gUML: Reasoning about Energy at Design Time by Extending UML Deployment Diagrams with Data Centre Contextual Information

Nigar Jebraeil, Adel Noureddine, Joseph Doyle, Syed Islam and Rabih Bashroush
School of Architecture, Computing and Engineering,
University of East London, United Kingdom
Email: u0950210@uel.ac.uk, a.noureddine@uel.ac.uk, j.doyle@uel.ac.uk, syed.islam@uel.ac.uk, r.bashroush@qub.ac.uk

***Abstract*—With the rising energy demand in ICT services and its associated environmental impact, the need for energy efficient Enterprise ICT solutions is growing. As data centres account for a large part of energy consumption in ICT, data centre operators strive to create opportunities to put more emphasis on reducing energy consumption. However, creating ICT Systems that are energy efficient by design remains a key challenge. In this paper, we identify and map contextual energy information about data centre operations in order to model their power related components. This contextual modelling is then mapped to deployment diagram where we introduce `greenUML (gUML)`, an extension to UML diagrams to improve energy efficiency through energy analysis at design time. `gUML` will allow system architects to reason about the energy footprint of their applications at design time.**

I. Introduction & Motivation

Tackling climate change and achieving low carbon emission have been an international priority over the past three decades. According to The Centre for Energy-Efficient Telecommunications (CEET), the energy consumption of ICT could exceed the global power supply by 10-15% [1]. Additionally, the ICT industry, which delivers the Internet, voice, video and other cloud services, creates more than 830 million tons of carbon dioxide (CO2) on a yearly basis, which counts for about 2 percent of global CO2 emissions [2]. This exceeds the emissions of the entire aviation industry [3]. Yet, this number is expected to double by 2020. This trend is clearly not sustainable. For instance, Japan is expected to spend all its energy capacity to support its ICT energy needs by 2030 if the current trend continues [4].

There has been considerable research to try and tackle this problem [5], [6], [7]. To the author's knowledge, however, a comprehensive view of the energy consumption of software in the design and execution lifecycles has not been proposed. While improvements to power management and load balancing solution can reduce the energy consumption of data centres, these improvements are severely limited unless this information can be communicated to software engineers so that appropriate designs and deployments can be created.

In this paper, we present our approach to bring energy related contextual information to design time. In the example of data centres, we identify the main energy consuming components, classify and organise them in an architectural view, then map these contextual information to UML deployment diagram. This mapping helps developers write energy efficient software by reasoning about energy and identifying energy concerns when designing their applications. This paper makes the following contributions:

- The modelling of contextual information related to power components in data centres.
- The extension of UML models with green contextual information in new extension called `gUML`.
- The evaluation of how `gUML` can be used to reduce energy consumption and carbon emissions when designing workload deployment. Our evaluation shows that energy consumption can be reduced by 21% and carbon emissions can be reduced by 92% over the traditional deployment strategy.

The paper is structured as follows. Section 2 discusses the related literature. In section 3, we discuss the conceptual mapping of the main layers of a data centre. Section 4 presents the extended UML deployment diagram. Section 5 provides an evaluation using an example workload. Finally, section 6 concludes the paper.

II. Related Work

Other works have been proposed to assist developers at designing energy-aware software. In [8], a software framework is proposed to transform applications based on developers' input and the energy profile of these applications. However, this approach requires manual input and studying multiple variations of the application in order to achieve efficiency transformations. Kwon et al. [9] presented guidelines to help developers select distributed programming abstractions to satisfy energy constraints. Kumar et al [10] use Data Envelopment Analysis to solve environmental decisions making problems. In [11], an HPC based cloud model is proposed to tackle energy optimisation at runtime. Bi et al. [12] presented an SLA-based approach to optimise resources in a virtualised environment in data centres. Cohen et al. [13] proposed a programming model in a type-based approach to help developers reason and promote energy efficient software. Their approach enables developers to specify phases and modes. The former represents program workloads while the latter represent required energy states. Finally, in [14], the authors

