MWGrid: A System for Distributed Agent-based Simulation in the Digital Humanities

Bart Craenen[1], Philip Murgatroyd[3], Georgios Theodoropoulos[2], Vincent Gaffney[3], Vinoth Suryanarayanan[1]

[1] School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK
[2] IBM Research, Ireland
[3] Institute of Archaeology and Antiquity, University of Birmingham, UK

Abstract—**Digital Humanities offer a new exciting domain for agent-based distributed simulation. In historical studies interpretation rarely rises above the level of unproven assertion and is rarely tested against a range of evidence. Agent-based simulation can provide an opportunity to break these cycles of academic claim and counter-claim. The MWGrid framework utilises distributed agent-based simulation to study medieval military logistics. As a use-case, it has focused on the logistical analysis of the Byzantine army's march to the battle of Manzikert (AD 1071), a key event in medieval history. It integrates an agent design template, a transparent, layered mechanism to translate model-level agents' actions to timestamped events and the PDES-MAS distributed simulation kernel. The paper presents an overview of the MWGrid system and a quantitative evaluation of its perfomance.**

Index Terms—**Dostributed Simulation, Agent-based systems, historical studies, medieval military logistics, Byzantine, Manzikert**

I. Introduction

Agent-based simulation, as a means to model and explore the effect of individual action in complex systems has recently witnessed an explosion of interest and application in a wide range of domains. Perhaps surprisingly, one of these domains is historical studies. Historical interpretation is associated with clear narratives defined by a series of fixed events or actions. In reality, even where critical historical events may be identified, contemporary documentation frequently lacks corroborative detail that supports verifiable interpretation. Consequently, for many periods and areas of research, interpretation rarely rises above the level of unproven assumptions, rarely or never tested against a range of evidence.

Agent-based simulation provides a vehicle to recreate and study historical societies and events. Recent include the analysis of resource exploitation by Mesolithic hunter gatherer groups [1] and the Prehispanic settlements in North America [2] have generally involved the analysis of small-scale groups at individual or household level rather than larger societies [3]. The "Medieval Warfare on the Grid" (MWGrid) project at the University of Birmingham in the UK utilised agent-based modelling to study behaviour dynamics at a large scale, within the context of modelling logistical arrangements relating to the march of the Byzantine army to the battle of Manzikert (AD 1071) - resulting in the collapse of Byzantine Empire in central Anatolia [4].

MWGrid developed a modelling framework for modelling agents and a series of complex models of the marching Byzantine army for what-if analysis incorporating a rich set of data, both historical and current. As a simulation infrastructure, it utilises the PDES-MAS system through an interface layer that was built for this purpose. Although the MWGrid infrastructure and framework was developed especially to support the MWGrid model, the interface it provides can also be used for other (types of) agent-based models. Different aspects of MWGrid and PDES-MAS have been previously described in [5] [6] [7] [8] and [9] [10] [11] [12] [13] [14][15] [16] [17] [18] [19] respectively. This paper presents an overview of the integrated MWGrid system and, for the first time, a quantitative evaluation in terms of performance and memory requirements.

The rest of the paper is structured as follows: Section II provides an overview of the problem domain to

This research was undertaken while Georgios Theodoropoulos was with the School of Computer Science, University of Birmingham, UK.

This project was funded by a joint AHRC/EPSRC/JISC e-Science research grant as the Medieval Warfare on the Grid (MWGrid) project. Computational resources for the project were provided by the University of Birmingham's BlueBEAR HPC service, acquired through HEFCE SRIF-3 funds.

facilitate understanding of the models involved. Section III presents a brief overview of the MWGrid framework while sections IV, V and VI look into the MWGrid contituent components in more detail. Section VII reports experimental results for different MWGrid models. The paper concludes in section VIII.

II. Modelling Medieval military logistics: the Manzikert campaign

A key problem in historical studies is to analyse and understand the implications of the need for medieval states to collect and distribute resources to maintain armies. Study of the events associated with the Byzantine army's march across Anatolia to the Battle of Manzikert is particularly attractive in this context. This was a major logistical challenge that involved the largest Byzantine army for over 50 years (tens of thousands of people, horses and mules along with tons of equipment) travelling more than 700 miles across what is now part of the modern state of Turkey, from near Constantinople (modern Istanbul) to Manzikert (modern Malazgirt) just north of Lake Van (figure 1).The Byzantines' subsequent defeat by the Seljuk Turks and their Sultan Alp Arslan on August 26th was considered so catastrophic that the Byzantine chroniclers dubbed it "the dreadful day"and it was the last time the Byzantine Empire exerted even minimal control over the whole of Anatolia.

