DISTRIBUTED BUILDING ENERGY SIMULATION WITH THE HLA

Kwabena Amponsah
Sameh Zakhary
School of Computer Science
University of Nottingham
Wollaton Rd, Nottingham, NG8 1BB, UK
{kwabena.amponsah, sameh.zakhary}
@nottingham.ac.uk

Darren Robinson

School of Architecture
University of Sheffield
Arts Tower, Sheffield, S10 2TN, UK
d.robinson1@sheffield.ac.uk

Paul Nathanail

Land Quality Management Ltd.
Sir Colin Campbell Building
Triumph Rd, Nottingham, NG7 2TU, UK
paul@lqm.co.uk

Brian Logan
Peer-Olaf Siebers
School of Computer Science
University of Nottingham
Wollaton Rd, Nottingham, NG8 1BB, UK
{brian.logan, peer-olaf.siebers}@nottingham.ac.uk

ABSTRACT

The primary aim of this work is to demonstrate the feasibility of applying the High Level Architecture (HLA) framework to the distributed execution of physical urban building energy simulations that are too large to be supported within the resource constraints of a single computing node. To this end, we modify the building energy simulation tool CitySim+ to make it HLA-compliant, and use this to conduct distributed computational experiments over a small network of computing nodes. For our experiments, we consider some real-world urban models containing up to about 120,000 building surfaces. In addition to demonstrating the feasibility of our approach, the experiments further provide some performance measures to help us evaluate the appropriateness of applying the HLA in the urban simulation context. From the results of the experiments, we briefly discuss the trade-offs between the computation time and the communication and synchronization overheads involved in our distributed building energy simulation.

Keywords: High Level Architecture, HLA, Building Energy Simulation, Distributed Simulation.

1 INTRODUCTION

The primary aim of this work is to demonstrate the feasibility of applying the High Level Architecture (HLA) framework to the distribution of large scale physical building energy simulations over multiple computing nodes. To this end, we develop an HLA-compatible version of the building energy simulation tool CitySim+ (Zakhary et al. 2016), and perform some computational experiments using this distributed simulation tool in order to measure and compare the communication and computation costs involved. Further, the measurements from these experiments help us to understand the limitations and trade-offs involved in using this approach and help us to evaluate the appropriateness of using the HLA framework for scaling up building energy simulations.

Physical simulation of building energy demands can be used to predict the energy requirements of a collection of buildings over a given period of time, using 3D models of the buildings in geographical space. Such building energy simulations can be highly demanding of computational resources such as memory and processor utilization, especially as the scale of the simulation model grows very large. Models of interest may range from those containing a few buildings representing a small neighborhood, to models that consist of several hundreds of thousands of buildings at the city level (Reinhart and Cerezo Davila 2016). Models with very large configurations approaching the city level with thousands of buildings are particularly challenging and can require more resources than a single commercial off-the-shelf (COTS) computing node with reasonably high specifications would be able to support, a limitation encountered during our experiments. To support the execution of such large building energy simulation models, therefore, it is necessary to adopt an approach that scales up the simulation execution by distributing the overall workload among multiple computing nodes and coordinating the execution of the simulation between the separate nodes. A further challenge to reckon with is that it is often desirable to coordinate the execution of building energy simulations with other related urban simulations such as building occupancy models (Zhao and Magoulès 2012; Chapman, Siebers, and Robinson 2014) or urban meteorological models (Mauree et al. 2017) in order to improve the accuracy of the predicted energy demands. Therefore, it is desirable to use a holistic approach that is not only capable of addressing the simulation scalability challenge, but should also be capable of supporting interoperability between heterogeneous simulations.

We consider the HLA as an appropriate framework that provides adequate functionality to meet the simulation scalability and interoperability challenges outlined above. Although this work focuses on the scalability aspect for large simulations, our selection of the HLA will enable potential interoperation of the building energy simulation with other urban simulations in future work. The HLA is an IEEE standard (IEEE 2010) that enables the coordinated execution of distributed simulations. The HLA defines the set of rules that govern the overall simulation execution, which is referred to as a *federation execution* in HLA terms. These include procedures for exchanging data between the disparate simulators, which are described as *federates* in HLA terminology. Such data exchange between federates in the HLA does not occur directly but is rather coordinated by the Run Time Infrastructure (RTI), a central component of the HLA framework, using the publish-subscribe pattern to ensure that federates need not know the identity of the other federates which produce the data they rely on. The RTI is an essential component of the HLA responsible for coordinating the progression of the federation execution. Some available open source RTI implementations include Portico (Portico 2018), OpenRTI (OpenRTI 2018) and CERTI (Noulard, Rousselot, and Siron 2009). The HLA has been used in many application areas ranging from biological simulations (Lees, Logan, and King 2007) to urban modeling (Jain et al. 2016). In this work, we examine the suitability of the HLA for the purpose of executing large scale building energy simulations. To this end, we develop a proof-of-concept HLA-compliant version of the existing CitySim+ building energy simulation tool (Zakhary et al. 2016) and run some experiments on a network of computing nodes in order to give some indication of how well suited the HLA framework is to building energy simulation.

The remaining sections are organized as follows. In section 2, we give a brief overview of the High Level Architecture (HLA) framework. In section 3, we describe the building energy simulation model. In section 4, we provide an explanation of the conceptual design of our HLA building energy simulation. In section 5, we present results from our experiments in which we distribute building energy simulations over multiple computing nodes using the HLA framework. In section 6, we provide a summary of work related to distributed building energy simulation. Finally, we conclude and discuss some directions for future work.