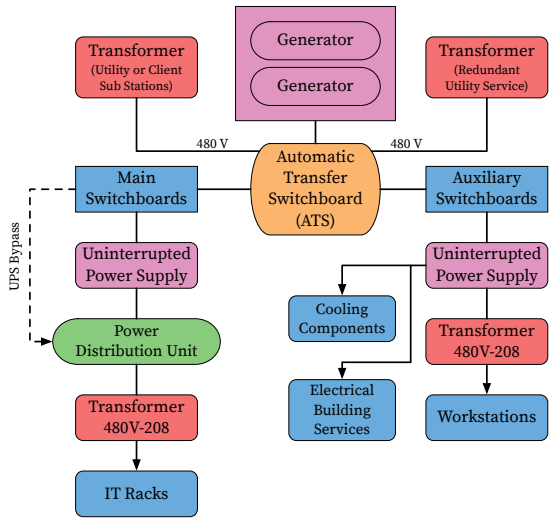

Fig. 1. Typical power flow architecture in a data centre

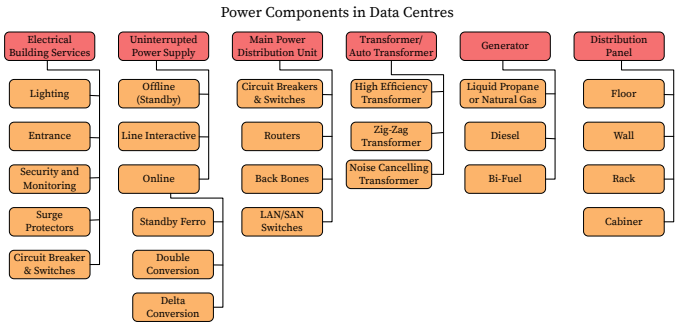

Fig. 2. Power components of a data centre

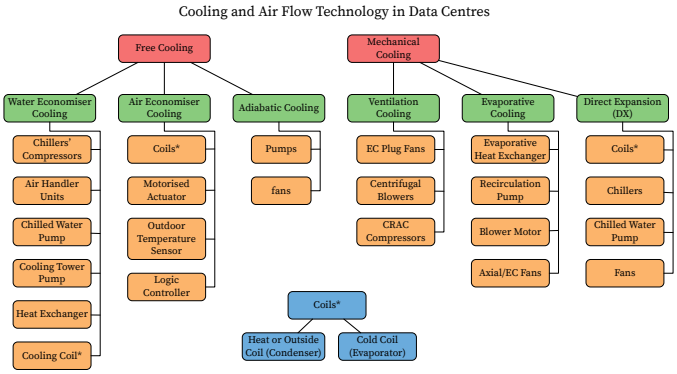

Fig. 3. Power consuming cooling components of a data centre

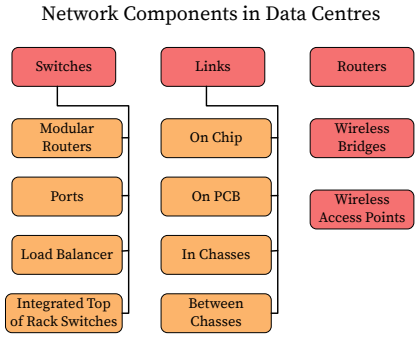

Fig. 4. Power consuming network components of a data centre

show how software developers can achieve energy savings by choosing energy efficient APIs along with optimal parameters in a number of use cases.

These approaches provide indications and preliminary steps into energy-aware software design. We aim to propose a comprehensive view of the energy consumption of software throughout their implementation, deployment and execution life-cycles. By proposing energy-aware contextual information modelling and an extension to the UML deployment diagram, software developers can better reason about software energy efficiency at design time and consider energy consumption at individual component and interaction level.

III. Contextual Information Modelling

In our work, we present a deployment UML diagram, called gUML, as an extension to UML deployment diagram. We add relevant contextual information to produce an energy model. The research process we have chosen in the design of gUML is the design science research process [15]. Accordingly we need to follow the model described in [15] namely:

- *Problem identification and motivation*. The problem is that energy consumption information is not available in UML deployment diagrams. Without this information software engineers are unable to establish the energy consumption of deployed software systems and make informed design decisions to minimise energy consumption. Energy aware software is a key component in delivering sustainable ICT services.
- *Objectives of a solution*. A solution should provide information on the energy consumption of components to software engineers so that design decisions and modifications can be implemented to deliver sustainable ICT services.
- *Design and development*. gUML extends UML deployment diagrams by labelling each component with a tuple which indicates the energy consumption of the component in the server power, UPS and cooling layers of the data centre.
- *Demonstration and Evaluation*. The effiacy of the solution is detailed in Section IV-B.

With this methodology defined the next step is to identify and capture the relevant data. Our first step is to produce a general map of power flow throughout data centres. In this section, we identify and classify the relevant information needed for our gUML model in the areas of power, cooling, and network components. A traditional UML diagram can model a variety of system characteristics such as activities, components, interactions and user interfaces to show the structure, behaviour or interactions that exist in a system. Our proposal augments this by placing energy information on the links between the

nodes of the UML diagrams so that the energy consumption of design decisions can be easily visualised.

To ensure that the energy information associated with design decisions in gUML is accurate it is essential to identify the most power hungry components of a data centre. Figure 1 illustrates a popular example journey of power flow amongst all levels of a data centre from its entrance point all the way through to its servers. A further breakdown of the most power-consuming components of the data centre as well as the ratio of the power component power consumption to its performance and its dependency to other components are illustrated in Figures 2, 3, 4 showing the power, cooling and networking component layers respectively. In the next sections we will discuss each of these layers in relation to gUML.

A. Identifying the Power Components of Data Centres

The distributed power is consumed unevenly amongst the six main power component branches that are known as the main streams of power consumption in data centres. Servers and conventional hardware devices currently benefit from energy efficiency solutions such as Dynamic Component Deactivation. These techniques were initially introduced to improve energy consumption in mobile devices. However, servers are rarely in idle mode. They have an average utilisation rate of 10% to 50%. This results in a considerably poor performance in terms of energy efficiency [16]. Therefore, in order to harness power at hardware level, methods such as *Dynamic Power Management* (DPM) techniques are applied. These techniques include Dynamic Component Deactivation (DCD) and Dynamic Performance Scaling (DPS). DCD techniques are created based on the idea of an idle mode at the stage of inactivity. In addition, computer components that can dynamically adjust their performance in regards to power consumption can apply different techniques of DPS. Some components, such as the CPU, can adjust clock frequency rather than shutting down completely. This technique lead to the proposal of Dynamic Voltage and Frequency Scaling (DVFS), a technique widely used and supported by modern processors and operating systems [17], [18], [19]. Thus, to accurately model the energy consumption of a system in gUML, information on the utilisation of power saving techniques in a data centre must be gathered and incorporated into the model so that design decisions accurately reflect the performance of the system.