The MWGrid project utilises an agent-based model of the Byzantine army's march to Manzikert to understand the context of how the Byzantine state supported the army and its progress, draw valuable conclusions regarding how the transport, taxation, agricultural production and military organisation systems interconnect and provide an insight into this pivotal historic event. The army (humans and animals) is modelled at a 1:1 scale to provide a plausible movement model, requiring tens of thousands of agents. To deal with the scale of such a model, distributed simulation is utilised.

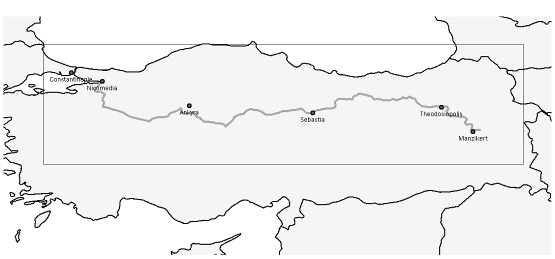

Fig. 1. Anatolia, extent of ABM and possible route

III. The MWGrid framework

A high-level view of the MWGrid framework is provided in figure 2. It can be seen as consisting of two major parts: the simulation system and the analysis environment.

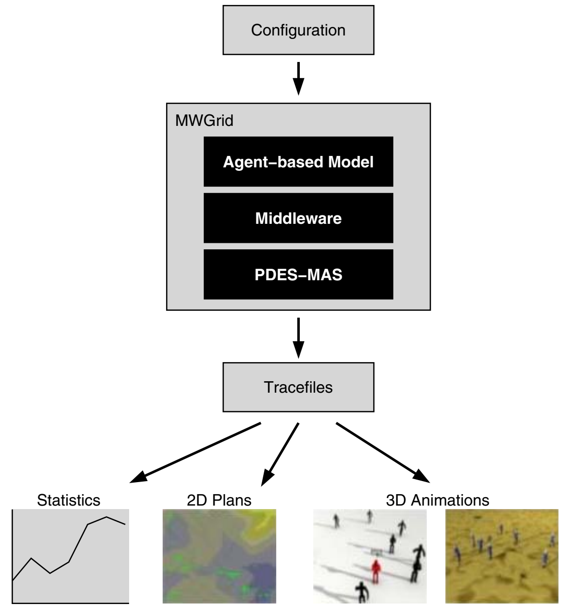

Fig. 2. The MWGrid framework

The simulation system executes the model and produces detailed trace files that are fed into the analysis system for off-line post-processing. This is achieved via a range of packages depending on the output required. Statistics can be produced detailing movement rates, food consumption, agent health status, amount of time spent on the move and the state of the environment after the army has moved on. These can involve individual agents or the aggregation of statistics of the whole army or certain subgroups. It is also possible to import a transformed trace-file into a 3D modelling package so that a 3D visualisation of the model can be created automatically.

The simulation system consists of three layers: the agent-based model (ABM); the PDES-MAS system; and the Middleware as an interface between the two.

IV. The model

The model consists of two main elements: the environment representing the terrain, infrastructure and resources of Anatolia; and the agents, representing the human and animal members of the Byzantine army on a one-to-one basis.

A. The environment

11^{th} century Anatolia represents a large and rich environment for an agent-based model, the distance from Constantinople to Manzikert is over 700 miles and contains a variety of different terrain types. The environment is represented in a series of slices, each slice dealing with a different aspect of the Anatolian landscape. Each slice can either be an array of values covering the whole of the ABM area or a list of locations with associated values, used for sparse data sets. Each cell of the environment is approximately 5 by 5 metres, for $25m^2$, giving a total size of 280,700 x 88,890 cells. To reduce memory requirements, two other resolutions are supported within the model for environmental data; 50 by 50 metres for $2{,}500m^2$ each and 500 by 500 metres for $250{,}000m^2$ each.

As the model is concerned with the practical aspects of feeding and moving the army, only certain characteristics of the historical environment are relevant to the model. These are:

- **Terrain**. The physical geography of Anatolia plays a large part in the process of route planning for the army. The highest quality terrain data is the ASTER GDEM data, a joint satellite mapping project by NASA and Japan's Ministry of Economy, Trade and Industry [20].
- **Roads**. Roads are an example of a sparse data set, a more efficient way of storing the data is by recording only the location of cells where roads are present. This can be done at 50 by 50 metre resolution in a file of around 3Mb. The Tabula Imperii Byzantini (TIB) maps [21] give locations for roads over the area that they cover but outside this area there are difficulties in extrapolating known data to fill in any gaps.
- **Settlements**. The army was supplied with food and equipment by the communities through whose territories it passed. This made settlements places to be visited along the march in order to pick up necessary supplies. Each settlement will have a surplus of food that, in addition to being affected by its surrounding landscape, is also affected by the time of year. The availability of grain is a commonly considered factor in the planning of military campaigns throughout antiquity and the amount of varies by considerable amounts [22]. To model variable food surplus throughout the year settlements are their own special type of agent in order to take advantage of each tick of the simulation to update each type of resource as availability will change throughout the year. As with roads, the TIB maps provide data for some areas.
- **Land-use**. The best guide currently available is modern land use as this is partly based on climate and terrain, both of which are largely similar to 11^{th} century conditions. The modern land-use data comes from the ESA's GLOBCOVER satellite programme [23]. However the same surveys and studies on the palaeo-environment and rainfall patterns (see [24][25][26] and others again) have been taken into account in the model's assumptions about vegetation and other factors relating to land-use where insufficient data was available from historical sources. As fine detail is unnecessary this data can be used at a resolution of approximately 500 by 500 metres per cell.
- **Water**. Availability of water is the single most important issue for the army, without it the army will not survive for long in the hot Anatolian summer. Modern data is used that has the benefit of containing flow rates, something unknowable for the 11^{th} century even if a map of Byzantine Anatolian rivers could be found.

B. The agents

The granularity and complexity of the agents is dictated by the project's aims. The practical problems involved with moving large numbers of people across broken terrain and through narrow ravines were known to medieval military leaders and terrain could greatly affect an army's progress [27]. Average movement rates have been calculated for armies based on historical itineraries but the relationship between army size and composition and the speed of march is one aspect that the model is ideally suited to address. With this in mind we have decided to model the people and pack animals of the army at a ratio of one agent per human or animal.

An agent consists of a plan queue, a messaging queue and a representation of a perception base along with a number of private and public variables modelling agent characteristics (figure 3). The plan queue contains a list of the tasks the agent has to perform. The message queue contains a list of messages from other agents, including orders from superiors and messages from comrades. The perception base contains the information that the individual agent has regarding the world, including information gathered by the agent's own senses and information introduced by communication with other agents.

Each agent has a series of variables, depending on

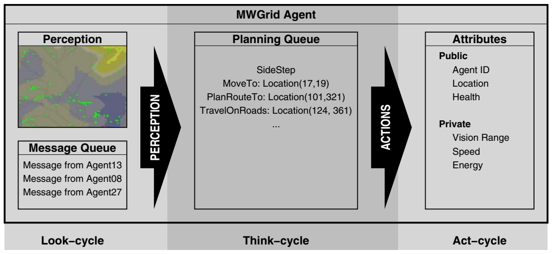

Fig. 3. The MWGrid agent

its type. These are dictated by the need to model the organisation of the army and its movement and the effects of the march on each individual agent. An agent's variables can be public (those that are apparent to other agents such as health and location) and private (speed, vision range). It can perform range queries to access the public variables of every agent within a certain distance. Orders are given when another agent passes a message to an agent with an order as the content. Provided the message comes from a valid source (a superior or a trusted comrade) the content of the message is added to the agent's plan queue.

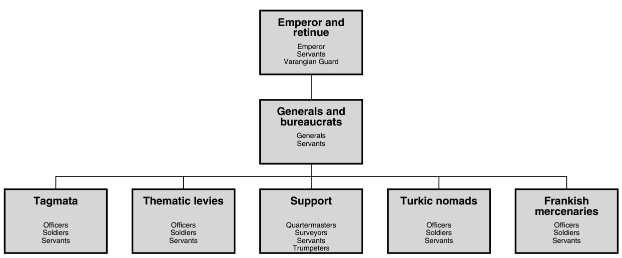

Fig. 4. Army (Agent) Organisation

The organisation of the Byzantine army provides a rigid hierarchy into which the agents can fit (figure 4). The Emperor occupies the apex of the pyramid and makes the decision of which route to take. His order to move, along with the location of the next day's camp, will be propagated down the command structure from officer to officer until it reaches the officers of each individual squad.

Agents lower in the hierarchy contain less decision making processes. The main task of each agent is to transport itself and any equipment it is responsible for from the previous camp to the next one, which, when reached, will need to be set up.

The Emperor's route planning is based on Probabilistic Roadmap (PRM) [28]. Once an overall route is planned by the Emperor, the settlements along the route are treated as waypoints between which an A* route planning algorithm can be used [29].

