Towards Model Driven Verifiable Deployment of Distributed Simulations in Cloud

Yogesh D. Barve
Institute for Software Integrated Systems,
Dept. of EECS, Vanderbilt University,
Nashville, TN 37212, USA
Email:{yogesh.d.barve}@vanderbilt.edu

***Abstract*—Running simulations on cloud computing platforms offers advantages to users such as on-demand computing and scalability. Despite these benefits offered by cloud computing platforms, limitations exists in verifiable and efficient deployment of these simulations in the cloud. Moreover, distributed simulations impose additional requirements of coordinated time stepped execution to progress the simulations. As such, deadline-aware resource allocation for different simulation entities and dynamic execution of load balancing strategies are required to minimize the total execution time. Also, the cloud provider's interest in minimal execution cost is another requirement which demands workload consolidation for these distributed simulations. In this context, there is a general lack of mechanisms that address these concerns in the cloud hosted distributed simulation space. To address these gaps, this research proposes Model Driven Verifiable Distributed Simulations in Cloud (MoViDiX). It provides DSML building blocks for rapid provisioning of distributed simulations in cloud. It also provides a verification subsystem that identifies simulation resource incompatibilities and finds solution for verified deployment of the simulation. Leveraging Models@Runtime principles, dynamic resource management for effective simulations for deadline aware execution is proposed.**

***Index Terms*—co-simulation, distributed simulations,verification, Cloud.**

I. INTRODUCTION

With the new focus on technologies like internet of things, fog computing, and blockchain enabled services, understanding the fundamental technologies of distributed systems like fault tolerance, publish-subscribe communication, client-server, and peer-to-peer technologies becomes critical. Simulations provide a rapid way to run and test new distributed systems algorithms in a controlled environment. But to simulate highly scalable distributed systems such as those that manifest in the scenarios like smart city, smart grid systems, one needs a large amount of computing resources. Running such large scale simulations can benefit from the scalability, on-demand, ubiquitous nature of cloud computing [1], [2].

Although these benefits are very compelling for running simulations in the cloud, the research community is faced with numerous challenges from moving the execution of simulations from grid like environment to the cloud computing model. Deployment of simulations to the cloud requires an in-depth understanding of various cloud-specific deployment tools and deployment engines. Lack of understanding of these tools can result in sub-optimal deployment of these simulations. Moreover, resource requirements for different participating entities in a distributed simulation can vary based on the workload characteristics of the individual simulation agent. As such, tools and strategies are required to provide seamless execution of the simulations improving performance and minimizing the makespan of execution of the simulation run. Migration of the simulations from the grid to the cloud environment is another challenge which the users face. Issues such as vendor lock-in can make the users tied to a given cloud provider, thus making it difficult for the user to move to another cloud provider in future.

II. PROBLEM STATEMENT

In this research we are exploring the use of cloud computing resources for running distributed simulations. Simulations can either consist of single participating entity (monolithic), or a group of coordinated simulation entities as in distributed simulations. In this doctoral research we address challenges imposed in running simulations in cloud along three dimensions. These are listed in the following section below:

- **Simulation-imposed challenges**
  - *CH1: Distributed and Synchronized*: Distributed simulations comprise a group of simulators which coordinates and advances time synchronously. The execution semantics follow time stepped execution. The distributed nature of the execution demands strict communication latency deadlines among participating entities. Also, time stepped execution demands state synchronization between participating entities before progressing into the next execution time step.
  - *CH2: QoS and Resource Integration challenges*: Different participating simulators in a distributed simulation settings may have different levels of resource requirements as execution progresses. Some simulators may need availability to specific hardware components such as GPU, cpu core, memory, etc. In a Hardware-in-the-Loop(HiL) simulation scenario, for instance, there may be a need to place certain simulation components like trigger unit close to the hardware while the actual computation intensive decision making can happen in the cloud environment. The simulation deployment and execution needs must some capture these requirements from the

