

COOPERATION OF SIMULATION AND DATA MODEL FOR PERFORMANCE ANALYSIS OF COMPLEX SYSTEMS

Kim, B. S. & Kim, T. G.

School of Electrical Engineering, KAIST, 34141, Daejeon, Republic of Korea

E-Mail: kevinzzang@kaist.ac.kr, tkim@kaist.ac.kr

Abstract

Modelling and simulation (M&S) is one of the fundamental methods of performance analysis. In other words, how well a modeller builds a model is a key point of a successful performance analysis. Before such a performance analysis, a model for prediction should be constructed. There are two types of models: data model and simulation model. Data model represents correlational relationships between one set of data and another. Conversely, simulation model represents causal relationships between a set of controlled inputs and corresponding outputs. This paper identifies the characteristics of each modelling method and presents a cooperative model development process for performance analysis of complex systems. The cooperative method contains conceptual modelling, model classification, and model integration/implementation. The model classification method effectively reflects and maximizes the features compared earlier. Then, they are modelled respectively and integrated. This paper also applies the proposed modelling to develop a model of Hadoop using artificial neural network (ANN) and discrete event systems specification (DEVS). To demonstrate the validity of the case study, it presents experiments to show the possibility of a proposed approach.

(Received in July 2019, accepted in October 2019. This paper was with the authors 2 weeks for 1 revision.)

Key Words: Cooperative Model Development, Data Modelling, Simulation Modelling, Artificial Neural Network, Discrete Event Systems Specification (DEVS), Hadoop

1. INTRODUCTION

Performance analysis of a complex system is an important issue for system engineers because it can be utilized to forecast the behaviour of the system. This analysis can be used for maximizing the performance, resource planning of the system, or parameter tuning [1]. Performance analysis can be a tricky task because it requires a deep understanding of the system. The more complex the system, the more cost and effort are required for the analysis.

The performance analysis generally consists of the system, evaluation techniques, metrics, and workloads [2]. Evaluation techniques include measurement, simulation and analytical modelling, and performance metrics, including throughput, response time, utilization, and so on. For successful performance analysis, a careful selection of analysis techniques is very important. M&S is one of the fundamental methods of performance analysis. In other words, how well a modeller builds a model with a purpose in mind is a key point for a successful performance analysis.

However, it is difficult to develop a model for performance analysis for several reasons. For example, a highly dynamic environment and heterogeneity of the system can make modelling difficult. Complexity of the system as well as the workload are the factors that make modelling difficult [3]. The operations of a real-world system or physical system can be modelled in two ways. One way is to use prior knowledge of the system, and the other is to use empirical data of the system. As the complexity of the system increases and as more data becomes available in the big data era, we need more efficient modelling strategies using both modelling methods. Big data has become widespread as a means of performance analysis for a diversified world [4]. Data modelling using such big data has focused on representing correlations of empirical data. It is based on analysing the data of a system, in particular finding connections between the system variables without explicit knowledge of the

behaviour [5] of the system as depicted in the left of Fig. 1. It has seen wide use to predict the future behaviour of the complex system [6-8]. Although it seems possible to make robust forecasts through the data model with sufficient information, contrary to this expectation, it is not always an accurate modelling method. One of the obvious limitations is that it is just able to express correlations between parameters, so it cannot handle anomalies and changes of the system [5]. Also, when analysing the performance of the system, it is necessary to change algorithms and models as well as parameters according to the purpose of the analysis. In other words, for flexible performance analysis, the model needs to have various types of inputs such as parameters, algorithms, and models as attributes. However, it is not possible with a data model after modelling is complete.

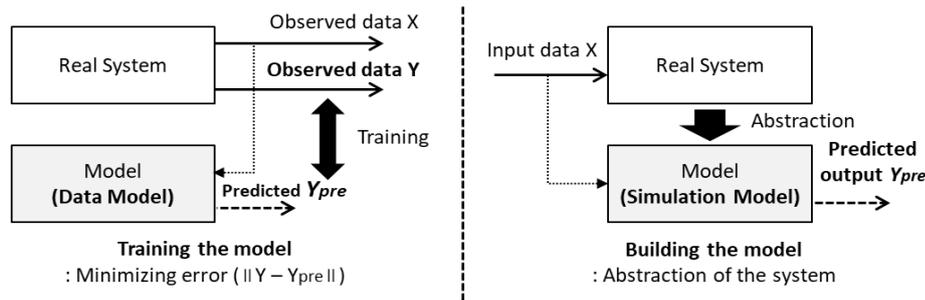


Figure 1: Definition of data modelling and simulation modelling.

On the other hand, simulation modelling is a theory-based approach based on abstraction of the system using prior knowledge, for example, physical or operational laws/theories as depicted in the right of Fig. 1. It is generally used in the research field of simulation. It is possible to clearly express the causal relationship between a set of controlled inputs and corresponding outputs with system knowledge, which means a statement of what causes what and why [9]. Nevertheless, when it is hard to acquire sufficient system knowledge, it is hard to construct a model completely that satisfies modelling objectives. Table I shows the characteristics of data/simulation modelling in various perspectives on the basis of the features mentioned previously.

Table I: Comparison between two modelling approaches.

Type	Data Modelling	Simulation Modelling
Model Representation	Associational relation with system data	Cause-effect relation with knowledge
Analysis Level	Predictive	Predictive, prescriptive
Input Types	Parameter	Parameter, algorithm, model
Valid Predict Condition	Structure unchanged after modelling	Model validation
Non-existing System	Not applicable	Applicable
Simulation Time	Very short	Relatively long
Fidelity	Low	Relatively high

Complex systems are likely to encounter these differences and limitations, because there are many components and attributes to consider in the system. This situation can lead to insufficient model expressiveness. Therefore, it is essential to use both models simultaneously and appropriately for more advanced modelling. In summary, there is a limit to completely model a complex system using data/simulation modelling, respectively. To develop a model efficiently through mitigating the disadvantages, a new cooperative modelling approach is needed. Accordingly, this paper proposes a systematic modelling process that merges the two methods. With the proposed cooperative approach, the model classification method in the process allows us to develop the model flexibly and efficiently for the performance analysis. It also provides a model integration and implementation for the execution of the developed

model in the same simulation environment. Then, this paper applies it to the Hadoop framework to demonstrate how to use the approach. We expect that the proposed method provides a guideline of modelling for performance analysis in the big data era.