B. Identifying the Cooling Components Power Consumption

Cooling management is traditionally considered the largest energy overhead in data centres due to the vast amount of heat produced by IT equipment. The power consumed to cool a data centre is between 30% and 50% of the total power consumption. This number has the potential to increase depending on the IT performance management and geographical situation. Therefore, applied cooling could limit the capacity of the data centre. Despite their remarkable *capacity management* capabilities, high density computing and workload consolidation are amongst the two most power hungry techniques applied to IT, which immensely affect the level of required cooling capacity [20], [5]. According to American Society of Heating, Refrigerator and Air-Conditioning Engineers' guideline [21], there is a broad range of standards introduced for optimal temperature and humidity in data centres. The cooling cost will rise against the cold air supplement. The cooler the data centre environment, the more power will be consumed, hence all best practices and cooling strategies propose to increase and maintain the operating temperature to its highest permissible value and reduce power assigned to cooling, humidity and heat removal in order to achieve an improved PUE [5], [22]. Figure 3 illustrates the model aimed to identify the cooling components that distress power consumption of data centres. Additionally, data centres employ techniques such as aisle containment [23] and the use of air-side economisers known as"free air cooling" [24] to reduce the energy consumption of the data centre. All of these factor are integrated into the gUML to accurately predict the energy consumption of the system in different design configurations. It should be noted that the heat load distribution, ACU flow and the general cooling behaviour of data centre that can be calculated by applying a typical Computational Fluid Dynamics (CFD) analysis [25], [26] and the results of the CFD simulation can be integrated into the gUML model.

C. Identifying the Network Components of Data Centre

While the power consumed by networking equipment is typically less than power required to cool a data centre, it is still significant. A Data Centre Network may consist of thousands of servers on site. These servers are connected in a wide variety of topologies but fat-tree networks are becoming increasingly popular. In a fat tree network the topology is built from a combination of identical switching elements, so relying on aggregation to higher-speed; more expensive switching elements is not unnecessary [6]. Nevertheless, the key feature to supply the requirements of huge bandwidth capacity and high-speed communications for Data Centre Network is to design an efficient interconnecting architecture [27], [28]. The particular network architecture used by the data centre is integrated into the gUML model so that the energy consumption of networking components can be accurately predicted by system architects.

D. Mapping Power to Workflow

The final factor that is considered in the gUML model is how the workflow is processed. This will greatly affect the energy consumption and as suggested by [29], modelling and design ought to be based on the substitution between energy consumption and other requirements. We present in Figure 5 an example of a system whose energy consumption is modelled under the gUML proposal. Each energy consuming component is linked to associated energy consuming components. For example, energy consumed by a server should be linked to a UPS component and a cooling component as power for the server must pass through the UPS before it can reach a server and cold air from the cooling component will be required to keep the server at a sufficiently low temperature.