V. PDES-MAS

PDES-MAS is a framework for the distributed simulation of MAS [9]. It implements a Distributed Shared Memory (DSM) structure where the shared state, i.e., the visible and publicly accessible attributes or variables of the agents, is represented by Shared-State Variables (SSV), data-structures that store the time-stamped history of values of a particular variable over time [15]. Following the Parallel Distributed Event Simulation (PDES) paradigm, agents in the MAS are assigned to Logical Processes (LP), known as Agent Logical Processes (ALP), while SSVs are assigned to Communication Logical Processes (CLP). An ALP potentially models more than one agent, with multiple ALPs allowed concurrent access to the set of SSVs associated to the agents by connecting them to a tree-like network of CLPs. SSVs are then distributed among the CLPs in a scalable and balanced manner. Figure 5 shows a depiction of the PDES-MAS with 4 ALPs, 3 CLPs, and 4 SSVs.

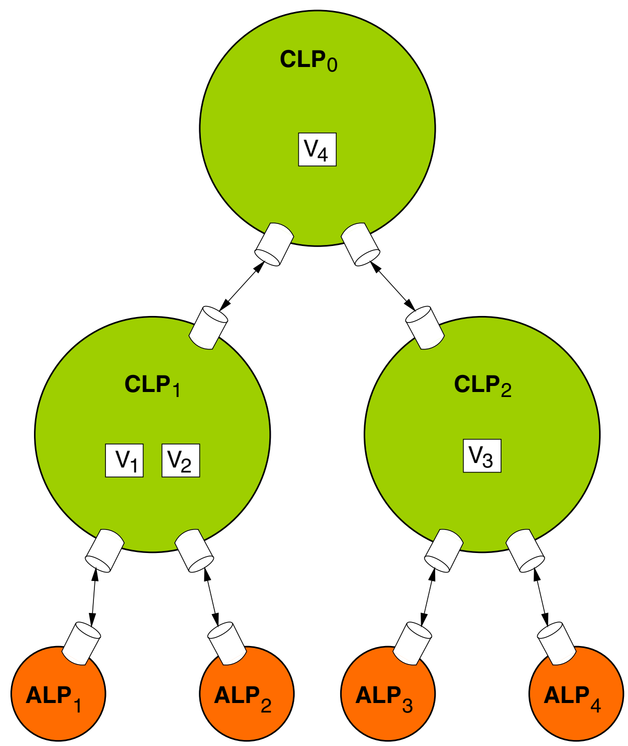

Fig. 5. The PDES-MAS Framework.

PDES-MAS allows SSV access through time-stamped events, i.e., within the sense-cycle it supports ID-query and Range-query events, while during the act-cycle it supports write-events. ALPs are connected to the leaf

CLP nodes in the tree-like structure, and an ALP issues SSV access events through its parent CLP. If the SSV is not assigned to the parent CLP, the access request is passed along the tree to the CLP where the SSV is located. The return information is then passed back along the same route in reverse, to the parent CLP and from there to the ALP.

Synchronising the events received from the ALP is the responsibility of the CLP. PDES-MAS uses optimistic synchronisation, the specifics of which have been reported in In general, each SSV is associated with a list of Write Periods (WP), representing the values of the variables at different times throughout the time progression of the simulation. When a WP is invalidated by a straggler write, any agents associated to the ALPs that have read that WP subsequent to the straggler write will be asked to roll-back to the time-stamp before the straggler write. An agent that is rolled back is then supposed to resume its time-progression from that time, using the now consistent data. As such, the data consistency is repaired for that time-stamp.

The CLP tree-like structure in PDES-MAS is reconfigured dynamically and automatically so as to reflect the interaction patterns of the agents as exhibited through the access patterns of the SSVs. SSVs that are accessed most frequently are pressured to move closer in the tree-like structure to the ALP handling that agent, i.e., towards the parent leaf CLP. The aim is to concurrently self-organise the SSVs in the tree so as to minimise the average number of hops required to access them, as well as to reduce the load imbalance between the CLPs. Reconfiguration of the CLP tree-like structure can be achieved by creating or deleting CLPs, moving ALPs to different parent CLPs, or by migrating SSVs between CLPs. In the PDES-MAS implementation used in this paper, the CLPs are configured in a fixed binary tree-structure with the leaf CLPs hosting a fixed and constant number of ALPs. Only SSVs are migrated through the tree to achieve redistribution.

VI. Middleware

The Middleware functions as the glue between the MAS simulation and the PDES-MAS framework and serves three overall goals: It provides a framework in which it is rela- tively straightforward to design and build a MAS; It trans- lates interactions between the agent's shared state variables into events for the PDES-MAS framework to handle; and it provides a means for the agents in the MAS to handle return events, from the PDES-MAS framework, like roll-backs, to take effect in the MAS. Figure 6 depicts the overall archi- tecture of the Middleware.