The remainder of the paper is organised as follows. Section 2 introduces previous research similar to our work. Then, Section 3 provides a cooperative model development method for performance analysis. The proposed method is applied to the Hadoop framework in Section 4. Finally, Section 5 concludes the discussion.

2. RELATED WORKS

Many studies have been studied on cooperation of data/simulation modelling for performance analysis in multiple areas. These studies can be classified into two groups. The first one is embedded, which is similar to system identification. There are some studies on this type of cooperation [10-12]. In these studies, researchers create a hypothetical structure model of the system through simulation modelling, and identify the validated parameters required for the structure through machine learning using acquired data. A complete model is built through the fusion of two modelling methods. However, instead of having two separate models, the two methods are used together to establish one functional model. In other words, data modelling alone is difficult to express one complete function in these studies.

The second case is domain-specific collaboration. In this case, collaboration is derived in a specific manner depending on the characteristics of each domain. There is a research that converges machine learning into agent-based modelling [13]. Machine learning is just used for refinement of action in agent-based modelling. There is also a study that improves validity of discrete event system model for decision making [14]. These studies provide better analysis of a system through collaboration of two models; however, data model is just used to represent only one-dimensional internal behaviours. In the Hadoop domain, which we target in the case study of this paper, several studies have been also conducted [15]. HSim [16] and MRPerf [17] are representative simulators. They can simulate the dynamic behaviours of clusters. Additionally, they can configure many system parameters, including hardware and cluster parameters. However, they just consider the data modelling implicitly and simply. They involved simulation modelling for the evaluation of the system and contained an abstracted data model. For example, the characteristics of each Hadoop application and disk I/O model that are difficult to model with low-level knowledge are simply reflected in these simulators. In other words, there are no independent data models to represent the subsystem in the model. In addition, because it used the existing developed model as the simulation model, it cannot be seen as a perspective of modelling method.

From all these considerations, the common problem of previous research studies has been that there was no methodology or guideline for integrating the two methods explicitly. From a modeller's point of view, there is a need to classify and integrate the features of each model in order to model more efficiently. It is also important to enhance model expressiveness and model flexibility for deriving more reliable simulation results. In this respect, we propose a cooperative modelling method, which allows us to develop an advanced model for better analysis of complex system through appropriate cooperation.

3. COOPERATIVE MODEL DEVELOPMENT PROCESS

In the previous section, we confirmed that complex systems cannot be fully modelled through a single modelling approach, respectively. Therefore, we present a cooperative modelling process that can take each of the advantages and explain how the mutual cooperation takes place in each step. The proposed process follows the general M&S process. Its cycle has distinct activities in the model, such as requirements analysis, conceptual model design,

detailed model design, implementation, verification and validation, experimental design, and simulation [18]. The proposed cooperative modelling is based on this hardware/software co-design [19]. The method is a modelling procedure for developing simulation systems by cooperating with each other (Fig. 2).