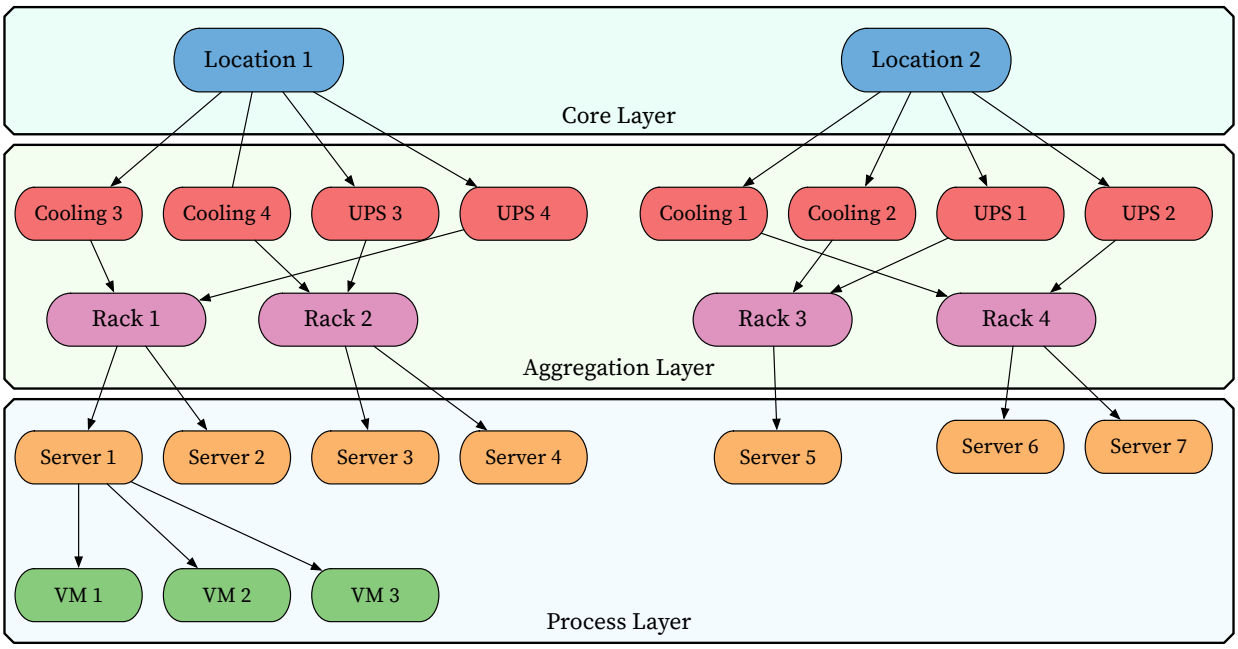

Fig. 5. The map of end to end workload flow

Thus, energy cannot be consumed at the server without power being supplied to these components and this must be reflected in the gUML model. This mapping could be then utilised to allocate the workflow towards its most efficient route. This model makes the contextual information available at the design level by extending the UML deployment diagram. This will allow software architects to better reason about the energy implications of their design reasoning by showing them where and how their software should be deployed.

Based on our modelling and identification of energy-related contextual information in data centres, we extend the traditional UML deployment diagram. The result is our extension, gUML. It builds on energy contextual information to help guide software developers in designing energy-aware applications. The next section describes our gUML extension.

IV. gUML Extended Deployment Diagram

This section presents the design of an extension to the Unified Modelling Language (UML), called gUML. It proposes a view of a holistic approach and is designed to address the workload power consumption in the most efficient way. This will enable data centres to efficiently reduce the amount of energy consumption without having to threaten SLAs or the performance of the entire system. gUML collects the data of each workload's CPU consumption, cooling consumption, and bandwidth consumption. Creating an energy efficient map for the workload to flow within the complex numerous systems requires accurate addressing amongst different levels and stages of workflow, as well as real-time communications that link these layers.

In order to draw an energy-aware UML Diagram, it is necessary to identify the key target parameters of this model. In this case, the model is designed with the target revolving around efficiency, high performance, maintenance, and scalability. To apply all above criteria, a UML deployment diagram is chosen which can best serve the purpose for this model. UML is specifically chosen for its customisation abilities that allow us to deploy the existing modelling tools in order to define our domain, while it is convenient for the end users to leverage the extension.

A. UML Deployment Diagrams

A typical UML deployment diagram models the actual deployment of software components into hardware nodes. It illustrates the configuration of the hardware components (nodes) as well as how software components and artifacts are mapped onto those nodes. Currently, these deployment models lack energy-related information. This is problematic in data centre environments where multiple factors and components have various effects on energy consumption (see Section III).

For instance, efficiently deploying an application to multiple nodes can be achieved by understanding in which server each virtual machine is installed, what rank are the servers installed, what UPS system is used for each server rack, and what cooling techniques or power transformation units are used for these servers and racks. Deploying two components of an application on two virtual machine may lead to varying energy footprint if these two VMs were installed in separate racks, data centres or cooled using different systems.

In our approach, we extend the diagrams with contextual information about the energy consuming hardware (*e.g.*, power generation, cooling, servers, etc.) in data centres. With this information available at design time, software developers can