The Middleware supports the design and development of a MAS by providing an agent template. All agents in the MAS implement a *Distributed Object Template*. The Middleware itself acts as the MAS counterpart of the PDES-MAS framework's ALP with several agents or distri- buted objects handled by each ALP. The Middleware allows the agent to interact with variables in four different ways: add, read, write, and range-query. It utilises a scheduler which maintains a list of all agents or distributed objects on a single ALP in the PDES-MAS framework and call upon all agents in that list to perform its sense-think-act cycle encapsulated in a step method, thus providing time-progression and synchronisa- tion. Another aspect of the functionality of the Middleware layer is to provide initialisation, start-up, and eventual output collection of/from the simulation. The Middleware is described in more detail in [8].

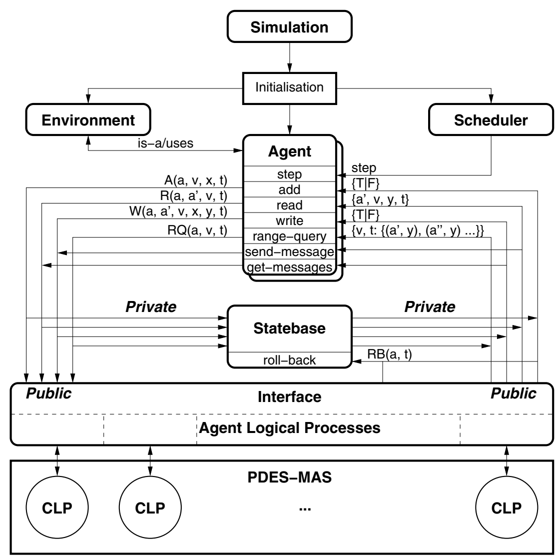

Fig. 6. The Architecture of the Middleware

VII. Results

For evaluation of the MWGrid system we have used two models: a test benchmark and a Manzikert model. The test benchmark was specifically devised to fully test the system (PDES-MAS system and the Middleware interface). All agents are the same, and all require four SSVs; two 'active' SSVs (current and previous location), and two 'inactive' ones as dictated by the software(Messages, and Class). For one time-step in the experiment, each agent: (a) polls its own location SSV to determine where in the environment it is (b) polls its environment through a range-query with range 5, i.e. a 25 location square with the agent location at its centre (c) stores its current location in the previous location SSV, and stores its new location, one to the right (increase X-coordinate by one) in this current location SSV. Throughout the values of the read SSVs and the range-query are validated. The agents are initialised in the location, in order of agent number, in a vertical line downward from the (0,0) coordinate in the environment. As such, agent 0 is initially located in location (0,0), agent 1 in location (0,1), and so on. The end-state of the experiment is a vertical line of agents from coordinate (0, max-time-step) downward. Throughout the experimental run rollbacks are initiated either from the mechanism of PDES-MAS itself (primarily state-migration), or because of the parallel nature of the execution and the use of the range-query [14]. The Manzikert model is based on the description outlined in section IV. Experiments were conducted for different number of army squads completing a day's march, ranging from 2-100, with each squad consisting of 10 agents (9 soldiers and one officer). In addition there is an Emperor agent and a number of officer agents reflecting the hierarchical structure of the army (i.e. one officer per 4 squads, plus one officer per 4 of those, and so on until the Emperor is reached). Experiments were executed for different architecture(i.e. PDES-MAS tree) sizes and different numbers of agents. Agents range from 10 to 10000 split evenly across ALPs Architecture sizes include 3-4, 7-8, 15-16 and 31-32 CLP-ALP configurations yielding 7, 15, 31 and 63 total LPs (mapped on separate CPU cores). The Manzikert experiment was run with a representative architecture size of 31. The platform used for the experiments is the University of Birmingham Blue Bear system[1]. The part of the system used in these experiments consitied of 64-bit 2.6 GHz dual-processor dual-core (4 cores/node) AMD Opteron 2218 worker nodes with each node having 8 GB of memory. The system is based on Scientific Linux 5.2 and has a total of over 150 TB (raw) disk space allocated using IBM's GPFS cluster file system.