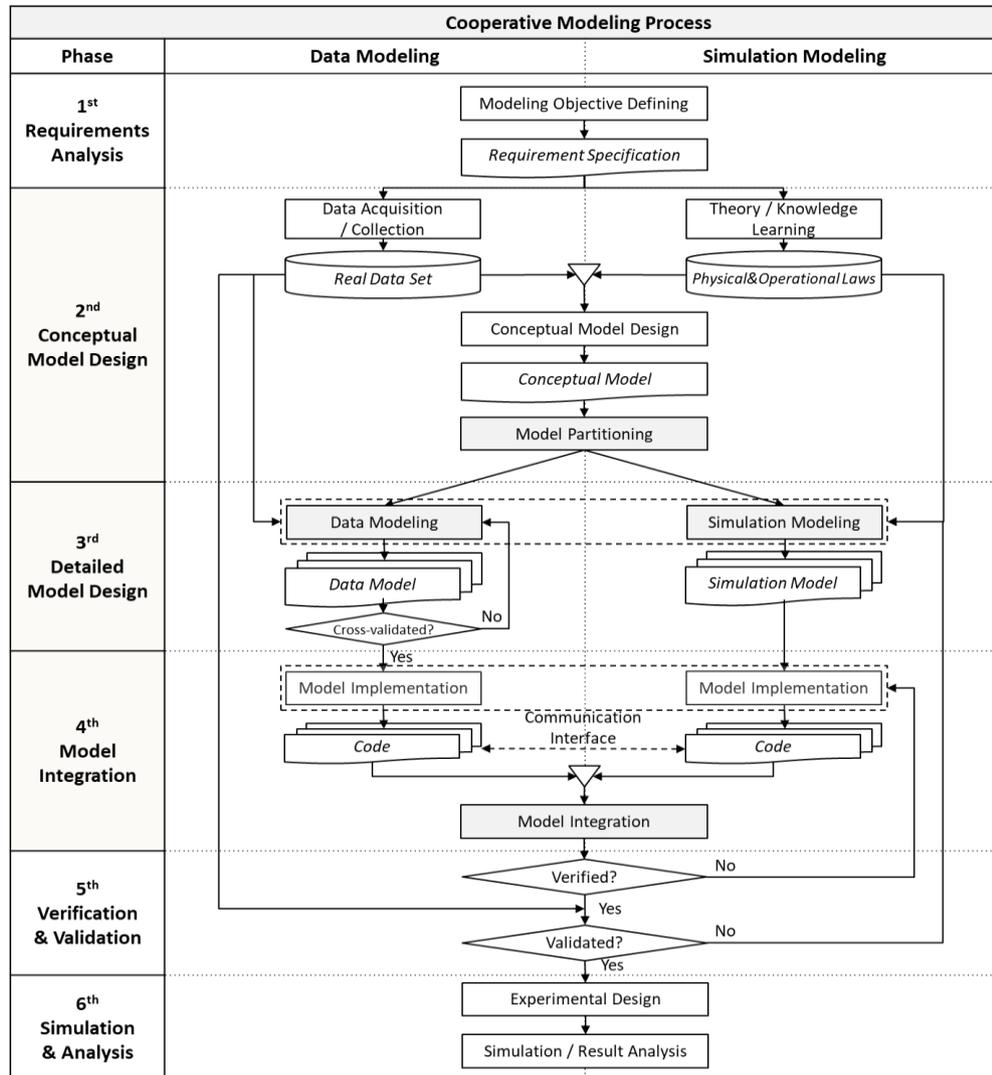


Figure 2: Proposed model development process.

There are some assumptions and constraints for this proposed process. The first is that there is no general-purpose model. The model must be developed with a purpose in mind. In other words, no model of a system will present all features of the system. The next is that a modeller should have an affinity for both data and knowledge. Depending on the purpose, the knowledge of the system and the data of the system should be acquired. The final one is that each model should be implemented in the same environment. A simulation environment that can understand the semantics of both the data/simulation modelling and how actions in one approach affect the state of the other is required for the proposed method.

3.1 Conceptual model design

A conceptual model is a statement of the contents and the internal representations that are the user's combined concept of the model. It contains algorithms and logics, and it represents assumptions and limitations explicitly [20]. In this process, it is also important to define the models clearly as functional units before model classification as shown in Fig. 3. A modeller

should consider some criteria for successful model classification [19]. Sometimes a model is desired that enables the function or algorithm to be easily changed, so modifiability should be considered. Data from one system cannot always be representative of the system. As we mentioned in Table I, we need another data set under the changed condition of the system. It is hard to evaluate changes in various parameters, models, and workload characteristics with data models. It is also difficult to use various types of input only with data models. In other words, the system has high modifiability when structural changes or various input types are required. On the other hand, low modifiability is defined as when no structural change is required or when only numerical inputs are used as input parameters.

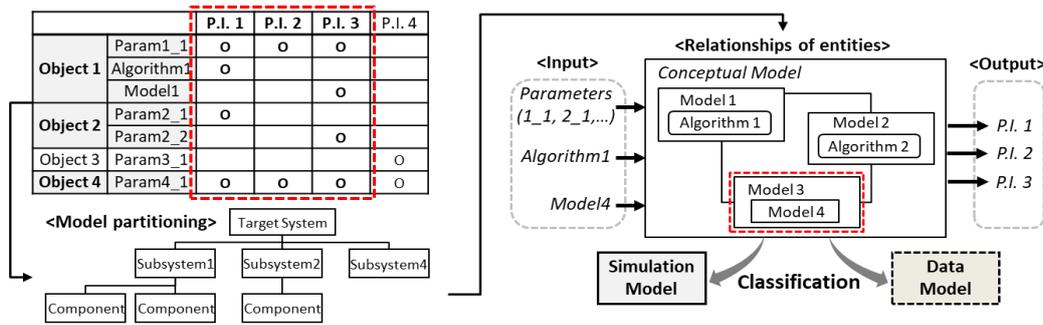


Figure 3: Conceptual model design.

Data/knowledge acquisition level is one of the most important criteria. Extensive physical/operational knowledge that includes idealistic assumptions and constraints is required for simulation modelling. For data modelling, sufficient data should be collected that cover the operating range of the system for the high reliability model. If it is difficult to obtain actual data, it can be modelled through the simulation data acquired from the existing simulator. If the data covers the operating range of the system, then the data model will be able to make a more accurate prediction than the simulation model involving idealistic assumptions and constraints.

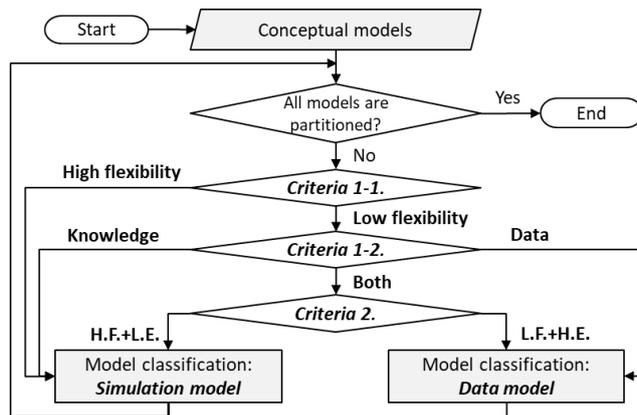


Figure 4: Model classification method with characteristics of two models.

Performance requirement is one of the factors to consider. Fidelity and efficiency requirements would have been considered previously for metrics that quantitatively evaluate system behaviour. For example, simulation models are available when high fidelity (H.F.) and relatively low efficiency (L.E.). Conversely, data models are available when relatively low fidelity (L.F.) and high efficiency (H.E.). After these three criteria are considered, the models should be classified through the flowchart created by applying them. Fig. 4 represents the flowchart of the model classification method. Through the flowchart, we could divide the conceptual models into two models.

3.2 Detailed model design

The models identified and classified in the conceptual modelling step should be built using each method separately. In this paper, we use discrete event system specification (DEVS) formalism [21] as one method of simulation modelling. It is a set theoretic specification of discrete event systems, which has been generally used for modelling many fields of science and engineering [22]. The DEVS formalism is modular, hierarchical, and object-oriented, so it is proper for the modelling of the dynamic system. The formalism consists of an atomic DEVS model representing the system behaviour and a coupled DEVS model representing the structure of the system. The entire system consists of a set of atomic or coupled component models, thus representing a hierarchical structure. In the simulation modelling, state transitions by I/O events are described using the DEVS formalism, and detailed operations at specific states are described using algorithms and mathematical equations.