2 OVERVIEW OF THE HIGH LEVEL ARCHITECTURE

Here, we provide a brief overview of the HLA and outline some portions of the framework relevant to this work. Originating as a military specification designed to facilitate the coordinated execution of separate simulation models, the HLA was eventually adapted into an IEEE standard (IEEE 2010). The HLA serves as a standard framework on which disparate application-specific simulations may be combined to create a single overall simulation execution by coordinating the separate simulations to exchange useful data and regulate their time advancement relative to one another. The HLA standard is composed of three parts: a specification that lays out the structure of the framework and defines a set of rules by which all federates and federations must abide, an interface definition consisting of eight classes of services that determine how the RTI interacts with the federates, and a template for specifying the type of data that can be exchanged between federates during a federation execution. Data exchange between federates is based on the publish-subscribe pattern and is accomplished by the use of HLA "objects". In a similar way to OOP objects, HLA objects are based on a class template. However, they do not have any associated methods and are simply used as structures for encapsulating a group of attributes. Each object attribute in the federation is owned by at most one federate, and owning federates are solely responsible for updating the values of the object attributes they own in the federation, although other interested federates may also read the values of these attributes.

With the publish-subscribe pattern, publishing federates register the objects that they own with the RTI and declare their intention to publish the relevant class attributes in order to be granted write access. Subscribing federates also notify the RTI about which class attribute updates they wish to receive. From this point onwards, the RTI accepts object attribute updates from registered publishing federates and forwards the updates to the appropriate subscribing federates. This arrangement ensures that federates are anonymous to one another. A subscribing federate need not know the identities of the other federates that publish the data it requires, and a publishing federate need not keep a list of other federates that need to receive the updates that it publishes. The RTI alone has access to the overall publish-subscribe information and acts as a router between federates, receiving and forwarding attribute updates as appropriate. In order to ensure that all federates have a mutual agreement regarding the data that can be exchanged during federation execution, the Federation Object Model (FOM) document is used to specify a federation agreement, providing details of all object attributes that are expected to be exchanged between federates during federation execution. The FOM document is based on the Object Model Template (OMT) supplied by the HLA standard.

3 SIMULATION OF BUILDING ENERGY DEMANDS

The following is a brief description of the building energy simulation model used in the CitySim+ tool. Physical building energy simulations make use of 3D representations of buildings positioned in geographical space and consider the weather patterns over the simulated period in order to calculate the energy exchanges between buildings and the environment. The results from these energy exchange calculations are used to estimate the overall the building energy demands over the simulated time period. Each 3D building consists of a combination of several surfaces including wall surfaces, roof surfaces and floor surfaces. These surfaces combine to create one or more internal thermal zones within each building which can transfer heat between one another. Such a model of a group of buildings represented in 3D space may be referred to as a *scene*. Scenes may range widely in complexity depending on the task to hand. On the extremely simple end of the scale, a scene may contain a few shoebox-like buildings with simple flat roofs, each having four walls that join together with the roof and floor to create a single internal thermal zone. On the more complex end of the scale, a scene may contain several thousand buildings with complicated structures and multiple internal thermal zones.

As explained by Robinson (2012), building energy simulations attempt to predict the radiation exchange of building surfaces with one another and with the environment in order to obtain a basis for estimating the energy demands required for occupant comfort. In the context of building energy simulation, *shortwave* radiation exchange refers to the portion of solar radiation with wavelengths in the range of 0.3μm to 3μm that is absorbed by building surfaces. Building surfaces also exchange infrared radiation with each other in the wavelengths ranging from 3μm to 100μm, which is referred to as *longwave* radiation exchange. These radiation exchange processes contribute significantly to indoor and outdoor temperature and heat transfer between and within buildings. Simulating these radiation exchanges in the urban setting is further complicated by the fact that surfaces obstruct one another from directly viewing the sun and sky as well as obstructing one another from viewing other surfaces. However, this also serves to make such simulations more computationally tractable as not all pairs of surfaces need to exchange radiation with each other, resulting in a sparse matrix of building surface-to-surface radiation exchanges. The shortwave and longwave radiation exchanges between buildings and the environment in an urban setting are illustrated conceptually in Figure 1.

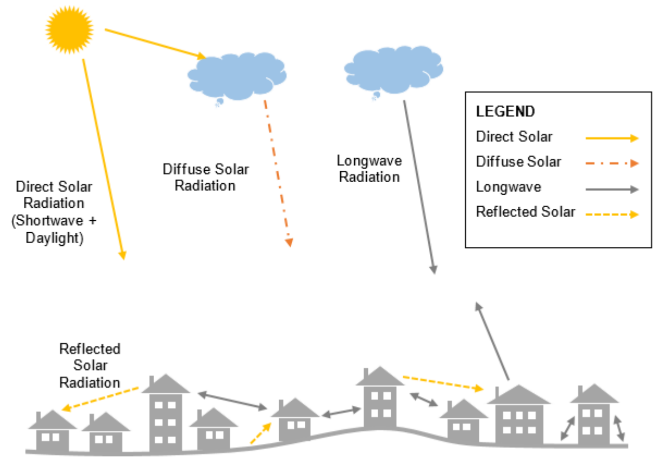

Figure 1: Illustration of some building-related radiation exchanges in an urban environment.