Figure 7 illustrates the wall clock time used by each experiment against the number of SSVs for these experiments. The graph shows that the wall-clock duration of the experiments increases roughly exponentially with the size of the experiment. As expected small experiments with few SSVs and agents take much shorter to finish than bigger experiments with more SSVs and agents. The resources needed to run the scenarios impose a

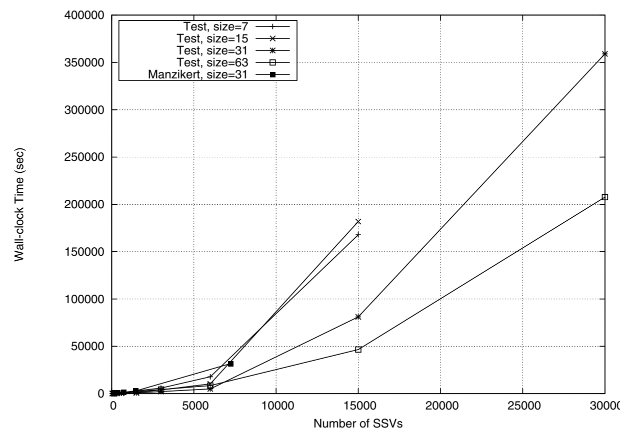

Fig. 7. MWGrid performance: Wall Clock Time

threshold on how large an experiment can be run on a given architecture. Experiments beyond 15,000 SSVs can not be run on architecture sizes of 7 and 15 LPs. Especially memory use is a bottleneck here. The cumulative memory available to architecture sizes with 7 and 15 LPs (one LP per CPU-core, with 4 cores available per CPU, resulting in 2 CPUs and 4 CPUs respectively) is not sufficiently large to run these bigger experiments. Of interest is the different curvature for architecture sizes 31 and 63. They indicate most clearly that wall-clock time is significantly reduced when the system is provided with more resources, in particular memory.

It is worth noting that the middleware interface,involving translation between Java to C++ and back to Java, incurs a fixed amount of overhead, independent from the scale of the experiments. The graph in figure

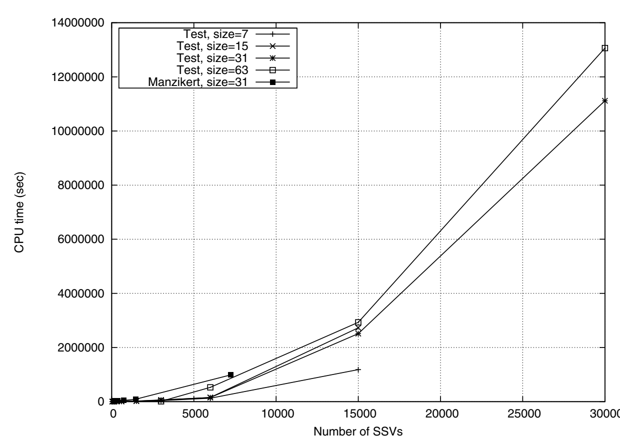

Fig. 8. MWGrid performance: CPU Time

8 shows the cumulative CPU-time for different experiments and architecture sizes. The CPU-time measure in this instance is the cumulative amount of time used by these experiments over all CPU (cores), excluding the

[1]http://www.bear.bham.ac.uk/bluebear/

time needed to do IO or network interaction. Of note here is that the curves all remain quite close together. Only for small architecture sizes and small number of CPUs is the cumulative CPU-time significantly lower for close-to-the-threshold experiments than for larger architecture sizes. Of note is also the difference between the results for large architecture sizes for the test experiments, in particular those for 30,000 SSVs. In general the graph shows the same exponential curvature as was seen in the graph in figure 7.

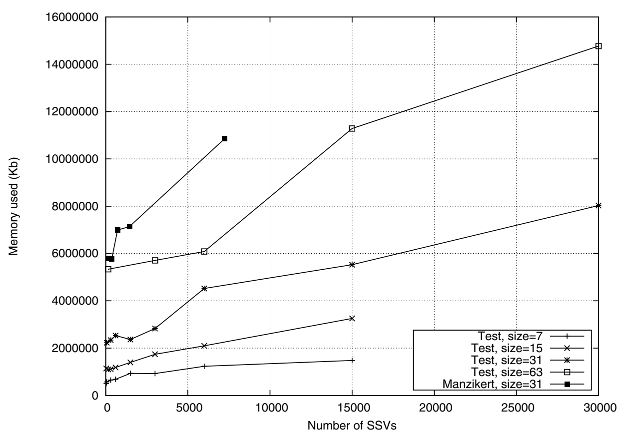

Fig. 9. MWGrid performance: Memory Usage

The graph in figure 9 shows the cumulative physical memory usage for the different experimental setups. Larger architecture sizes mean more CPU and thus physical memory available to the system. The overall trend in the graph shows that for the test experiment, for different architecture sizes, for increasing numbers of SSVs, the system uses progressively more memory to successfully finish the experiment. Of note though is that the increase is roughly linear, though with a large proportional increase for experiment increase (number of SSVs) for each architecture size. The Manzikert experiment remains the odd-on-out, as the more complex model requires more memory to begin with, increasing more rapidly when the size of the experiment is increased as well. Comparison with figures 7 and reffig:cputime however shows that a roughly exponential increase in duration for increased sizes of the experiment only results in a roughly linear increase in memory usage for those same experiments. We take this to indicate a scalable implementation of the MWGrid system.

