurately model the response of a photonic crystal, and illustrate the ability of the cloud to accelerate algorithm development and verification. We propose that as time goes on, cloud computing will play an increasingly important role in the development of algorithms.

Acknowledgements

We gratefully acknowledge funding from Microsoft.

References

[1] S. G. Johnson and J. D. Joannopoulos, "Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis," *Opt. Express*, vol. 8, no. 3, pp. 173–190, 2001.

[2] G. Aloisio and S. Fiore, Eds., *Grid and Cloud Database Management*. Springer, 2011, ch. Scientific computation and data management using Microsoft Azure.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R., A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," *Technical report*, February 2009, UCB/EECS-2009-28.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010, 10.1007/s13174-010-0007-6. [Online]. Available: http://dx.doi.org/10.1007/s13174-010-0007-6

[5] S. J. Coles, J. G. Frey, M. B. Hursthouse, M. E. Light, A. J. Milsted, L. A. Carr, D. DeRoure, C. J. Gutteridge, H. R. Mills, K. E. Meacham, M. Surridge, E. Lyon, R. Heery, M. Duke, and M. Day, "An E-Science Environment for Service Crystallography–from Submission to Dissemination," *J. Chem. Inf. Model.*, vol. 46, no. 3, pp. 1006–1016, 2006.

[6] Microsoft Corporation, "Windows Azure Marketplace DataMarket," http://www.microsoft.com/windowsazure/Whitepapers/DataMarket/, 2010.

[7] C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, and J. Good, "On the use of cloud computing for scientific workflows," in *eScience, 2008. eScience '08. IEEE Fourth International Conference on*, dec. 2008, pp. 640 –645.

[8] S. Hazelhurst, "Scientific computing using virtual high-performance computing: a case study using the amazon elastic computing cloud," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, ser. SAICSIT '08. New York, NY, USA: ACM, 2008, pp. 94–103. [Online]. Available: http://doi.acm.org/10.1145/1456659.1456671

[9] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?" in *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*, ser. UCHPC-MAW '09. New York, NY, USA: ACM, 2009, pp. 17–20. [Online]. Available: http://doi.acm.org/10.1145/1531666.1531671

[10] M. D. de Assuncao, A. di Costanzo, and R. Buyya, "Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters," in *Proceedings of the 18th ACM international symposium on High performance distributed computing*, ser. HPDC '09. New York, NY, USA: ACM, 2009, pp. 141–150. [Online]. Available: http://doi.acm.org/10.1145/1551609.1551635

[11] D. Chappell, "Windows HPC Server and Windows Azure," 2011, David Chappell and Associates.

[12] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, *Photonic Crystals: Molding the Flow of Light*, 2nd ed. Princeton University Press, 2008.

[13] Lord Rayleigh, "On the maintenance of vibrations by forces of double frequency, and on the propagation of waves through a medium endowed with a periodic structure," *Phil. Mag.*, vol. 24, pp. 145–159, 1887.

[14] S. John, "Strong localization of photons in certain disordered dielectric superlattices," *Phys. Rev. Lett.*, vol. 58, no. 23, pp. 2486–2489, Jun 1987.

[15] E. Yablonovitch, "Inhibited spontaneous emission in solid-state physics and electronics," *Phys. Rev. Lett.*, vol. 58, no. 20, pp. 2059–2062, May 1987.

[16] P. A. Kuchment, *Floquet Theory for Partial Differential Equations.* Birkhäuser Basel, 1993.

[17] D. H. Beckett, S. J. Cox, J. M. Generowicz, B. P. Hiett, M. Molinari, and K. S. Thomas, "Application of finite element methods to photonic crystal modelling," in *Computation in Electromagnetics, 2002. CEM 2002. The Fourth International Conference on (Ref. No. 2002/063)*, 2002, p. 2 pp.

[18] E. E. Hart, S. J. Cox, and K. K. Djidjeli, "Compact RBF meshless methods for photonic crystal modelling," *J. Comp. Phys.*, vol. 230, no. 12, pp. 4910 – 4921, 2011.

[19] Z. Wu, "Compactly supported positive definite radial functions," *Advances in Computational Mathematics*, vol. 4, pp. 283–292, 1995.

[20] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, December 1995.

[21] J. Goh, I. Fushman, D. Englund, and J. Vučković, "Genetic optimization of photonic bandgap structures," *Opt. Express*, vol. 15, no. 13, pp. 8218–8230, 2007. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-15-13-8218

[22] G. E. Fasshauer, *Meshfree approximation methods with MATLAB.* World Scientific Puslishing, 2007.

[23] E. Hart, S. Cox, K. Djidjeli, and V. Kubytskyi, "Solving an eigenvalue problem with a periodic domain using radial basis functions," *Engineering Analysis with Boundary Elements*, vol. 33, no. 2, pp. 258 – 262, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2N-4SVD1B1-1/2/e1572ec01a8edd8b06cafabb8c67f827

[24] G. Liu and Y. Gu, *An Introduction to Meshfree Methods and Their Programming.* Springer, 2005.