Next, various techniques can be used for data modelling, including data mining and machine learning. For example, machine learning can be a means of data modelling. With a machine learning method, such as ANN, a user can map associations between one set of data and another. After passing through a validation process of the mapping with data in the real world, using general performance indices such as root mean square error (*RMSE*), we can acquire a data model. Then, it enables the prediction of a future value of a data set using a given data set. In this paper, we use ANN as a method of data modelling [23].

3.3 Model integration and implementation

After detailed model design, it is necessary to integrate both simulation and data model. For the integration of the models, we use the two integration types (logical and physical boundaries), as shown in Fig. 5 [19]. The simulation model and data model can be physically (horizontal) separated. And the simulation model is considered to be executing the data model in the logical boundary (vertical). This simulation model may contain one or more data models. Generally, the data model can be used to implement some of the functionality of the simulation model. In the cooperative modelling, models can be integrated with both types of integration. After integration of all the models, they can be implemented in the heterogeneous environments and then interoperated using a middleware. Alternatively, they can be developed with a homogeneous environment. After they are implemented, simulation is conducted through the remaining model development process, such as verification/validation, experimental design, and execution. The next section shows the application of the proposed process through a practical example.

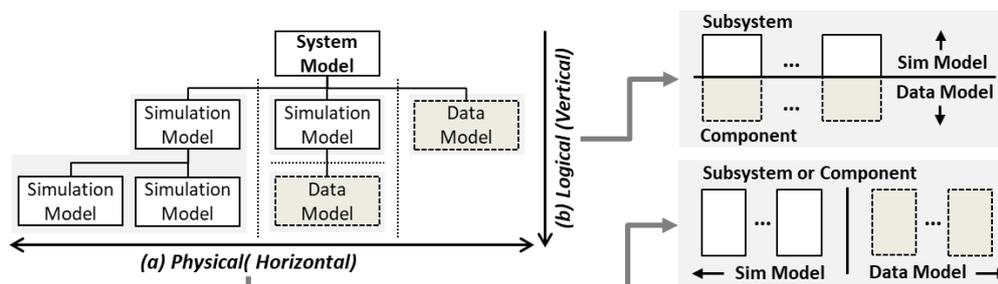


Figure 5: Two types of model integration.

4. APPLICATION: HADOOP

This chapter provides a case study that applies the proposed method to Hadoop [24]. It is an open-source framework used for distributed storage and processing of big data. Because it is difficult to set up a real Hadoop environment to evaluate the scalability of application up to a

thousand nodes, M&S is necessary for the performance analysis. It represents the highly dynamic environment of a computer system and includes various levels of computer layers, and it is easy to collect actual data for data modelling of the system. The core of Hadoop consists of a processing part, which is a MapReduce, and a storage part, known as a Hadoop Distributed File System (HDFS). MapReduce is a computing framework for large-scale distributed data processing based on the divide and conquer paradigm. It works by breaking the processing into map and reduce functions [25]. HDFS is a distributed file system of Hadoop that stores data reliably using commodity clusters [26]. This section represents the sequential application of Hadoop to the proposed model development method.

4.1 Conceptual model design

To model the Hadoop system, one must build a conceptual model that expresses the structure, abstraction level, and system elements. It should be classified into two models according to the objective of the analysis and the acquisition level of data/knowledge. In the Hadoop, there are four types of models: MapReduce, HDFS, platform, and application model. Table II and Fig. 6 show the classification result of these models through the classification method described in the previous chapter. These classified models can be modelled using each modelling approach.

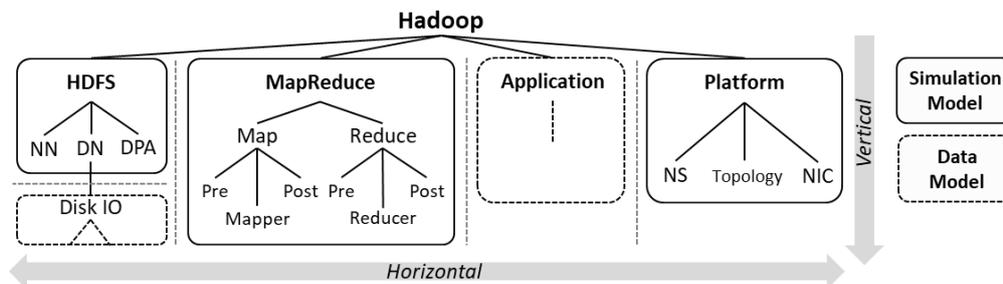


Figure 6: Classified model of Hadoop

Table II: Model classification of Hadoop.

Component	Subcomponent	Criteria 1	Criteria 2	Criteria 3	Result
MapRudce	Job/Task Tracker	High	-	-	Sim Model
	Map/Reduce	Low	Non-existing	-	Sim Model
HDFS	Name/Data Node	High	-	-	Sim Model
	Disk I/O	Low	Both	L.F.+H.E.	Data Model
Platform	Cluster	Low	Both	H.F.+L.E.	Sim Model
	Network	High	-	-	Sim Model
Application	-	Low	Black-box	-	Data Model

4.2 Detailed model design

Through the model classification, we can see that the application and the disk I/O are built by data modelling. The application describes the Hadoop programs such as WordCount and TeraSort. The disk I/O describes a storage model for file write, read, and shuffle. It is necessary to understand their internal operation mechanisms; however, they are very complex and require a deep level of knowledge. We can use existing simulators such as DiskSim [27] for the disk I/O model. However, this can overload the time and resource, which does not fit the simulation objective. To do data modelling using ANN, firstly, the environmental data should also be collected or extracted from executions of the Hadoop to use as training data. Then, each model can be built through a learning process using the data set. We use the Lavenberg-Marquardt optimization technique as a learning algorithm [28] and *RMSE* as a

measurement of learning performance. Detailed data modelling procedure, acquired data list and input/output are shown in Fig. 7.