user so as to make sound decisions for optimal and successful execution of the distributed simulation.
  - *CH3: Straggler Mitigation*: In time stepped distributed simulations, all participating entities wait on each other before progressing to the next time step for execution. This is the BSP model of computation [3]. In the BSP computation model, if some of the participants have a slower execution speed compared to the rest of the participants, the progress of the simulation as a whole is dictated by the slowest progressing participant. As such, in a distributed simulation, we need to capture this execution behavior either apriori or during runtime so as to mitigate the slow down caused due to such straggler participants.
- **Cloud-imposed challenges**
  - *CH4: Multi-tenant Cloud Environment*: In a cloud computing environment, resources are shared among various applications hosted by different providers. Thus, issues such as application interference [4] can affect the simulation performance and execution progress. Host system overloading can also affect the execution makespan of the simulations.
  - *CH5: Fault Tolerance Support*: The cloud computing environment is prone to regular system outages due to software and hardware failures. As such when running large-scale long running simulations, it becomes critical to have a fault tolerance strategy to mitigate loss of computation and restart execution time.
- **Usability Challenges**
  - *CH6: Ease of Use* : There is general lack of tools to support deployment of simulations in the cloud with ease. A user should be able to configure the experimental setup, provide resource specification, execution and fault tolerance policies using a graphical web portal. Also to enable collaboration, the web application should be able to provide Git style versioning of simulation models.
  - *CH7: Distributed Systems Learning Toolkit*: A repository of commonly used distributed systems algorithms to quickly test and deploy simulations in the cloud. This enables easier learning tools to understand fundamentals of distributed systems in this era of ubiquitous computing.

III. RELATED WORK

In this section we describe related works along the lines of the challenges that have been highlighted in the previous section.

- **Simulation and cloud imposed challenges**: DEXSim [5] presents a framework for replicated execution of simulations utilizing the available hardware resources for speeding up executions of different scenarios in the simulation experiment. Although this is similar to the distributed simulation, the experiments are not geared towards running in a multi-tenant cloud where applications are susceptible to interference across tenants.

  The INDICES [6] framework also addresses the application interference issues in the cloud and finding an appropriate hosting platform that can provide guarantees to the QoS latency requirements of the application. But the application use cases are monolithic and does not address distributed simulation-specific requirements such as coordination, and distributed execution.
- **Usability challenges** Previous work has shown MDE and DSML being effective tools in providing intuitive abstractions for constructing simulation experiments [7]. Our work builds on top of [7], and will provide a visual DSML to specify simulation resource requirements and verification module for checking correct deployment of the simulation execution.

IV. PROPOSED SOLUTION

To address the challenges presented in Section II, we propose a Model Driven Verifiable Distributed Simulations in Cloud(**MoViDiX**). An overview of the framework is shown in the Figure1. In this research we propose to build a platform

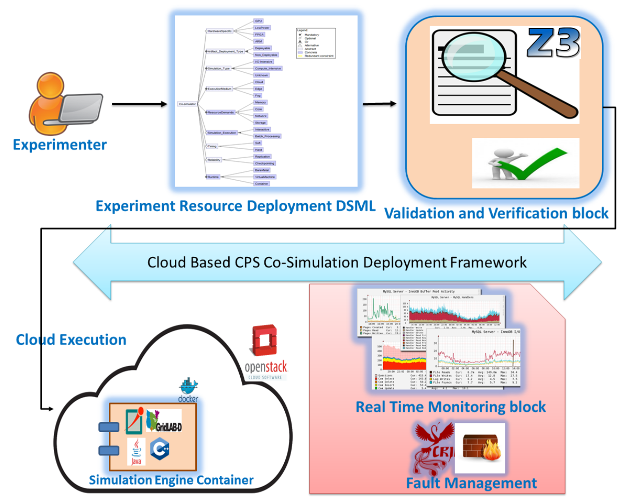

Fig. 1: Architecture Overview of MoViDiX

that enables users to better understand distributed systems concepts and build simulation models to test and run in the cloud. The platform leverages domain specific concepts from distributed systems such as pub/sub, client server, peer-to-peer, etc, to construct simulation (**CH 6,7**). The MDE capabilities like modeling and code generation as put into practice for rapidly taking initial ideas to building a simulation of distributed system in cloud environment with ease. We will also work on a DSML which will allow users to provide resource and runtime specification(**CH 1,2**). A verification module is also being developed that can check the simulation constraints and notify of any violations generated.

