Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2, Issue 12, December -2015

SURVEY: ANALYSIS OF VARIOUS MESSAGE PASSING INTERFACE IMPLEMENTATIONS

Atish Ghone[1], Dr. J.S. Umale[2]

[1,2]*Computer Engineering, PCCOE pune*

Abstract *— Now a days the more number of the applications are developed on the parallel platform. The parallel applications are may or may not on the same nodes, that requires the way of the communication and hence in high performance computing the Message Passing Interface comes into the picture. The implementation of the message passing interface is done for the various requirements that may be the vendor specific or the open source operated for any type of the available network. So the MPI libraries provide the functionality that is the vendor's system specific or general services. If the services are generally then what about the performance of the MPI libraries on the different platforms, so that there is a need to analyze these all the questions are needed to be studied very briefly so that the efficient and well structured lightweight MPI libraries can develop for the faster communication having the lowest latency. Second part is that to understand the how MPI is going to use the services provided by the OFED (Open Fabrics Enterprise Distribution). Also to understand the type of the verbs and OFED services utilized by the MPI.*
In this paper, we focused on comparative analysis of various MPI library implementations.

Keyword s- Message Passing Interface, OFED, uDAPL, infiniband ,verbs .

I. INTRODUCTION

The MPI is the first standardized, vendor independent, message passing library. The advantages of developing message passing software using Message Passing Interface closely match the design goals of portability, efficiency, and flexibility**.** Basically the MPI is the library of functions that is used to create parallel programs. This library runs with standard C and programs by using the operating system services to create parallel processes and exchanging the information between them.

The Infiniband Architecture system can range from a small server with one processor and a few I/O devices towards a massively parallel supercomputer installation with hundreds of processors and thousands of input and output devices. In small configurations, a processor node may be connected to I/O nodes directly. This infiniband is the switched communications fabric allowing many devices to concurrently communicate with high bandwidth and low latency in a protected and remotely managed environment. In which the end node is able communicate over multiple infiniband architecture's ports and also utilize multiple paths through the infiniband fabric. Basically the software architecture is based on the Open Fabrics Enterprise Distribution (OFED) Architecture.

Open Fabrics Enterprise Distribution (OFED) is open source software, committed to provide a common communication stack to all RDMA capable System Area Networks (SANs). It supports high performance MPIs and legacy protocols for HPC domain and Data Centre community. Currently, it supports InfiniBand (IB) and Internet Wide Area RDMA Protocol (iWARP) [3].Verbs in the OFED is responsible for component identification, initialization and registration with the OS to obtain kernel mode resources. Most important functionality of the verb is the abstraction of the definition of functionality provided with a host by a channel interface.

The second functionality contained in the OFED stack is the User Direct Access Programming Library (uDAPL). Which defines a single set of user Application Programming Interfaces for Remote Direct Memory Access (RDMA) capable transports. Developers of uDAPL have to write test programs for verification of APIs and for integrated testing of the software stack along with the underlying hardware.

This paper elaborates the different message passing library implementations, which required understanding that whether the need for communication between processes in high performance computing is satisfied clearly or not. Also observe the basics behind the implementation of the message passing libraries and analyze the performance of the message passing libraries.

II. LITERATURE REVIEW

1. **MPI over uDAPL: Can High Performance and Portability Exist Across Architectures?**
   Author. Lei Chai,Ranjit Noronha, Dhabaleswar K. Panda

In the work, MPI over uDAPL: Can High Performance and Portability Exist Across Architectures? (2006) [1], by using the uDAPL interface they designed the Message Passing Interface which having the characteristics both portability and the portable high performance. By using the facility of uDAPL that is, it provides the application's communication regardless that of network, architecture and operating system. Some new applications may take advantage of the uDAPL interface, a large number of applications exist which are written in MPI. So to port these applications to uDAPL may be

significant in concern with the given their size and complexity. In some cases, it may be impractical, especially if the expertise needed to rewrite these applications is not available.

So that in this work developed the MPI over the uDAPL.

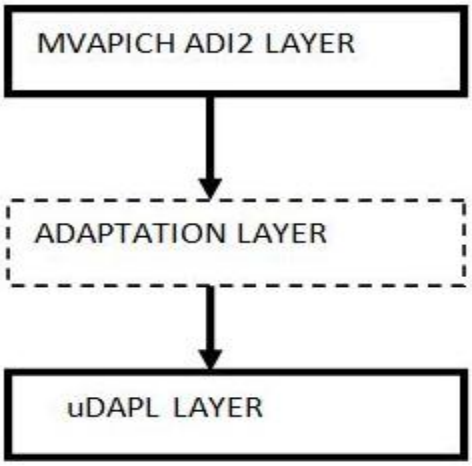

Fig. MPI over uDAPL

The above figure shows the implementation of the MPI over the uDAPL. The dotted line box is the implementation of layer that map the MPI level services to uDAPL layer. The library of MPI used is MVAPICH, which has four major components, these are connection management, communication channels, progress engine and memory management. And the uDAPL provides various services. So designing the high performance MPI over uDAPL, they mapped the MVAPICH components onto uDAPL services. As shown in above figure. The MPI provides a fully connected topology and the uDAPL currently only supports Reliable Connection (RC) service, that is every process needs to establish a connection with every other process at the initialization phase. To accommodate this server-client model is used for connection establishment. Also multi stream connection is designed through which the multiple connections are being set up (i.e. Multiple pairs of End Points) between every two processes so that the achieved aggregate bandwidth is high.