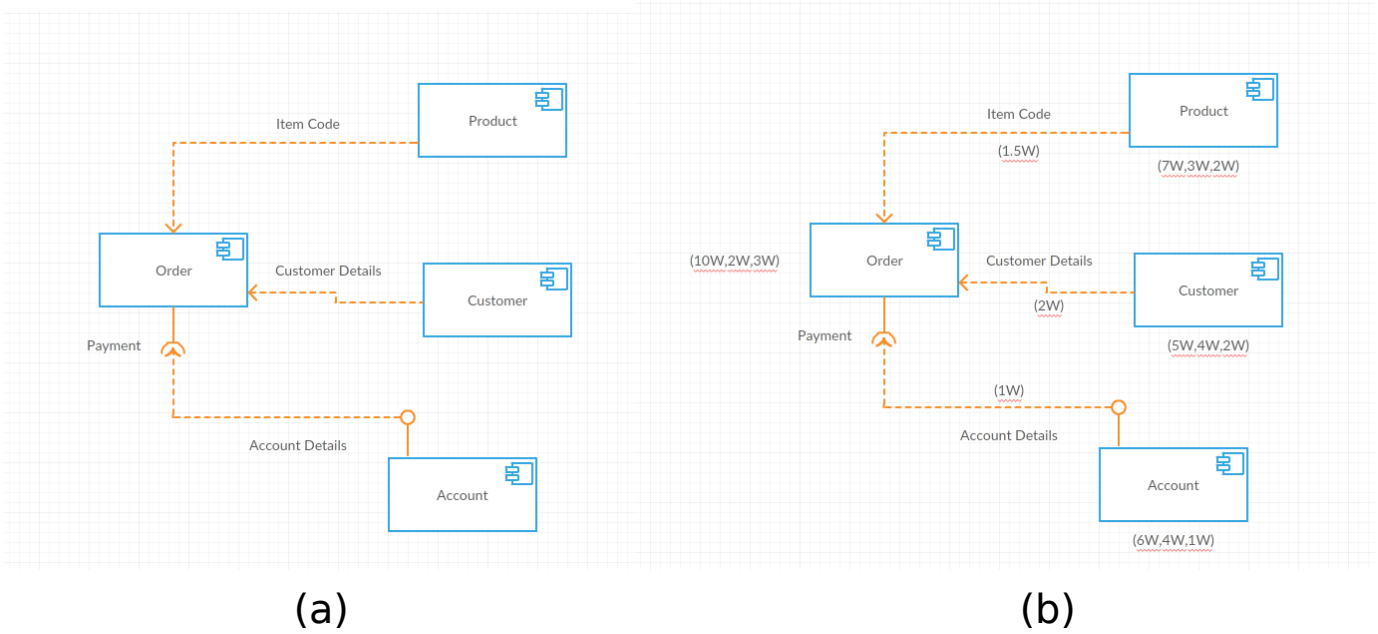

Fig. 6. A typical UML component diagram and its gUML counterpart.

design the different components of their applications with knowledge about energy beyond computational power.

Figure 6(a) illustrates a typical UML component diagram for a simple order system which consults customer, product and account systems to process orders. Figure 6(b) illustrates the gUML counterpart of this diagram. Each component is labelled with a tuple which indicates the energy consumption of the component in the server power, UPS and cooling layers of the data center[1] and each connection is labelled with the energy consumption necesscary to communicate with different components. This will mainly relate to network energy consumption. These figures will be entered by hardware systems architects and will help visualise energy aware software design through greater collaboration between software and hardware architects.

B. Validation

In order to illustrate how gUML can be used to redesign workload placement in an enterprise information system we present a worst case example of a High Performance Computing workload placement and a best case example which can be designed with the aid of the gUML diagram. We assume that the workload is a daxpy or LINPACK like workload which are representative of power hungry HPC workloads [30]. We assume that the total number of FLOPs required to complete the workload is 1TFLOP. We also assume that each virtual machine has a throughput of 250MFLOP/s. We assume that each physical machine can host 4 virtual machines and that each virtual machine processing the workload causes an increase of 10W to the idle power of the physical server whose

[1]This can be extended to incorporate additional energy consumption layers as necesscary.

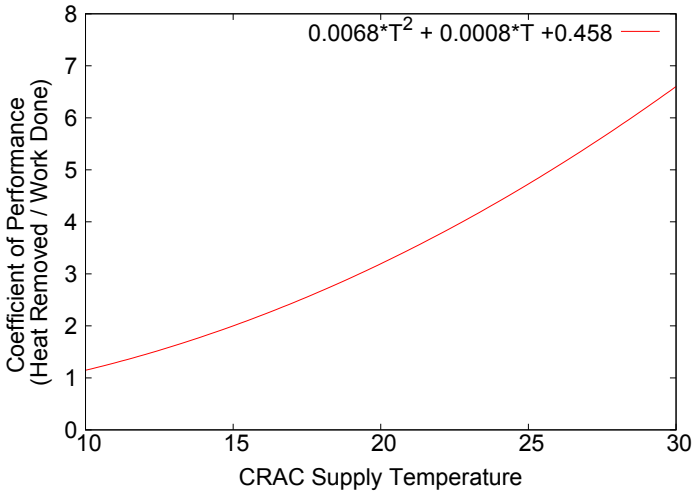

Fig. 7. Typical COP of water chilled computer room air conditioner.

idle power is 140W. We assume that the desired completion time of the workload is 5 minutes meaning that 14 virtual machines are required to complete this workload in the desired time. This workload data is based upon the experimentation of Verma *et al.* [30]. The workload can be sent to two data centres. One located in Sweeden which uses free air cooling and has a carbon intensity of 32g/kWhr. The other is in Germany which uses cold aisle containment cooling and has a carbon intensity of 570g/kWhr [31]. The network topology of both data centres is a two tier fat tree network [32]. We assume that the supply temperature of the cooling system in Germany is 20°C. All of the virtual machines communicate with each other to execute the workload.