We have provided an overview of the building energy simulation model here, but do not discuss the details of the mathematical models which form the basis for physical calculations of the radiation exchanges that occur between buildings and the environment. We recommend that the interested reader consult Robinson (2012) for a more detailed treatment of these mathematical models. In brief, these mathematical models calculate the contributions to the shortwave radiation that each building surface receives directly from the sun, from diffuse solar radiation through the sky, and by reflected solar radiation from other building surfaces. They also calculate the contributions to longwave radiation on each building surface that is received from the sky and from the other surrounding surfaces which are not obstructed from view. The CitySim+ building energy simulation tool (Robinson et al. 2009) makes use of these mathematical models to calculate the radiation exchanges between building surfaces over the simulation time period, utilizing the Simplified Radiosity Algorithm (SRA) of Robinson and Stone (2006). The calculations are done in an hourly time-step loop, with each time-step consisting of four main physical calculation steps: shortwave radiation exchange calculations, daylight calculations, longwave radiation exchange calculations and thermal zone calculations. The daylight calculations are useful for determining the amount of indoor illumination supplied by natural sunlight, while the thermal zone calculations are important for determining the resulting temperature of internal thermal zones and building surfaces. These measures are useful as a basis for determining the additional heating and lighting needs that will be required for ensuring occupant comfort in residential buildings over the simulation time period. The representation of building scenes in CitySim+ is based on CityGML(CityGML 2018), an XML-based standard which uses markup to describe urban structures in 3D physical space. For the purposes of urban energy simulations, previous work on CitySim+ (Zakhary et al. 2016) has extended its capabilities to utilize CityGML with the Energy Application Domain Extension (ADE) in order to enrich the scene with additional energy-related features.

4 DISTRIBUTED SIMULATION ARCHITECTURE

As described in the previous section, a CitySim+ building scene is a CityGML representation of a collection of 3D buildings, each composed of multiple surfaces and having one or more internal thermal zones. As obstructions in the building scene means that not all surface pairs are able to exchange radiation with one another, radiation exchange relationships between building surfaces are captured in a sparse view-factor matrix which is computed in a pre-processing step before the hourly time-stepped radiation exchange simulation loop begins. These view-factor relationships between building surfaces can be represented as a graph in which vertices represent surfaces, and edges connect surfaces that can exchange radiation with each other.

To facilitate decomposition of the building scene for distribution among federates, such a graph of view-factor relationships can be used partition the surfaces into different clusters. The work by Zakhary et al.

(2018) provides a fuller account of the details of the clustering process of which the results are used in the experiments conducted in this work. We do not reproduce all the details of the clustering process here, as the clustering approach is not the main focus of this paper. However, we provide a brief summary: a greedy community detection algorithm is used for clustering in an effort to group surfaces that are strongly connected to one another within the same cluster, while minimizing the number of edges between surfaces in different clusters. Each of these clusters can then be assigned to a separate CitySim+ federate for local processing. The minimization of connections between clusters is important for keeping communication overheads low between federates during federation execution. We also note that while the actual radiation exchange relationships exist between building surfaces, there is the additional constraint that all surfaces belonging to the same building should be assigned to the same cluster. This essentially ensures that the clustering is performed at the building level rather than the surface level. This constraint is good for the quality of the clustering, as surfaces belonging to the same building tend to have the strongest relationships with one another, thereby helping to reduce the total number of edges between different clusters, and hence resulting in fewer communication links between federates.

Figure 2 displays a conceptual illustration of the relationships between federates and building surface objects. The solid line boundaries indicate the cluster boundaries. Surface objects contained within the solid cluster boundaries are owned and simulated by the same federate. On the other hand, the dashed line boundaries show the extent of the full area of interest of each federate. A federate's area of interest also includes external surface objects, owned by other federates, that have radiation exchange relationships with the federate's own internal surface objects. Each federate needs to keep track of the attribute values of these external surface objects as they affect the accuracy of its own local computations.

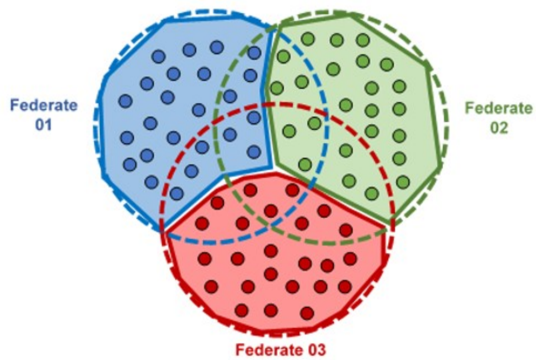

Figure 2: Conceptual illustration of federates' areas of interest.

It is clear that the surface objects which fall into the overlapping areas of interest of different federates are the only ones that need to be considered during the data exchanges that occur during federation execution. Conversely for surfaces that only have radiation exchange relationships with other surfaces within the same federate, their attribute values can be kept solely for local computations and need not be shared with any external federates, and therefore need not be registered with the RTI as objects available for publication.

Figure 3 presents a conceptual diagram showing the execution model of the CitySim+ HLA federation. As explained in section 3, each hourly time-step in a CitySim+ simulation is composed of four sub-computations: shortwave radiation exchanges (SW), daylight calculations (DL), longwave radiation exchanges (LW) and thermal zone heat transfers (TH). These computations have some dependencies on one another. The thermal zone computations depend on the results from the longwave computations within the same time-step. In turn, the longwave computations depend on the results of the shortwave computations within the same time-step, as well as the results of the thermal zone computations from the previous time-step. As a result of these dependencies, data needs to be exchanged between federates after some of the local computation steps have been completed in order to ensure the correctness of the final results. As shown in Figure 3, data is exchanged between federates during the shortwave, daylight and thermal calculations. However, no data needs to be exchanged between CitySim+ HLA federates after the longwave computations. This is due to the fact that the subsequent thermal zone computations only depend on the longwave results from surfaces in the same building within the same time-step. As described in section 3, the clustering is effectively done at the building level and therefore all surfaces belonging to the same building are local to the same federate. Consequently, data exchange is not required after longwave calculations for the thermal zone calculations to be correct.

