Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(1): 65-67

Research Article ISSN: 2394 - 658X

Process Model for Performance Modeling and Performance Evaluation of DDS Applications

S Jagannatha, T V Suresh Kumar and Rajanikanth

Department of Computer Applications, M S Ramaiah Institute of Technology (MSRIT), Bangalore, INDIA
jagannatha@msrit.edu

ABSTRACT

In Distributed Database System (DDS), applications require that performance to be determined preferably during the early stages of software development due to the complexity of Distributed system. Performance of DDS applications is determined by considering the different characteristics namely fragmentation, allocation, load sharing, resource allocation, effort estimation, reallocation, etc. Evaluation of the performance at the end of DDS applications development leads to increase in the cost of design change. Software Performance Engineering (SPE) is a field of software engineering, which provides a lot of techniques to predict the performance of the system before its implementation. Hence we propose a process model for modeling and predicting the performance of DDS by adopting SPE approach. Thus, there is a need for a process model that represents the tasks of performance prediction by considering the characteristics DDS applications. In this paper, we describe a process model, Distributed Database System Performance Prediction Process model that allows modeling and evaluating DDS with the explicit goal of assessing performance of the software system by considering the characteristics of database using SPE approach.

Key words: Performance engineering, UML, distributed database

---

INTRODUCTION

Distributed Database System is one of the upcoming areas in the research / industry for building complex distributed application. DDS encompasses multiple features for distributed application. For complex applications, the non-functional requirements have the importance as functional requirements. The non-functional requirements are performance, reliability, maintainability, etc. We are focusing on the prediction of performance of DDS by considering the characteristics of DDS using SPE approach. The evaluation of performance at the end of DDS applications leads to increase in the cost of design if changes are necessitated because of performance reasons. We propose a process model for predicting the performance of a Distributed Database System by considering the characteristics of DDS using Software Performance Engineering (SPE). SPE provides a systematic, quantitative approach to constructing software systems that meet performance objectives during early design stages. We adopt SPE approach in DDS characteristics for prediction of performance during preliminary design stages.

THE PROPOSED PROCESS MODEL

The proposed model for predicting the performance of DDS using SPE is expressed in the form of the flowchart in Fig. 1. The activities involved in the elements of the process model are:

Step 1: Model the DDS Applications using UML

DDS is formed by the fragmentation and allocation of distributed applications. The important factors are that affect the performances of the distributed database system are fragmentation, allocation, resource allocation, workload sharing, dynamic reallocation, system architecture, data distribution, frequency of arrival of queries, local access, and remote access, etc. Fragmentation is based on the set of transaction requirements. Allocation is placing the

fragments into the nodes where the transaction requires locally. Considering these issues of DDS described above, model the DDS applications using Unified Modelling Language (UML).

Step 2: Study the Characteristics of DDS

We consider different approaches for performance analysis of DDS applications during the preliminary design stages by considering the DDS characteristics namely: fragmentation, allocation, reallocation, dynamic reallocation, resource allocation, assessment of workload and Effort estimation, The other approach for early design stage performance analysis is SPE(Software Performance Engineering).

Step 3: Develop UML Models

DDS applications of software specifications are represented in UML. Many approaches in UML consider the design and modelling of DDS applications. The UML model characteristics of DDS have a high impact on the performance of the system. We propose to use UML as the modelling language for modelling the DDS characteristics for predicting the performance. The UML Profile for DDS provides modelling constructs that enable the use of Model-Driven Development and Architecture practices to meet these challenges for large-scale DDS applications. UML captures the performance requirement data by modelling the application using a modelling technique. Since UML is a universally accepted modelling language, we have extended the UML to model the applications developed. UML Profile defines a collection of constructs that represent, Data Centric Publish, subscribe Entities and data construction layer.

Step 4: Apply Techniques to Obtain Performance Metrics:

We require few techniques to estimate performance during early stages, i.e. with high level information. We have constructed a methodology to predict the performance metric by considering the resource allocation, dynamic reallocation, use case approach. We also constructed mathematical models for the characteristics of DDS such as Shapley value. The models are solved, and performance metrics is obtained. The models can be solved analytically and/or simulation. Simple systems can be solved analytically, whereas simulation is preferred for larger and complex systems. So we propose methodologies for solving the models analytically and/or simulation.