For this particular workload the worst case scenario occurs

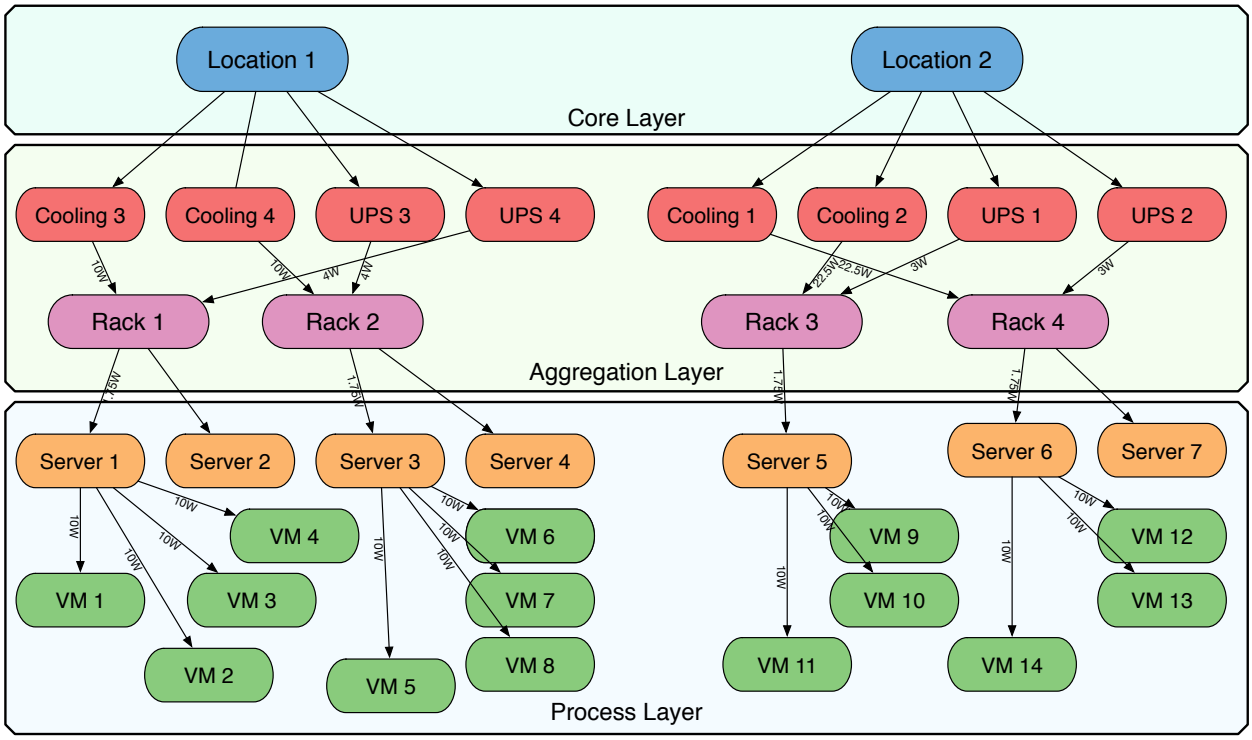

Fig. 8. Diagram of power consumption in worst case scenario

when the virtual machines are distributed among the physical servers at different physical locations. This could easily occur if a traditional global load balancing system such as the least connection method is used. This scenario is depicted in Figure 8. The cooling cost of the data centers is calculated using the following formula:

$$C = \frac{Q}{COP(T_{sup})} + P_{fan}$$

Where Q is the amount of power the servers consume, T_{sup} the temperature of the air that the CRAC units supply, P_{fan} the power required by the fans of the CRAC units and COP is the coefficient of performance (COP), that is the ratio of heat removed to work necessary to remove the heat, is a function of the temperature of the air being supplied by the CRAC unit. The COP of a typical chilled-water CRAC unit used in the calculations of cooling costs is depicted in Figure 7. In the case of the Swedish data centre free air cooling is used and the only power requirement is for the fans. The network power consumption is calculated by assuming that the physical machines utilise 1Gbps ethernet ports to connect, calculating the number of ports required to communicate with the other physical machines[2] and assuming that the power value required to open a port is 0.7W. This is the mid-range value from those presented in [7]. The UPS power consumption is assumed to be 10% of the dynamic power supplied to the physical machines which is typical of power losses during normal operation [33]. We could also consider other components such as transformers which operate at the highest efficiency when the load is in the 50-75% range and whose naive use can result power losses in the 60-80% range [34] but this is left for future work.

The energy consumption of the software components implementing the HPC workload would be illustrated in a `gUML` diagram similar to Figure 6(b). By seperating each of the energy components into tuples it is easy for the software engineer to identify potential power hungry components in the hardware which supports the software deployment. The deployment can then be redesigned in consultation with the data center architect to create a more energy efficient mapping between software and hardware components. Thus, the software engineer of the workload in collaboration with the data center architect would be able to redesign the load balancer to achieve the best case scenario which is depicted in Figure 9.

[2]In some cases the ports are shared by the physical machines and the power consumption is split between them.