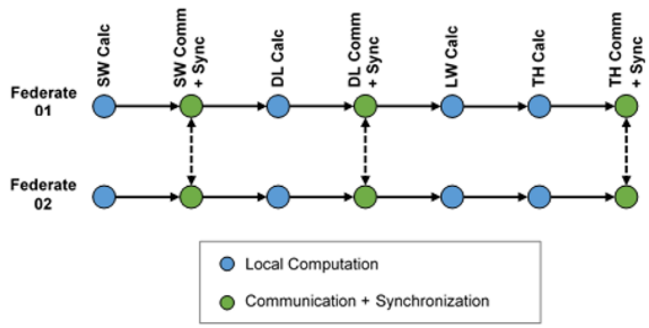

Figure 3: Conceptual diagram of CitySim+ federation execution.

5 EXPERIMENTS AND RESULTS

All experiments carried out with the CitySim+ HLA federation were conducted on a network of 12 virtual machines running CentOS Linux 7.4, each with 8GiB of RAM and 2x Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz. The experiments made use of the open source Portico RTI (Portico 2018) to support the CitySim+ HLA federation executions. Our reasons for selecting this RTI implementation include its open source status and its active development community. In our experiments, each federate is executed on a separate computing node, and we may use the terms federate and node interchangeably in certain places.

The experiments were conducted on a CitySim+ scene composed of 2,980 buildings representing an area of the township of Sneinton in Nottingham, UK. This scene is available in two types, as a "simple" scene and as a "complex" scene. In the simple scene, all the buildings are represented as simple shoebox-like shapes, and this design makes up to a total of 25,514 building surfaces. The complex scene represents the building structures in more realistic detail, raising the total number of building surfaces by almost five-fold to 122,847. We note here that for purposes of comparison between experiments, a clustering produced for the complex scene can also be used for the simple scene, as the clustering is done at the building level and both scenes contain the same buildings. Each experiment was run at an hourly simulation time-step for one year of simulation time, plus an initial fifteen-day warm-up period that is discarded and only required for booting up the model. The code of the CitySim+ HLA federate implementation was instrumented to measure various timings of interest with regard to the time spent in executing the various computations as well as the time spent in synchronization and exchange of data between federates. The additional processing overheads introduced by this instrumentation was small, typically adding less than 1% of additional processing time. Computation and communication alternate during federation execution as explained in section 4. Figure 4 further illustrates the events that occur in one computation and communication cycle for the case of two federates with equal workloads. Each federate completes its workload, publishes its surface data to the RTI and waits to receive surface data from other federates via the RTI. Following data exchange the RTI grants a time advance to all federates, allowing them to proceed.

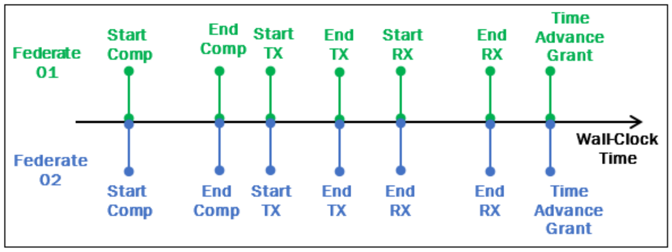

Figure 4: Computation and communication events.

The graph in Figure 5 compares the computation wall-clock times versus communication and synchronization wall-clock times when distributing the simple and complex scenes across varying numbers of CitySim+ HLA federates. The varying numbers of federates were obtained from the clustering of the simple and complex scenes into sub-scenes of 2 to 12 clusters. The complex scene, being almost five times larger than the simple scene, was too large to fit into the memory of 2 computing nodes in our experimental setup. For this reason, the minimum number of federates used in the complex scene experiments is 4.

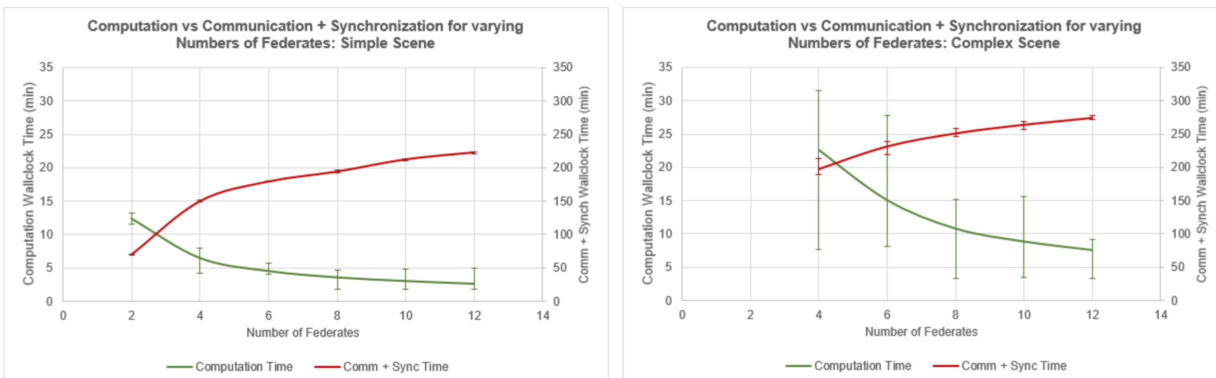

Figure 5: Computation times vs communication + synchronization times with varying numbers of federates for simple scene (left) and complex scene (right).