We plan on utilizing the Z3 SMT solver to provide verification and solution satisfying simulation deployment constraints. We shall leverage Models@Runtime approach for dynamic resource management of individual simulation entities to provide runtime optimal performance and meeting simulation QoS requirements(**CH 3,4**). We plan on using Docker technology that provides encapsulation for simulation execution(**CH 4, 6**). The required files which contain the specification of the software dependency is auto-generated using MDE principles. We are also leveraging web-based generic modeling environment WebGME to perform our distributed simulation modeling and deployment(**CH 3,4**).

V. Plan for Evaluation and Validation

To evaluate our system, we will be conducting experiments in our university's cloud datacenter as well as leveraging the NSF Chameleon cloud platform. Using the MDE we should be able to select what cloud provider we would like to deploy our experiments. Also user studies are being carried out to test the usability and the educational learning aspects of the tool.

VI. Expected contributions

This research will make the following contributions:

- MDE framework for design and deployment of distributed simulations in cloud.
- Dynamic resource management strategies for effecient execution of distributed simulations in cloud by leveraging Models@Runtime principles.
- DSML for experiment and resource specification for distributed simulations in cloud.
- Verifiable deployment of distributed simulations.

VII. Current Status

Currently we have addressed the usability issue for designing simulation experiments and running them in the cloud environment(**CH 4, 6, 7**) in [8], [9]. We are currently working on designing the DSML to address **CH 2**. In the future we plan on addressing resource management issues using the Models@runtime principles for providing effective resource allocation by using the monitoring, decision and action methodology. This research is being conducted under the supervision of Dr. Aniruddha Gokhale, Associate Professor, Vanderbilt University.

Acknowledgments

This work is supported in part by NIST contract number 70NANB15H312, NSF CPS VO contract number CNS-1521617 and NSF US Ignite CNS 1531079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] S. Shekhar, H. Abdel-Aziz, M. Walker, F. Caglar, A. Gokhale, and X. Koutsoukos, "A simulation as a service cloud middleware," *Annals of Telecommunications*, vol. 71, no. 3-4, pp. 93–108, 2016.

[2] F. Caglar, S. Shekhar, A. Gokhale, S. Basu, T. Rafi, J. Kinnebrew, and G. Biswas, "Cloud-hosted simulation-as-a-service for high school stem education," *Simulation Modelling Practice and Theory*, vol. 58, pp. 255–273, 2015.

[3] L. G. Valiant, "A bridging model for parallel computation," *Communications of the ACM*, vol. 33, no. 8, pp. 103–111, 1990.

[4] F. Caglar, S. Shekhar, and A. Gokhale, "A performance interferenceaware virtual machine placement strategy for supporting soft realtime applications in the cloud."

[5] C. Choi, K.-M. Seo, and T. G. Kim, "Dexsim: an experimental environment for distributed execution of replicated simulators using a concept of single simulation multiple scenarios," *Simulation*, vol. 90, no. 4, pp. 355–376, 2014.

[6] S. Shekhar, A. Chhokra, A. Bhattacharjee, G. Aupy, and A. Gokhale, "Indices: Exploiting edge resources for performance-aware cloud-hosted services."

[7] H. Neema, J. Sztipanovits, M. Burns, and E. Griffor, "C2wt-te: A model-based open platform for integrated simulations of transactive smart grids," in *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2016 Workshop on*. IEEE, 2016, pp. 1–6.

[8] Y. D. Barve, P. Patil, and A. Gokhale, "A cloud-based immersive learning environment for distributed systems algorithms," in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 754–763.

[9] Y. Barve, P. Patil, A. Bhattacharjee, and A. Gokhale, "Pads: Design and implementation of a cloud-based, immersive learning environment for distributed systems algorithms," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, 2017.