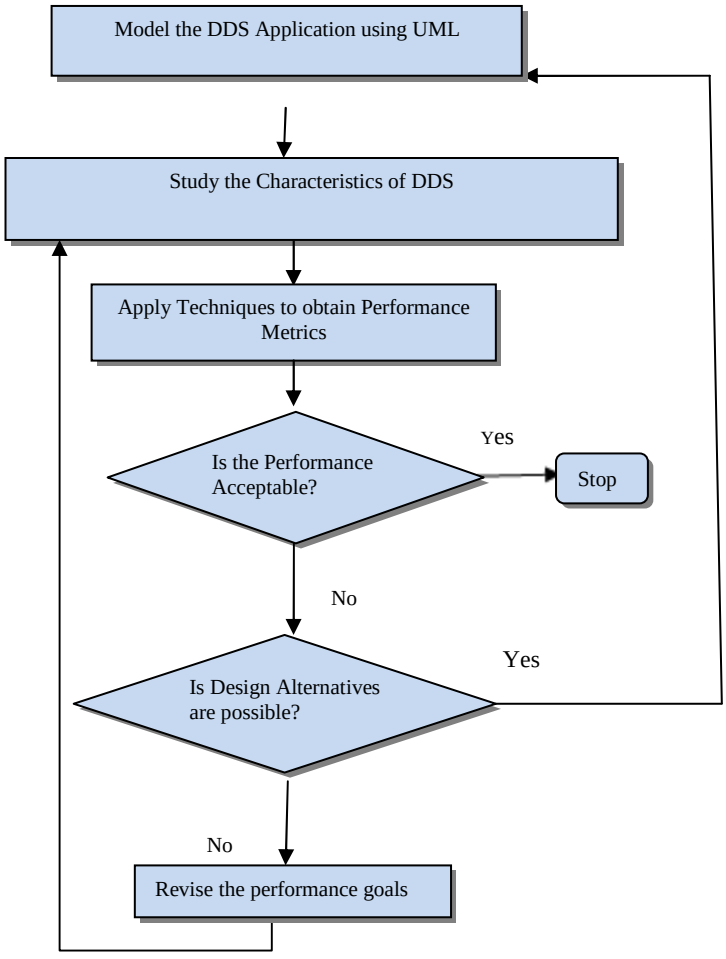

Fig.1 Process Model for Performance Prediction

Step 5: Report the Performance Metrics
If the performance metrics obtained by considering a characteristic of DDS is acceptable, then stop the procedure for performance prediction.

Step 6: Alternatively
If the performance metric obtained for the particular characteristic is not acceptable, if design alternatives are possible, then look for different design models and repeat the same procedure for the model the DDS application using UML.

Step 7: Alternatively
If the design alternatives are not possible, then revise the performance goals as defined at the beginning of design stage, and repeat the same procedure to define the SPE assessment for the given DDS till the obtained performance metrics is acceptable.

CONCLUSION

In this paper a process model for Performance Modeling and Performance Evaluation of a DDS Application is proposed. The process model defines the steps to evaluate and assess the performance of DDS applications during preliminary design stages with the goal of assessing performance using the SPE approach. The process model uses the considering the various characteristics of DDS applications namely fragmentation, allocation, resource sharing, load balancing shapely value etc. The process model describes the elements of DDS application and provides flexibility to integrate the software performance prediction process. The proposed process model provides the possibility of Predicting the performance of DDS applications using resource sharing, shapely value for work load assessment. The characteristics of DDS applications are model using UML and simulate with different methodologies. The obtained results are compared with performance goals, and any design alternative is suitable for performance evaluation. The process is repeated with different alternative to achieve performance goals. The frame work is used for database designer to predict and assess the performance of DDS application during early stages of SDLC.

REFERENCES

[1] M Özsu, and P Valduriez, *Principles of Distributed Database Systems*, New Jersey, Prentice-Hall, **1991**.
[2] S Ceri and G Pelagatti. *Distributed Databases Principles and System*, 1st Edi. McGraw-Hill, New York, **1984**.
[3] J J Shena and T L Hungb, Minimizing Transmission Cost for Distributed Database Design, *Proceedings of the 17th International Conference on Advanced Information Networking and Applications*, **2003,** p. 638-641.
[4] F Baiao, M Mattoso and G Zaverucha, A *Framework for the Design of Distributed Databases*, Computer Science Department, **2002.**
[5] S F El-Zoghdy, M Nofal, M A Shohla and A El-sawy, An Efficient Algorithm for Resource Allocation in Parallel and Distributed Computing Systems. *International Journal of Advanced Computer Science and Applications, IJACSA*, **2013,** 4(2):251-259.
[6] C U Smith, *Performance Engineering of Software Systems*. Addison Wesley, **1990.**
[7] C U Smith and L G Williams, *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison Wesley, **2002.**
[8] C U Smith and L G Williams. Performance Engineering Evaluation of Object-Oriented Systems with SPE.ED2. *LNCS, Springer*, **1997,** 1245, 135-153.
[9] L G Williams and C U Smith, PASA: A Method for the Performance Assessment of Software Architectures. In *Proceedings of the International workshop on Software and Performance, ACM*, **2002,** p.179-189.
[10] L G Williams and C U Smith, Performance Evaluation of Software Architectures. In *Proceedings of the International workshop on Software and Performance, ACM*, **1998,** p. 164-177.
[11] C U Smith and L G Williams, Building Responsive and Scalable Web Applications, In *Proceedings of Computer Measurement Group Conference*, **2000.**
[12] C U Smith and L G Williams, Performance Engineering Models of CORBA-based Distributed-Object Systems. In *Proceedings of the Performance Engineering Services and Software Engineering Research*, **1998.**
[13] D C Petriu and H Shen, Applying the UML Performance Profile: Graph Grammar-Based Derivation of LQN Models from UML Specifications, In *Proceedings of the Seventh International Conference on Modelling Technique and Tools for Performance Evaluation*, **2002,** p.159-177**.**
[14] M Woodside and D C Petriu, *Performance by Unified Model Analysis (PUMA)*, WOSP'05, ACM, **2005**.
[15] D A MenasceÂ and H Gomaa, A Method for Design and Performance Modelling of Client/Server Systems, *IEEE Transactions on Software Engineering*, **2000,** 26(11), 1066-1085.
[16] M D Miguel, UML Extensions for the Specifications and Evaluations of Latency Constraints in Architectural Models, In *Proceedings of the Second International Workshop on Software and Performance*, **2000,** p.83-88.