The communication + synchronization wall-clock time is measured from the start of transmitting data to the receipt of a time advance grant from the RTI. The computation error bars in Figure 5 show some small differences in computation time among federates due to variances in their assigned workloads. These variances ensue from the need to minimize inter-federate communication links during scene partitioning. Two separate axes are used in Figure 5 to make the trends clearer, as the communication and synchronization time seems to outweigh the computation time by a factor ranging from about 5x to 45x. As expected, the computation time decreases as the number of federates increases and each federate is assigned a smaller portion of the overall workload. Conversely, the communication + synchronization time rises with the number of participating federates. The communication + synchronization costs are lowest when the scene is distributed between 2 and 4 federates respectively for the simple and complex scenes. This suggests that for large model configurations that cannot be supported by the resources of a single computing node, it may be favorable to distribute over the minimum number of computing nodes that can support the entire model configuration in order to minimize potential communication overheads. It should be clarified that there are no actual intersection points in Figure 5, and the lines only appear to intersect as a result of the two separate axes used.

It is worth noting that although the scene size has increased by about 5x from simple scene to complex scene, the communication overheads only grow about 1.3x on average. For example, in the 4-federate experiments the communication overheads take about 150 min on average for the simple scene compared with 200 min for the complex one, and in the 12-federate experiments the overheads take 223 minutes on average for the simple scene and 274 minutes for the complex one.

Figure 6 shows how the computation time varies with average workload assigned to each federate (the number of building surfaces per federate). The error bars indicate the variations in workload and computation time between federates. As expected, the trend shows that the local computation time on each federate is proportional to its assigned workload.

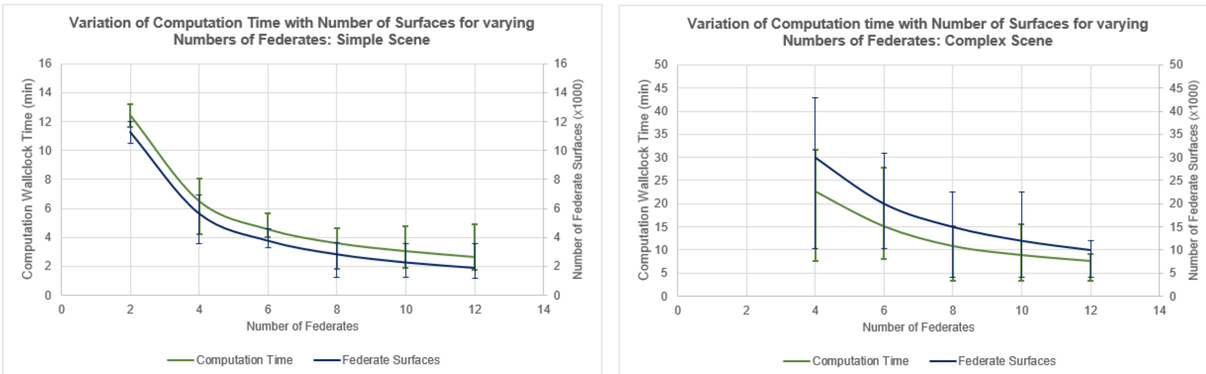

Figure 6: Variation of computation time with number of surfaces for varying numbers of federates, simple scene (left) and complex scene (right).

Figure 7 shows how the communication + synchronization time varies with the number of inter-federate links and the total number of published surfaces for the simple and complex scenes. The number of published surfaces represents the total volume of data that is sent out from all federates, and this covers the set of surfaces in the overlapping areas of interest. On the other hand, the total number of inter-federate links represents the volume of data received by federates, and is higher as a single published surface may be received by multiple subscribing federates. The trends show that the growth in communication costs corresponds more closely to the published surfaces than it does to the growth of inter-federate links, which increases faster than the other two. This is not surprising, as the portico RTI makes use of reliable multicast to transmit data from a single sender to multiple recipients (Portico 2018, Ross 2012).

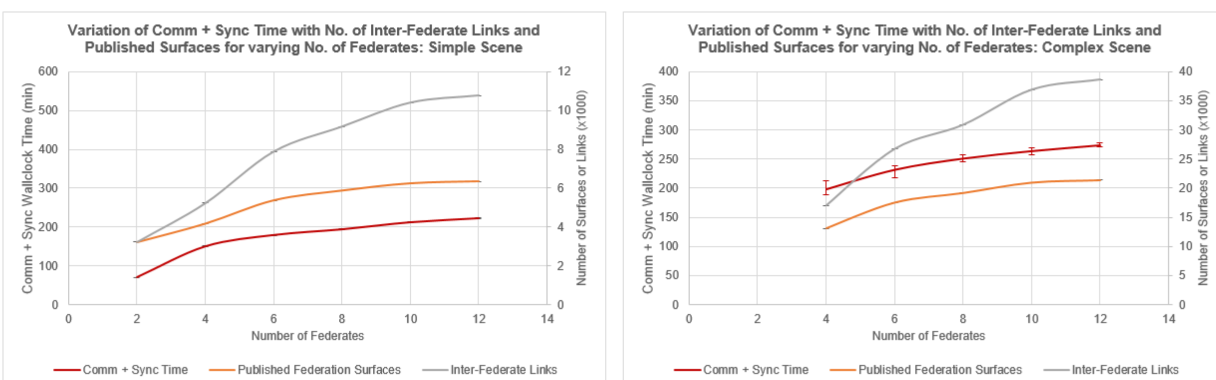

Figure 7: Variation of communication + synchronization time with number of published surfaces and inter-federate links, simple scene (left) and complex scene (right).