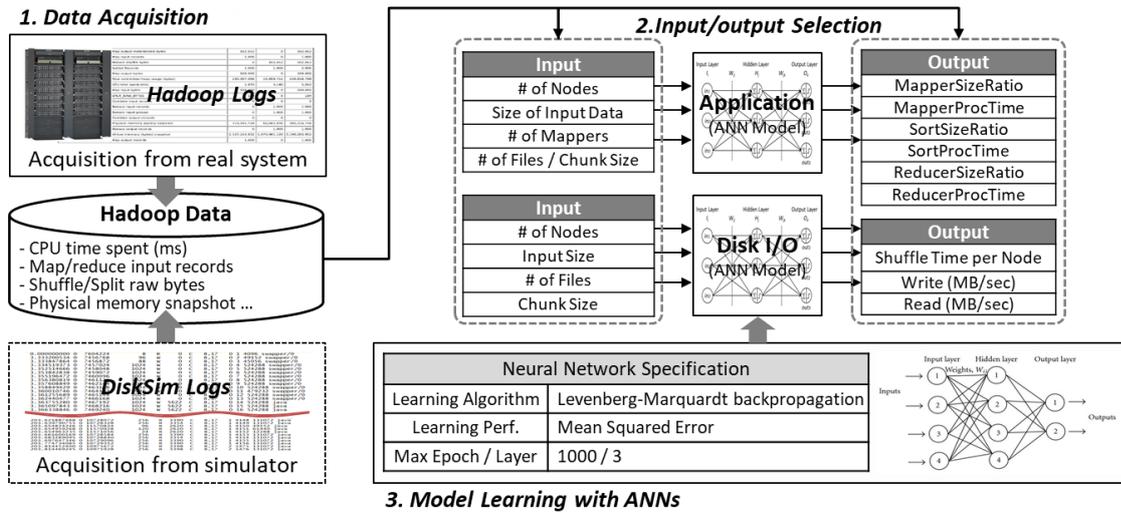


Figure 7: Detailed data modelling procedure.

On the other hand, the others can be modelled using simulation modelling. The MapReduce operations are performed through the map, shuffle, and reduce processes, and they run independently in parallel. The HDFS includes a structure of distributed file system and operation of name node/data node. The platform model includes a structure of cluster and hardware models. The network model describes a topology model, thus indicating connections between the clusters [29]. When these models are modelled by data modelling, details can be highly abstracted. This makes it hard to perform behaviour analysis and structural change.

Also, because it is difficult to represent heterogeneous computing environments of the Hadoop platform with numerous nodes using data modelling, simulation modelling through understanding the whole process is needed. In this paper, we use DEVS formalism to perform the simulation modelling described in Section 3.2.

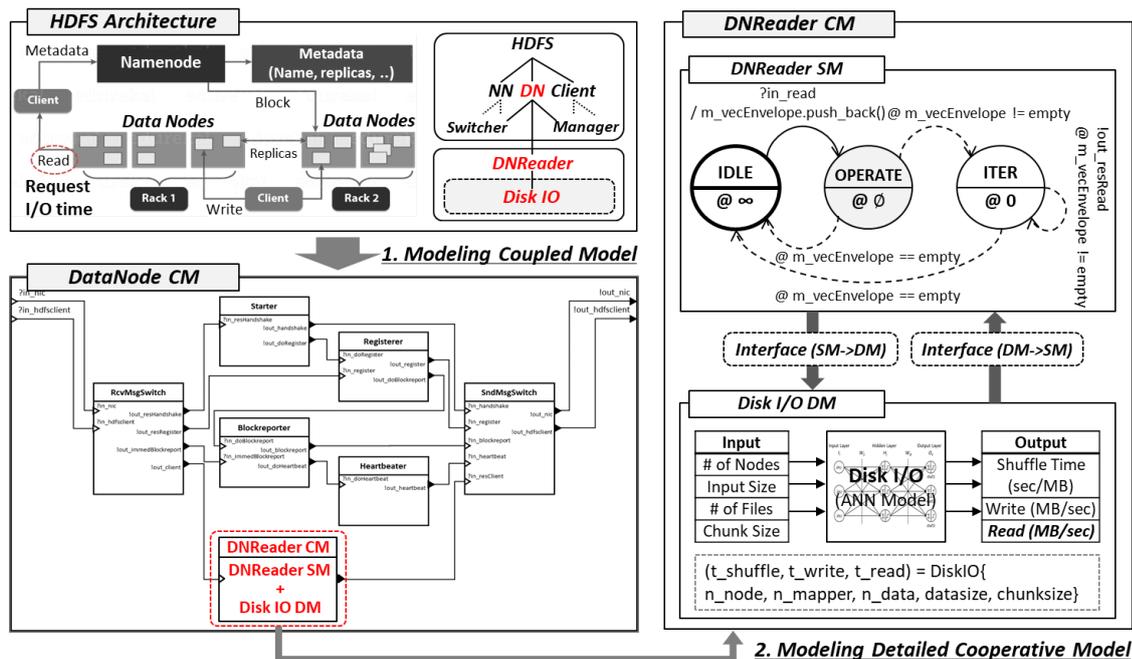


Figure 8: Example of simulation model and data model: HDFS.

4.3 Model integration and implementation

After detailed modelling, it is required to integrate them. Each model can be connected through predefined input/output relationships. They can be implemented each other in the heterogeneous environments and then interoperated using a middleware. Or they can be developed and integrated with the homogeneous environment. In this dissertation, we develop the Hadoop model in the same environment for DEVS formalism called DEVSim++ [30]. Fig. 8 shows the example of details of integration using HDFS model, and Fig. 9 shows the integrated model and the connections among the models.