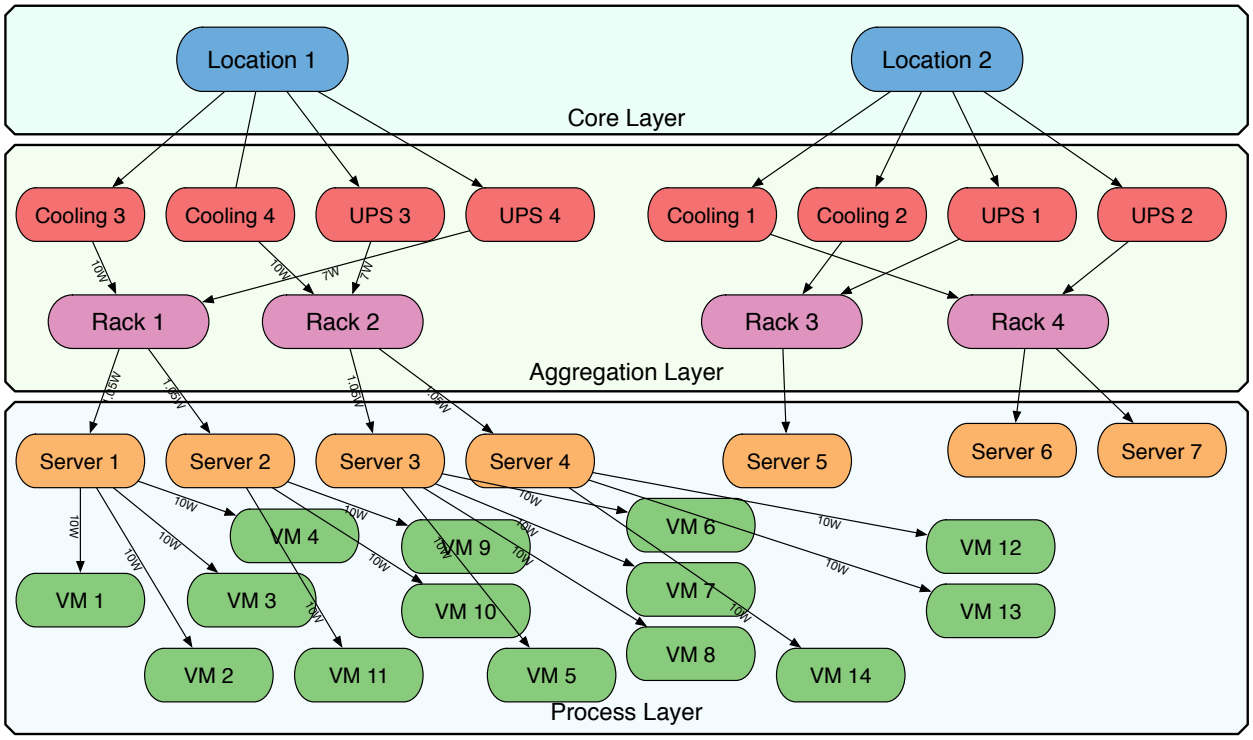

Fig. 9. Diagram of power consumption in best case scenario

TABLE I
ENERGY CONSUMPTION AND CARBON EMISSIONS FOR TWO SCENARIOS WITH PERCENTAGE REDUCTION OF BEST CASE SCENARIO OVER WORST CASE SCENARIO

Scenario	Virtual Machine Energy (kJ)	Cooling Energy (kJ)	Networking Energy (kJ)	UPS Energy (kJ)	Total Energy (kJ)	Total Carbon Emissions (g)
Worst Case	42	19.5	2.1	4.2	67.8	20.6
Best Case	42	6	1.3	4.2	53.5	1.7
Percentage Reduction	0%	69%	38%	0%	21%	92%

In the best case scenario, the virtual machines are consolidated onto physical machines in the Swedish data centre where the cooling power consumption is lower due to the use of free air cooling. In addition, the networking power consumption is lowered as less networking link are required and thus, ports can be switched off. A comparison of the power consumption and environmental impact of the two scenario are depicted in Table I.

From Table I we can see that the cooling energy consumption can be reduced by 69% through an increased use of free air cooling and shutting down fans which are not required. We can also see that networking energy consumption can be reduced by 38% by reducing the number of networking ports required for communication between the servers and by shutting down networking ports which are not required. This leads to an overall reduction of energy consumption of 21%. Finally, we can see that by reducing the overall energy consumption and directing all of the workload to the data centre with the lower carbon intensity we can reduce carbon emissions by 92%, illustrating the effectiveness of gUML.

V. CONCLUSION

In this paper, we identify and model various sources of power consumption in data centres in order to collect contextual information for each of these components. These sources are then mapped and contextualised in order to implement our extension for the UML deployment diagram called gUML. Our approach creates an advancement in tackling software energy efficiency in data centres at the design time by giving developers the tools to reason and design their application with

contextual energy information for data centres deployment and execution.

However, the scope of profiles discussed in this paper are limited to the power components of data centres. We aim to extend our approach to tackle additional profiles, such as modeling the power components in mobile devices. In addition, tracing cooling components to server points depends on thermodynamics, which can be difficult to compute. Finally, we plan to add software defined data centres features, such as security or monitoring, to our approach.

REFERENCES

[1] “Centre for energy-efficient telecommunications: Annual report 2013,” Report, 2014. [Online]. Available: http://www.ceet.unimelb.edu.au/publications/downloads/ceet-annualreport-2013.pdf

[2] Gartner, “Gartner estimates ict industry accounts for 2 percent of global co2 emissions,” 2007. [Online]. Available: http://www.gartner.com/newsroom/id/503867

[3] Herzog and Tim, “World greenhouse gas emissions in 2005 — world resources institute,” Report, 2009. [Online]. Available: http://www.wri.org/publication/world-greenhouse-gas-emissions-2005

[4] G. A. Plan, “An inefficient truth,” Report, 2007. [Online]. Available: https://docs.google.com/viewer?a=v&pid=sites&srcid=Z3JlZW5pY3Qub3Jn[illegible]

[5] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, “Renewable and cooling aware workload management for sustainable data centers,” in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40. ACM, 2012, Conference Proceedings, pp. 175–186.

[6] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastictree: Saving energy in data center networks,” in *NSDI*, vol. 10, 2010, Conference Proceedings, pp. 249–264.

[7] P. Mahadevan, S. Banerjee, and P. Sharma, “Energy proportionality of an enterprise network,” in *Proceedings of ACM GreenNet*, New Delhi, 30 August 2010, pp. 53–60.

[8] I. Manotas, L. Pollock, and J. Clause, “Seeds: A software engineer’s energy-optimization decision support framework,” in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 503–514. [Online]. Available: http://doi.acm.org/10.1145/2568225.2568297

[9] Y.-W. Kwon and E. Tilevich, “The impact of distributed programming abstractions on application energy consumption,” *Information and Software Technology*, vol. 55, no. 9, pp. 1602 – 1613, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584913000608

[10] A. Kumar, V. Jain, S. Kumar, and C. Chandra, “Green supplier selection: a new genetic/immune strategy with industrial application,” *Enterprise Information Systems*, vol. 10, no. 8, pp. 911–943, 2016.