The boxplots in Figure 8 show the variations between federates for the experiments involving 12 federates. In the simple scene experiment, the range in the computational costs shows the slowest federate lagging behind the fastest by about 3.1 min. The communication + synchronization overheads for each federate includes the time spent waiting for other federates to finish their workloads, the time spent in sending and receiving simulation data, and the time spent in sending and receiving signaling information to/from the RTI (such as for time advancement). As the total cost of communication + synchronization is more than 220 min, the maximum lag of 3.1 min due to differing workloads only represents a small fraction of the total communication + synchronization costs. Similarly for the complex scene, the lag of 5.8 min computation time between the fastest and slowest federates is small in comparison with the overall communication + synchronization time, which exceeds 270 min. It is worth noting also that while the complex scene size has increased almost 5x from the simple scene, the maximum communication costs involved have only increased from about 224 min to about 278 min which is less than 1.3x. This may be explained by the fact that the overlaps between federates' areas of interest do not increase proportionally with the rise in scene size, and therefore the number of published surfaces does not grow as quickly as the scene size does.

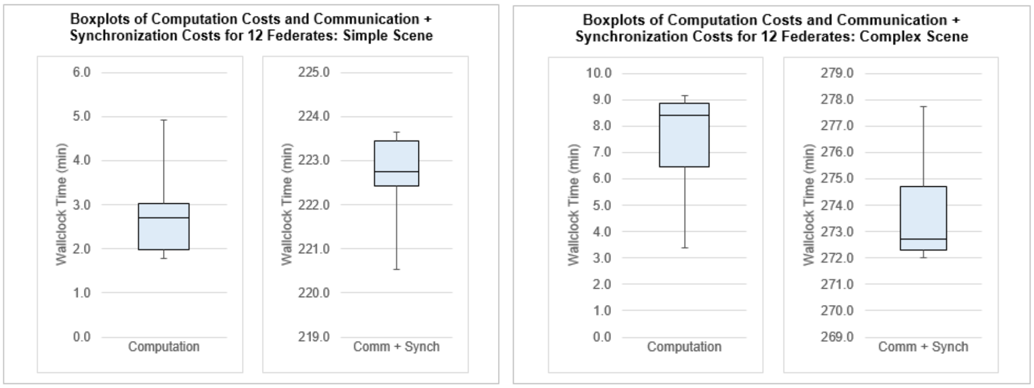

Figure 8: Computation time and communication + synchronization time boxplots for 12-federate experiments, simple scene (left) and complex scene (right).

The results presented in this section have demonstrated that large scale distributed building energy simulation is feasible using the HLA framework to distribute partitioned building scenes over multiple computing nodes. The communication overheads involved tend to be greater than the computation costs. However the overheads do not tend to grow proportionally with scene size, only increasing by about 1.3x in our experiments even as the scene size grew 5x from the simple scene to the complex scene.

Our experiments here may be limited in the sense that we have only considered building scenes containing less than 3,000 buildings, while our target scale of an entire city could contain 100,000 buildings or more. However, the trends identified in these experiments are useful for the primary aim of this work, which is to demonstrate the feasibility of our approach for large scale simulations. Our experiments may also be limited with respect to integration of building energy simulation with other simulations such as occupant presence simulations, which was one of the reasons for selecting the HLA approach. We hope to address these limitations in future work.

In these experiments, we have also made the key assumption that relationships between building surfaces are static for the entire duration of the simulation. This is a reasonable assumption for our building energy simulation, and we assume that the building scene does not change over the simulated time. For example, no new buildings are added and existing buildings are not removed or modified during the time period of simulation execution. Consequently, the publish-subscribe relationships between federates are not expected to change during simulation run. This property may be specific to our building energy simulation model, and therefore our approach as described here may not apply to simulations in which relationships between simulation objects are normally expected to change dynamically as simulation execution proceeds.

6 STATE OF THE ART

Regarding standards-based distributed simulation, some of the main distributed simulation frameworks that have been used within the urban modeling context include the HLA and the Functional Mockup Interface (FMI) (Blochwitz et al. 2011). The FMI standard has two objectives: to produce simulation models in a standard format such that they can be exchanged between different simulation tools, and to enable co-simulation between different simulation models. With regards to co-simulation, the FMI requires the modeler to create a custom *master algorithm* to coordinate the execution of the FMI's simulation components, which are referred to as Functional Mock-up Units (FMUs). The specification of this master algorithm is not standard, although it provides a function similar to that of the HLA RTI. Although distributed simulation over multiple computing nodes is not one of the objectives of the FMI standard, some work has been done by Galtier et al. (2015) to make this feasible. Some effort has been made to ease the development of FMUs for co-simulation, such as the work of Aslan et al. (2015). Work has also been done to bridge the HLA and FMI standards. Awais et al. (2013) propose the approach of using FMUs as federates in an HLA federation coordinated by an RTI, with the aim of removing the need to create custom master algorithms for different FMI co-simulations. Yilmaz et al. (2014) adopt this approach and propose a design for a wrapper around FMUs to enable them participate in an HLA federation. Bouanan et al. (2018) also discuss approaches to combining HLA with FMI and provide a demonstration FMU federate.