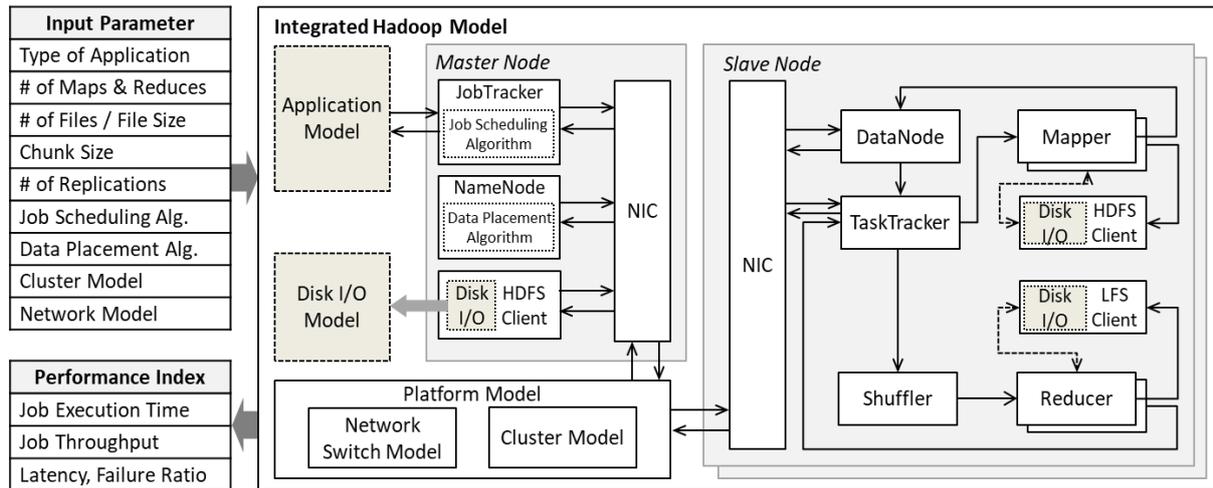


Figure 9: Integrated Hadoop model.

4.4 Experiment

In this subsection, two experiments are designed to demonstrate the effectiveness of the proposed work [31]. The experiments are to show the prediction accuracy and simulation efficiency respectively using the Hadoop model. Because there is no method that can be compared directly, we simply compare the results using models made with the modelling methods commonly used in previous studies. For the experiments, two models are used as a control group. The first model is created using only simulation modelling. We use the DEVS formalism in the same way as the proposed work for HDFS, MapReduce, platform, and network models. The application is made into a simple simulation model through the abstraction, and the disk I/O uses the existing DiskSim made by the domain expert. On the other hand, the data model is simpler than the simulation model. It is modelled at once using the entire input and output data of Hadoop, without distinction among the component models.

Meanwhile, prediction accuracy can be compared through the *RMSE*. It can be calculated using the difference between the real execution and the simulation result. The smaller the value, the closer the predicted result is to the actual result. We compare the prediction accuracy of the proposed model with that of the control groups. We use job completion time and throughput according to the number of data nodes. Also, we can compare the simulation efficiency using ratio of prediction accuracy to simulation execution time. It can be simply compared through the inverse efficiency score (*ISE*) that deals with speed-accuracy trade-offs [32]. The smaller the value, the better the simulation efficiency.

Fig. 10 shows the detailed parameters used in the experiments and the two experimental results. Firstly, it shows that the proposed model has a lower *RMSE* value than the other two groups (Table III). In other words, it means that the proposed model has enhanced the prediction accuracy compared to the other groups.

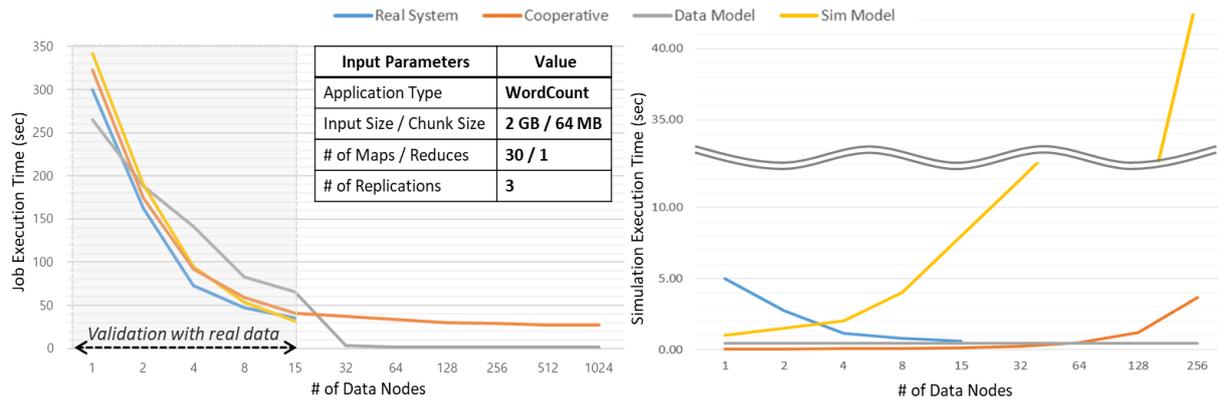


Figure 10: Experimental parameters and results: prediction accuracy (left) and efficiency (right).

We can see that the accuracy of the data model is lowest. This is because it is difficult to express the internal conditions accurately with only the input/output data used for data modelling. The simulation model shows relatively accurate predictions, which can vary depending on how the application affects the overall process of the system. Because the application model uses an abstracted one, the less impact the application has on the overall system. However, the DiskSim can cause the low speed of the simulation model.

Table III: Comparison of experimental results.