VIII. Summary and Future Work

Digital Humanities offer a new exciting domain for agent-based distributed simulation. MWGrid is the first project to utilise this approach in this domain. The project has delivered a large scale agent-based model designed to study medieval military logistics in general, and the Manzikert campaign (as a case study) in particular. As an important historical event, the Manzikert campaign has been described in detail in historical record, providing well established focal points that can be used to validate the model's outputs. These outputs in turn can be used to define parameters within which the historical evidence can be framed and which would enable historians to evaluate existing theories.

MWGrid has also exemplified the utilisation of the PDES-MAS kernel for realistic large scale agent-based models. PDES-MAS implements a DSM paradigm where all agents' interactions are realised through accessing shared state variables. Translating model-level agents' actions to timestamped read/write events to be processed by PDES-MAS (or any other DSM-based kernel for that matter) is a challenging problem. The Middleware layer of MWGrid is an attempt to address this issue. It provides a transparent, layered mechanism to perform this translation and an agent template that can be extended to straightforwardly implement the agents in a MAS.

This paper has presented the first quatitative evaluation of the MWGrid system. Results obtained so far hasve confirmed the scalability of the PDES-MAS system. For smaller models, small tree sizes deliver better performance keeping the communication and management overhead of the tree low. For bigger agent models, better performance is achieved with increasing tree sizes - up to a point when the overhead of the tree becomes the dominant factor. More experiments are currently being conducted to enable an in-depth performance analysis of the different components of the system for different and larger model scenarios.

References

[1] M. Lake, "Magical computer simulation of mesolithic foraging," in *Dynamics in Human and Primate Societies: Agent-Based Modelling of Social and Spatial Processes*, T. Kohler and G. Gumerman, Eds. New York: Oxford University Press, 2000, pp. 107–143.

[2] T. Kohler, C. D. Johnson, M. Varien, S. Ortman, R. Reynolds, Z. Kobti, J. Cowan, K. Kolm, S. Smith, , and L. Yap, "Settlement ecodynamics in the prehispanic central mesa verde region," in *The Model-Based Archaeology of Socionatural Systems*, T. Kohler and S. van der Leeuw, Eds. Santa Fe: School for Advanced Research Press, 2007.

[3] M. Lake, "Agent-based modeling: a question of scale," 2007, http://www.nesc.ac.uk/talks/772/Lake_e_sci_07_v1.pdf.

[4] J. Haldon, *General issues in the study of medieval logistics: sources, problems, methodologies*. Brill, 2005.

[5] B. Craenen, G. Theodoropoulos, V. Suryanarayanan, V. Gaffney, P. Murgatroyd, and J. Haldon, "Medieval military logistics: A case for distributed agent-based simulation," in *DIstributed SImulation & Online gaming (DISIO 2010)*. ICST/CREATE-NET, March 2010.

[6] B. Craenen and G. Theodoropoulos, "Interfacing multi-agent models to distributed simulation platforms: The case of pdesmas," in *Winter Simulation Conference (WSC 2010)*, December 2010.

[7] P. Murgatroyd, B. Craenen, G. Theodoropoulos, V. Gaffney, and J. Haldon, "Modelling medieval military logistics: an agent-based simulation of a byzantine army on the march," *Computational and Mathematical Organization Theory*, pp. 1–19, 10.1007/s10588-011-9103-9. [Online]. Available: http://dx.doi.org/10.1007/s10588-011-9103-9

[8] G. T. Bart Craenen, Vinoth Suryanarayanan, "A middleware for interfacing with simulation systems of multi-agent models," in *ICST-CREATE-NET 2nd Workshop on DIstributed SImulation and Online gaming (DISIO 2011), in conjunction with SIMUTools 2011 (DSRT 2005)*, Barcelona, Spain, Mar 2011.

[9] B. Logan and G. Theodoropoulos, "The distributed simulation of multi-agent systems," in *Proceedings of the IEEE*, vol. 89(2), 2001, pp. 174–186.

[10] A. Chen, D. Ewald, R. Theodoropoulos, G. Minson, R. Oguara, T. Lees, M. L. B., and Uhrmacher, "Data access in distributed simulation of complex systems," *Journal of Systems and Software*, vol. 81, no. 12, pp. 2345–2360, Dec 2008.