Some examples of work that has applied the HLA or FMI standards to urban simulations include the work of Jain et al. (2016), in which the HLA is used to integrate the land use model UrbanSim (Waddell 2002) with the transport model MATSim (Horni, Nagel, and Axhausen 2016), and the work of Menassa et al. (2014), where a building energy simulation is integrated with an occupancy model using the HLA. The work of Wang, Siebers, and Robinson (2017) also uses the FMI to couple the building energy simulation tool EnergyPlus with the occupant behavior simulation tool No-MASS. However, we are not aware of any previous work that applies the HLA framework to enable large scale building energy simulation by distribution over multiple computing nodes.

7 CONCLUSION

In this paper, we have demonstrated the feasibility of applying the HLA to enable execution of large building energy simulations that cannot be supported on a single computing node by distributing it over multiple nodes. To this end, we have developed an HLA-compatible version of the building energy simulation tool CitySim+, and have presented results from experiments using the CitySim+ HLA federation with differing numbers of federates and scene sizes.

Furthermore, in our experiments we have also measured the trade-offs between computational costs and communication overheads. From the results of these experiments, we have noted that the communication costs involved in the building scenes we have tested seem to outweigh the processing costs involved in performing the simulation's computations. It is desirable that these additional communication overheads be minimized to allow the distributed simulation to be run as efficiently as possible. Our plans for future work involve implementing methods to improve the communication efficiency of the HLA framework for distributing building energy simulations. This work will involve the identification and analysis of the most significant factors contributing to the high communication overheads present in our HLA distributed simulation. From this analysis, the future work will explore methods to suppress or circumvent the effects of such factors, perhaps sacrificing small losses in overall simulation accuracy in order to obtain a comparatively large gain in communication efficiency. The validation of our distributed experiments is also a matter that will be dealt with in future work.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Leverhulme Trust, sponsors of the Sustaining Urban Habitats project under which this work was conducted (grant RP 2013-SL-015). We also thank Dr. Julian Rosser for providing the CityGML building scene files used in the experiments.

REFERENCES

Aslan, Memduha, Halit Oguztuzun, Umut Durak, and Koray Taylan. 2015. *MOKA: An Object-Oriented Framework for FMI Co-Simulation*. *Proceedings of the Conference on Summer Computer Simulation*. Society for Computer Simulation International.

Awais, Muhammad Usman, Peter Palensky, Atiyah Elsheikh, Edmund Widl, and Stifter Matthias. 2013. "The High Level Architecture RTI as a Master to the Functional Mock-up Interface Components." *2013 International Conference on Computing, Networking and Communications, ICNC 2013*, 315–20.

Blochwitz, T., M. Otter, M. Arnold, C. Bausch, C. Clauss, H. Elmqvist, A. Junghanns, et al. 2011. "The Functional Mockup Interface for Tool Independent Exchange of Simulation Models." In *Proceedings from the 8th International Modelica Conference, Technical Univeristy, Dresden, Germany*, 63:105–14. Dresden, Germany: Linköping University Electronic Press.

Bouanan, Youssef, Simon Gorecki, Judicael Ribault, Gregory Zacharewicz, and Nicolas Perry. 2018. "Including in HLA Federation Functional Mockup Units for Supporting Interoperability and Reusability in Distributed Simulation." *Proceedings of the 50th Computer Simulation Conference*. Society for Computer Simulation International.

Chapman, Jacob, Peer-olaf Siebers, and Darren Robinson. 2014. "Coupling Multi-Agent Stochastic Simulation Of Occupants With Building Simulation." *Building Simulation and Optimization Conference (BSO14)*.

CityGML. 2018. "CityGML Standard." 2018.

Galtier, Virginie, Stephane Vialle, Cherifa Dad, Jean-Philippe Tavella, Jean-Philippe Lam-Yee-Mui, and Gilles Plessis. 2015. "FMI-Based Distributed Multi-Simulation with DACCOSIM." *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M &S Symposium* DEVS '15: 39–46.

Horni, Andreas, Kai Nagel, and Kay W Axhausen. 2016. *The Multi-Agent Transport Simulation MATSim*. *The Multi-Agent Transport Simulation MATSim*.

IEEE. 2010. "IEEE Std 1516-2010, High Level Architecture (HLA) Framework and Rules."

Jain, Ajitesh, David Robinson, Bistra Dilkina, and Richard M. Fujimoto. 2016. "An Approach to Integrate Inter-Dependent Simulations Using HLA with Applications to Sustainable Urban Development." *Proceedings - Winter Simulation Conference*.

Lees, Michael, Brian Logan, and John King. 2007. "HLA Simulation of Agent-Based Bacterial Models." *Proceedings of the 2007 European Simulation Interoperability Workshop*.

Mauree, Dasaraden, Silvia Coccolo, Jérôme Kaempf, and Jean Louis Scartezzini. 2017. "Multi-Scale Modelling to Evaluate Building Energy Consumption at the Neighbourhood Scale." *PLoS ONE* 12 (9).

Menassa, Carol C., Vineet R. Kamat, SangHyun Lee, Elie Azar, Chen Feng, and Kyle Anderson. 2014. "Conceptual Framework to Optimize Building Energy Consumption by Coupling Distributed Energy Simulation and Occupancy Models." *Journal of Computing in Civil Engineering* 28 (1): 50–62.

Noulard, Eric, Jean-yves Rousselot, and Pierre Siron. 2009. "CERTI , an Open Source RTI , Why and How." *Spring Simulation Interoperability Workshop*.

OpenRTI. 2018. "OpenRTI Repository." 2018.

Portico. 2018. "Portico Repository." 2018.

Reinhart, Christoph F., and Carlos Cerezo Davila. 2016. "Urban Building Energy Modeling - A Review of a Nascent Field." *Building and Environment* 97: 196–202.