Evaluation type	Evaluation index	Model type		
		Data model	Sim model	Proposed model
Prediction accuracy	<i>RMSE</i> (job execution time)	42.1	24.6	18.7
Simulation efficiency	<i>IES</i> (real simulation time)	0.66	15.02	0.78

Fig. 10 and Table III also show the simulation efficiency using real simulation execution time. In the simulation, the run-time of the data nodes increases as the number of nodes increases when the data size is constant. This is because, as the number of nodes increases, the computing resources required for the simulation increase. The data model has the highest simulation speed and efficiency because there is no consideration of the cluster topology inside it. The number of nodes, topology, and specifications are simply abstracted numerically inside the model. The low speed of the simulation model is caused by DiskSim model. The proposed model has an intermediate speed between the two models. As a result, we can see that the proposed model has a low *IES* value. It has almost the same efficiency as the data model, but because the accuracy is much higher, it can be considered to be a model that satisfies both efficiency and accuracy. In other words, we can see that the proposed work creates synergy through the cooperative modelling for performance analysis.

5. CONCLUSION

How well a modeller builds a model is an important issue for a successful performance analysis of the complex system. Before such a performance analysis, a system model should be constructed using two types of modelling methods: simulation and data modelling. Even though both approaches have pros and cons, it is hard to build a complete model with each model alone. To overcome the weakness, this paper suggested a cooperative modelling approach, able to combine advantages of data/simulation modelling for performance analysis. The cooperative model development process is proposed on the basis of merits and demerits of each model. It can be a systematic and formal method for the modelling of complex system, and follows the general M&S process. The proposed method contains conceptual modelling,

model classification, and model integration/implementation. The model classification method effectively satisfies the criteria made by characteristics of two modelling approach. Then, these are modelled respectively and integrated into the same simulation environment through the integration method. To show how the proposed new approach works, this paper applied it to the Hadoop framework. In the process of modelling the Hadoop, the system is represented by a simulation model using the DEVS formalism and a data model using the ANN, respectively. This experimental result indicated that prediction accuracy and simulation efficiency are relatively enhanced in comparison with the existing approach. In conclusion, we demonstrated how the proposed modelling method could be applied in the complex system for performance analysis. From this paper, we can see how to complement the defects of both methods and to amplify the synergy through appropriate cooperation. In future works, we will research semantics for modelling and integrating these heterogeneous models. We will also conduct additional evaluations including quantitative evaluations for the proposed method.

REFERENCES

- [1] Villalpando, L. E. B.; April, A.; Abran, A. (2014). Performance analysis model for big data applications in cloud computing, *Journal of Cloud Computing*, Vol. 3, No. 1, Paper 19, 20 pages, doi:[10.1186/s13677-014-0019-z](https://doi.org/10.1186/s13677-014-0019-z)
- [2] Hashem, I. A. T.; Yaqoob, I.; Anuar, N. B.; Mokhtar, S.; Gani, A.; Khan, S. U. (2015). The rise of “big data” on cloud computing: review and open research issues, *Information Systems*, Vol. 47, 98-115, doi:[10.1016/j.is.2014.07.006](https://doi.org/10.1016/j.is.2014.07.006)
- [3] Feitelson, D. G. (2002). Workload modeling for performance evaluation, *Proceedings of the 2002 IFIP International Symposium on Computer Performance Modeling, Measurement and Evaluation*, 114-141, doi:[10.1007/3-540-45798-4_6](https://doi.org/10.1007/3-540-45798-4_6)
- [4] Zhang, D. (2017). High-speed train control system big data analysis based on fuzzy RDF model and uncertain reasoning, *International Journal of Computers Communications & Control*, Vol. 12, No. 4, 577-591, doi:[10.15837/ijccc.2017.4.2914](https://doi.org/10.15837/ijccc.2017.4.2914)
- [5] Kim, B. S.; Kang, B. G.; Choi, S. H.; Kim, T. G. (2017). Data modeling versus simulation modeling in the big data era: case study of a greenhouse control system, *Simulation*, Vol. 93, No. 7, 579-594, doi:[10.1177/0037549717692866](https://doi.org/10.1177/0037549717692866)
- [6] Bock, H. G.; Carraro, T.; Jäger, W.; Körkel, S.; Rannacher, R.; Schlöder, J. P. (2013). *Model Based Parameter Estimation: Theory and Applications*, Springer-Verlag, Berlin
- [7] Al-Refai, A.; Li, M.-H.; Jarbo, M.; Yeh, C.-H. B.; Nour, B. (2014). Imprecise data envelopment analysis model for robust design with multiple fuzzy quality responses, *Advances in Production Engineering & Management*, Vol. 9, No. 2, 83-94, doi:[10.14743/apem2014.2.178](https://doi.org/10.14743/apem2014.2.178)
- [8] Kazi, Z.; Radulović, B.; Berković, I.; Kazi, L. (2017). Ontology-based reasoning for entity-relationship data model semantic evaluation, *Technical Gazette*, Vol. 24, Supplement 1, 39-47, doi:[10.17559/TV-20140711141546](https://doi.org/10.17559/TV-20140711141546)
- [9] Christensen, C. M. (2013). *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, 3rd edition, Harvard Business Review Press, Boston
- [10] Li, X.; Yeh, A. G. O. (2001). Calibration of cellular automata by using neural networks for the simulation of complex urban systems, *Environment and Planning A: Economy and Space*, Vol. 33, No. 8, 1445-1462, doi:[10.1068/a33210](https://doi.org/10.1068/a33210)
- [11] Pijanowski, B. C.; Brown, D. G.; Shellito, B. A.; Manik, G. A. (2002). Using neural networks and GIS to forecast land use changes: a land transformation model, *Computers, Environment and Urban Systems*, Vol. 26, No. 6, 553-575, doi:[10.1016/S0198-9715\(01\)00015-1](https://doi.org/10.1016/S0198-9715(01)00015-1)
- [12] Almeida, C. M.; Gleriani, J. M.; Castejon, E. F.; Soares-Filho, B. S. (2008). Using neural networks and cellular automata for modelling intra-urban land-use dynamics, *International Journal of Geographical Information Science*, Vol. 22, No. 9, 943-963, doi:[10.1080/13658810701731168](https://doi.org/10.1080/13658810701731168)
- [13] Rand, W. (2006). Machine learning meets agent-based modeling: when not to go to a bar, *Proceedings of the 2006 Conference on Social Agents: Results and Prospects*, 9 pages