[11] J. Bi, H. Yuan, M. Tie, and W. Tan, “Sla-based optimisation of virtualised resource for multi-tier web applications in cloud data centres,” *Enterprise Information Systems*, vol. 9, no. 7, pp. 743–767, 2015.

[12] I. Petri, H. Li, Y. Rezgui, Y. Chunfeng, B. Yuce, and B. Jayan, “A hpc based cloud model for real-time energy optimisation,” *Enterprise Information Systems*, vol. 10, no. 1, pp. 108–128, 2016.

[13] M. Cohen, H. S. Zhu, E. E. Senem, and Y. D. Liu, “Energy types,” in *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications*, ser. OOPSLA ’12. New York, NY, USA: ACM, 2012, pp. 831–850. [Online]. Available: http://doi.acm.org/10.1145/2384616.2384676

[14] J. Singh, K. Naik, and V. Mahinthan, “Impact of developer choices on energy consumption of software on servers,” *Procedia Computer Science*, vol. 62, pp. 385 – 394, 2015, proceedings of the 2015 International Conference on Soft Computing and Software Engineering (SCSE’15). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915025582

[15] Peffers, G. Tuunanen, H. Rossi, and B. Virtanen, “The design science research process: A model for producing and presenting information systems research,” 2006.

[16] L. A. Barroso and U. Holzle, “The case for energy-proportional computing,” *Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[17] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” *Advances in Computers*, vol. 82, no. 2, pp. 47–111, 2011.

[18] Y. C. Lee and A. Y. Zomaya, “Minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling,” in *Cluster Computing and the Grid, 2009. CCGRID’09. 9th IEEE/ACM International Symposium on*. IEEE, 2009, Conference Proceedings, pp. 92–99.

[19] L. Benini, A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 8, no. 3, pp. 299–316, 2000.

[20] C. Jeonghwan, S. Govindan, B. Urgaonkar, and S. Anand, “Profiling, prediction, and capping of power consumption in consolidated environments,” in *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, 2008, Conference Proceedings, pp. 1–10.

[21] R. American Society of Heating and A. C. Engineers, “Standards and guidelines,” 2014. [Online]. Available: https://www.ashrae.org/standards-research–technology/standards–guidelines

[22] M. Wang, N. Kandasamy, A. Guez, and M. Kam, “Adaptive performance control of computing systems via distributed cooperative control: Application to power management in computing clusters,” in *Autonomic Computing, 2006. ICAC’06. IEEE International Conference on*. IEEE, 2006, Conference Proceedings, pp. 165–174.

[23] Mikko Pervilä and Jussi Kangasharju, “Cold air containment,” in *Proceedings of ACM SIGCOMM Workshop on Green Networking*, Toronto, 19 August 2011, pp. 7–12.

[24] L. A. Barroso and U. Hölzle, “The datacenter as a computer: An introduction to the design of warehouse-scale machines,” *Synthesis Lectures on Computer Architecture*, 2009.

[25] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, “Powerpack: Energy profiling and analysis of high-performance systems and applications,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 5, pp. 658–671, 2010.

[26] C. Gruter, P. Gysel, M. Krebs, and C. Meier, “Eod designer: A computation tool for energy optimization of data centers,” in *Green and Sustainable Software (GREENS), 2012 First International Workshop on*. IEEE, 2012, Conference Proceedings, pp. 28–34.

[27] K. Dzmitry, B. Pascal, and K. S. Ullah, “Dens: data center energy-efficient network-aware scheduling,” 2011. [Online]. Available: http://disi.unitn.it/ klezovic/papers/DENS-cluster.pdf

[28] K. Patel, M. Annavaram, and M. Pedram, “Nfra: Generalized network flow based resource allocation for hosting centers,” 2012.

[29] D. Shorin and A. Zimmermann, “Evaluation of embedded system energy usage with extended uml models,” in *2nd Workshop EASED@ BUIS 2013*, 2013, Conference Proceedings, p. 21.

[30] A. Verma, P. Ahuja, and A. Neogi, “Power-aware dynamic placement of hpc applications,” in *Proceedings of the 22Nd Annual International Conference on Supercomputing*, ser. ICS ’08. New York, NY, USA: ACM, 2008, pp. 175–184. [Online]. Available: http://doi.acm.org/10.1145/1375527.1375555

[31] “Carbon Monitoring for Action,” http://carma.org/.

[32] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in *ACM SIGCOMM Computer Communication Review*, vol. 38. ACM, 2008, Conference Proceedings, pp. 63–74.

[33] V. Kontorinis, L. E. Zhang, B. Aksanli, J. Sampson, H. Homayoun, E. Pettis, D. M. Tullsen, and T. Simunic Rosing, “Managing distributed ups energy for effective power capping in data centers,” in *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*. IEEE, 2012, Conference Proceedings, pp. 488–499.

[34] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya *et al.*, “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” *Advances in computers*, vol. 82, no. 2, pp. 47–111, 2011.