[11] M. Lees, B. Logan, and G. Theodoropoulos, "Using access patterns to analyze the performance of optimistic synchronization algorithms in simulations of mas," *Transactions of the Society for Computer Simulation International*, vol. 84, pp. 481–492, 2008.

[12] ——, "Analysing probabilistically constrained optimism," *Concurrency and Computation: Practice and Experience Journal, special issue on Distributed Simulation, Virtual Environments and Real Time Applications*, vol. 21, pp. 1467–1482, Aug 2009.

[13] T. Oguara, D. Chen, G. Theodoropoulos, M. Lees, and B. Logan, "An adaptive load management mechanism for distributed simulation of multi-agent systems," in *The 9-th IEEE International Symposium on Distributed Simulation and Real Time Applications (DSRT 2005)*, Montreal, Oc. Canada, Oct 2005, pp. 179–186.

[14] V. Suryanarayanan, R. Minson, and G. Theodoropoulos, "Synchronised range queries in distributed simulations of multi-agent systems," in *13th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2009)*, Singapore, Oct 2009.

[15] M. Lees, B. Logan, and G. Theodoropoulos, "Adaptive optimistic synchronisation for multi-agent simulation," in *Proceedings of the 17th European Simulation Multiconference (ESM 2003)*, D. Al-Dabass, Ed. Delft: Society for Modelling and Simulation International and Arbeitsgemeinschaft Simulation, Society for Modelling and Simulation International, 2003, pp. 77–82.

[16] M. Lees, B. Logan, C. Dan, T. Oguara, and G. Theodoropoulos, "Decision-theoretic throttling for optimistic simulations of multi-agent systems," in *Proceedings of the Ninth IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2005)*, A. Boukerche, S. J. Turner, D. Roberts, and G. Theodoropoulos, Eds. Montreal, Quebec, Canada: IEEE Press, Oct 2005, pp. 171–178.

[17] ——, "Analysing the performance of optimistic synchronisation algorithms in simulations of multi-agent systems," in *Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 37–44.

[18] M. Lees, B. Logan, and G. Theodoropoulos, "Time windows in multi-agent distributed simulation," in *Proceedings of the 5th EUROSIM Congress on Modelling and Simulation (EuroSim04)*, Paris, Sep 2004.

[19] T. Oguara, D. Chen, G. Theodoropoulos, B. Logan, and M. Lees, "An adaptive load management mechanism for distributed simulation of multi-agent systems," in *Proceedings of the 9th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2005)*, A. Boukerche, S. J. Turner, D. Roberts, and G. Theodoropoulos, Eds. Montreal, Quebec, Canada: IEEE Press, Oct 2005, pp. 179–186.

[20] C. I. o. T. NASA Jet Propulsion Laboratory, "http://www2.jpl.nasa.gov/srtm/."

[21] H. Hellenkemper and F. Hild, *Lykien und Pamphylien*. Wien: Verlag der Österreichischen Akademie der Wissenschaften, 2004.

[22] D. Engels, *Alexander the Great and the Logistics of the Macedonian Army*. University of California Press, 1978.

[23] P. Bicheron, P. Defourny, C. Brockmann, L. Schouten, C. Vancutsem, M. Huc, S. Bontemps, M. Leroy, F. Achard, M. Herold, F. Ranera, and O. Arino, *GLOBCOVER Products Description and Validation Report*, 2008, http://ionia1.esrin.esa.int/docs/GLOBCOVER_Products_Description_Validation_Report_I2.1.pdf.

[24] N. R. M.D. Jones and M. Leng, "Quantifying climatic change through the last 21,000 years in central turkey based on lake isotope palaeohydrology," *Quaternary Research*, vol. 67, pp. 463–473, 2007.

[25] M. H. R. Touchan, U. Akkemik and N. Erkan, "May-june precipitation reconstruction of southwestern anatolia, turkey during the last 900 years from tree rings," *Quaternary Research*, vol. 68, pp. 196–202, 2007.

[26] M. Jones, C. Roberts, M. Leng, and M. Türkeş, "A high-resolution late holocene lake isotope record from turkey and links to north atlantic and monsoon climate," *Geology*, vol. 34, pp. 361–364, 2006.

[27] G. Dennis, "Three byzantine military treatises," in *Dumbarton Oaks Texts*. Washington: Dumbarton Oaks Research Library and Collection, 1985, vol. 9.

[28] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[29] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics SSC4*, vol. 4, no. 2, pp. 100–107, 1968.