- [14] Elbattah, M.; Molloy, O. (2016). Coupling simulation with machine learning: a hybrid approach for elderly discharge planning, *Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 47-56, doi:[10.1145/2901378.2901381](https://doi.org/10.1145/2901378.2901381)
- [15] Wu, X.; Liu, Y.; Gorton, I. (2015). Exploring performance models of Hadoop applications on cloud architecture, *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*, 93-101, doi:[10.1145/2737182.2737197](https://doi.org/10.1145/2737182.2737197)
- [16] Liu, Y.; Li, M.; Alham, N. K.; Hammoud, S. (2013). HSim: a MapReduce simulator in enabling cloud computing, *Future Generation Computer Systems*, Vol. 29, No. 1, 300-308, doi:[10.1016/j.future.2011.05.007](https://doi.org/10.1016/j.future.2011.05.007)
- [17] Wang, G.; Butt, A. R.; Pandey, P.; Gupta, K. (2009). Using realistic simulation for performance analysis of MapReduce setups, *Proceedings of the 1st ACM Workshop on Large-Scale System and Application Performance*, 19-26, doi:[10.1145/1552272.1552278](https://doi.org/10.1145/1552272.1552278)
- [18] Ferayorni, A. E.; Sarjoughian, H. S. (2007). Domain driven simulation modeling for software design, *Proceedings of the 2007 Summer Computer Simulation Conference*, 297-304
- [19] Adams, J. K.; Thomas, D. E. (1996). The design of mixed hardware/software systems, *Proceedings of the 33rd Annual Design Automation Conference*, 515-520, doi:[10.1109/DAC.1996.545630](https://doi.org/10.1109/DAC.1996.545630)
- [20] Sung, C.; Kim, T. G. (2012). Collaborative modeling process for development of domain-specific discrete event simulation systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42, No. 4, 532-546, doi:[10.1109/TSMCC.2011.2135850](https://doi.org/10.1109/TSMCC.2011.2135850)
- [21] Zeigler, B. P.; Praehofer, H.; Kim, T. G. (2000). *Theory of Modeling and Simulation*, 2nd edition, Academic Press, Amsterdam
- [22] Vieira, A. A. C.; Dias, L. M. S.; Santos, M. Y.; Pereira, G. A. B.; Oliveira, J. A. (2018). Setting an Industry 4.0 research and development agenda for simulation – a literature review, *International Journal of Simulation Modelling*, Vol. 17, No. 3, 377-390, doi:[10.2507/IJSIMM17\(3\)429](https://doi.org/10.2507/IJSIMM17(3)429)
- [23] Narendra, K. S.; Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks, *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 4-27, doi:[10.1109/72.80202](https://doi.org/10.1109/72.80202)
- [24] Apache Hadoop. Hadoop, from <http://hadoop.apache.org>, accessed on 10-09-2018
- [25] Dean, J.; Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters, *Communications of the ACM*, Vol. 51, No. 1, 107-113, doi:[10.1145/1327452.1327492](https://doi.org/10.1145/1327452.1327492)
- [26] Shvachko, K.; Kuang, H.; Radia, S.; Chansler, R. (2010). The Hadoop distributed file system, *Proceedings of the 26th Symposium on Mass Storage Systems and Technologies*, 10 pages, doi:[10.1109/MSST.2010.5496972](https://doi.org/10.1109/MSST.2010.5496972)
- [27] Bucy, J. S.; Schindler, J.; Schlosser, W.; Ganger, G. R. (2003). *The Disksim Simulation Environment Version 3.0 Reference Manual*, Carnegie Mellon University, Pittsburgh
- [28] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters, *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, 431-441, doi:[10.1137/0111030](https://doi.org/10.1137/0111030)
- [29] Huang, P.; Lin, F.; Xu, L. J.; Kang, Z. L.; Zhou, J. L.; Yu, J. S. (2017). Improved ACO-based sweep coverage scheme considering data delivery, *International Journal of Simulation Modelling*, Vol. 16, No. 2, 289-301, doi:[10.2507/IJSIMM16\(2\)9.385](https://doi.org/10.2507/IJSIMM16(2)9.385)
- [30] Kim, T. G.; Sung, C. H.; Hong, S.-Y.; Hong, J. H.; Choi, C. B.; Kim, J. H.; Seo, K. M.; Bae, J. W. (2011). DEVSim++ toolset for defense modeling and simulation and interoperation, *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Vol. 8, No. 3, 129-142, doi:[10.1177/1548512910389203](https://doi.org/10.1177/1548512910389203)
- [31] Kim, B. S.; Kim, T. G. (2017). Cooperation between data modeling and simulation modeling for performance analysis of Hadoop, *Proceedings of the 2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 1-7, doi:[10.23919/SPECTS.2017.8046769](https://doi.org/10.23919/SPECTS.2017.8046769)
- [32] Bruyer, R.; Brysbaert, M. (2011). Combining speed and accuracy in cognitive psychology: is the inverse efficiency score (IES) a better dependent variable than the mean reaction time (RT) and the percentage of errors (PE)?, *Psychologica Belgica*, Vol. 51, No. 1, 5-13, doi:[10.5334/pb-51-1-5](https://doi.org/10.5334/pb-51-1-5)