Robinson, Darren. 2012. "The Urban Radiant Environment." In *Computer Modelling for Sustainable Urban Design*, 35–74. Routledge.

Robinson, Darren, Frédéric Haldi, Jérôme Henri Kämpf, P Leroux, D Perez, a Rasheed, and U Wilke. 2009. "CITYSIM: Comprehensive Micro-Simulation Of Resource Flows For Sustainable Urban Planning." *International IBPSA Conference*, 1083–90.

Robinson, Darren, and Andrew Stone. 2006. "Internal Illumination Prediction Based on a Simplified Radiosity Algorithm." *Solar Energy*.

Ross, Peter. 2012. "Comparison of High Level Architecture Run-Time Infrastructure Wire Protocols – Part One."

Waddell, Paul. 2002. "UrbanSim: Modeling Urban Development for Land Use, Transportation, and Environmental Planning." *Journal of the American Planning Association* 68 (3): 297–314.

Wang, Kunpeng, Peer-olaf Siebers, and Darren Robinson. 2017. "Towards Generalized Co-Simulation of Urban Energy Systems." In *Procedia Engineering*.

Yilmaz, Faruk, Umut Durak, Koray Taylan, and Halit Oğuztüzün. 2014. "Adapting Functional Mockup Units for HLA-Compliant Distributed Simulation." *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden* 96: 247–57.

Zakhary, Sameh, Andrew Allen, Peer-olaf Siebers, and Darren Robinson. 2016. "A Computational Workflow for Urban Micro-Simulation of Buildings' Energy Performance." In *Urban Transitions Global Summit*.

Zakhary, Sameh, Julian Rosser, Peer-olaf Siebers, Yong Mao, and Darren Robinson. 2018. "Using Unsupervised Learning to Partition 3D City Scenes for Distributed Building Energy Micro-Simulation." *Manuscript Submitted for Publication*.

Zhao, Hai Xiang, and Frédéric Magoulès. 2012. "A Review on the Prediction of Building Energy Consumption." *Renewable and Sustainable Energy Reviews* 16 (6): 3586–92.

AUTHOR BIOGRAPHIES

KWABENA AMPONSAH is a Ph.D. student at the School of Computer Science, University of Nottingham. He is a member of the inter-disciplinary "Sustaining Urban Habitats" project funded by the Leverhulme Trust, and his research concerns parallel and distributed simulation in the urban context. His email address is kwabena.amponsah@nottingham.ac.uk.

SAMEH ZAKHARY is a Leverhulme Research Fellow at the School of Computer Science, University of Nottingham (UoN). He received his B.Eng. in Computer Engineering from Cairo University in 2000; MBA from the American University in Cairo (AUC) in 2009; and an M.S./PhD in Computer Science from UoN in 2009 and 2016 respectively. Currently, he is researching technologies for sustainable cities in a prestigious Leverhulme Trust-funded project "Sustaining Urban Habitats: An Interdisciplinary Approach". Before his Ph.D., Sameh has worked as a technology and business consultant for major multinational information technology companies. He is the recipient of Jameel Fellowship from AUC in 2007; the Best ITI M.S. Dissertation Award and International Research Excellence scholarship from UoN in 2009 and 2010 respectively. He has published work in venues including ACM MobiCom, IEEE ICC, IEEE WONS, and IEEE IWCMC. His research interests include delay tolerant, P2P, privacy and security, and mobile ad-hoc networks. His email address is sameh.zakhary@nottingham.ac.uk.

DARREN ROBINSON is Chair in Architectural and Urban Sciences at the University of Sheffield. His main passion is for research that makes a difference. Much of his research is conducted at the interface between social physics (people), building physics (buildings) and urban physics (city). This research is conducted at multiple scales: from the performance of individual buildings in their urban context and how this performance is influenced by occupants; to the performance of entire complex urban systems and how their spatial and functional structures can be improved upon to maximize metabolic efficiency. His email address is d.robinson1@sheffield.ac.uk.

BRIAN LOGAN is a Member of the Agents Lab and the Mixed Reality Laboratory at the University of Nottingham. His research interests lie in the area of agent systems, and span the specification, design and implementation of agents, including agent architectures, agent programming languages and logics and theories for agent-based systems. He is also interested in applications of agents, particularly in virtual environments and in simulation. His email address is brian.logan@nottingham.ac.uk.

PAUL NATHANAIL is a Specialist in Land Condition (SiLC) and a Chartered Geologist. Paul's expertise includes developing decision support tools for regulators and industry, developing spatial and non-spatial environmental risk assessment tools and developing tools to facilitate sustainable brownfield regeneration. He has extensive contacts with industry and regulators in the UK, Europe, Africa, Asia and North America. He has recently been appointed to the Defra expert panel on contaminated land. He has acted as an expert witness in several planning, environmental and criminal cases. His email address is paul@lqm.co.uk.

PEER-OLAF SIEBERS is an Assistant Professor at the School of Computer Science, University of Nottingham, UK (http://www.cs.nott.ac.uk/~pszps/). His main research interest is the application of computer simulation to study human-centric complex adaptive systems. He is a strong advocate of Object Oriented Agent-Based Social Simulation. This is a novel and highly interdisciplinary research field, involving disciplines like Social Science, Economics, Psychology, Operations Research, Geography, and Computer Science. His current research focuses on Urban Sustainability and he is a co-investigator in several related projects and a member of the university's "Sustainable and Resilient Cities" Research Priority Area management team. His email address is peer-olaf.siebers@nottingham.ac.uk.