2. **SoC-MPI: A flexible Message Passing Library for Multiprocessor Systems-on-Chips**

Author. Philipp Mahr, Christian L¨orchner, Harold Ishebabi and Christophe Bobda

SoC-MPI: A flexible Message Passing Library for Multiprocessor Systems-on-Chips (2009) [2] , gives the message passing interface library implementation of the hardware architecture of the System On Chip. In these types of high performance systems, there is highly need of the communication between components. This paper argued that the MPI paradigm has the best speedup potential in Multiprocessor System-on-Chip.In this MPI based communication library is implemented, which responsible to give the user an easy to handle interface for describing communication between processing elements and also gives the characteristic of good performance in terms of latency and bandwidth of the underlying communication networks. This implementation does not require an operating system and is in terms of functionality needed by an application. At the time of the configuration of the library, because of the library is dependent on the underlined communication network it may not include the functionality that does not require for underlined communication network hardware. This makes the lightweight library.

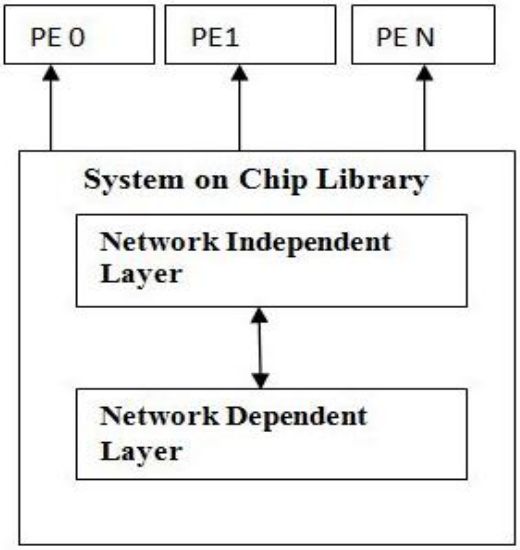

Fig. SoC MPI library flow.

This figure shows that the working and flow of the System on Chip MPI library.As shown in figure the library is divided into a Network Independent Layer (NInL) and a Network Dependent Layer (NDeL). The reason behind the dividation of library into two layers is the separation of the functionalities that the library supports more efficiently these are platform independent and platform dependent layers. Network Independent Layer is responsible for the actual implementation of the MPI functionality by using the standardized basic functions. And the Network Dependent Layer provides the basic communication functions for communication networks, as well as hardware supported functions, like broadcast.

3. Supporting OFED over Non-InfiniBand SANs

Author. Devesh Sharma.

They gives the way to support Open Fabrics Enterprise Distribution software stack over non-Infiniband Remote Direct Memory Access capable SAN. In this they proposed the design of Virtual Management Port (VMP) so as to enable the Infiniband subnet management model. Also the integration of VMP with IB-Verbs interface driver prevents a hardware and OFED modification which enables the connection manager that is required to run user applications.

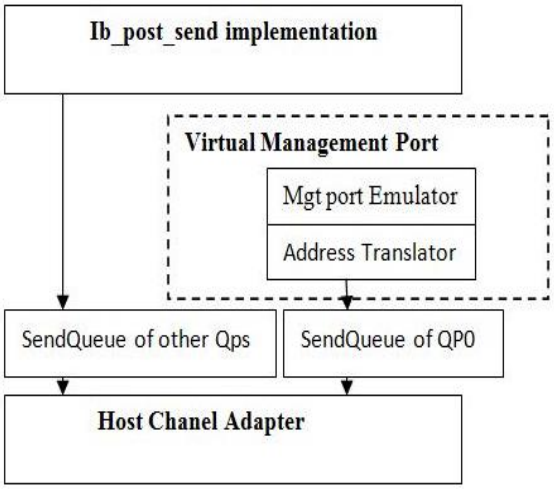

Fig. Architecture of VMP implementation

The above figure shows the implementation of the Virtual Management Port, which is responsible for enabling the connection manager so that the Infiniband subnet management model supports without changing the OFED and underlying hardware. The functionality of VMP is to emulate Infiniband switch management-port and route Directed Routed Subnet
Management Packets (DR-SMP). To achieve this VMP is integrated with Infiniband-Verbs interface.

III. OBSERVATIONS

In the first work, there is the design of the MPI libraries over the uDAPL interface. They are not going to modify the actual services provided by the uDAPL layer. In this work the MVAPICH is adapted over the uDAPL interface by inserting a layer called Adaptation layer in between them. Also the communication is done by using the connection establishment through the client server model for the connection between the communicating processes.

Likewise, in the second work towards WCET estimate, it also designs the Message Passing Interface Library, but it is especially for the System on Chip architectures. In this the work they developed the library in two modules that is network dependent functionality and network independent functionality, it makes that the designed library is easy for the user. And also the tasks are getting divided to make the lightweight message passing library.

In the third work, the Open Fabrics Enterprise Distribution stacks which work as MPI only for the Infiniband fabrics. But by inserting the Virtual Management Port (VMP).The VMP is responsible for the work as an emulator between the Infiniband verbs and the Host Channel Adapter.

IV. RESULTS AND INTERPRETATIONS

In the first work, Conjugate Gradient (CG) and Integer Sort (IS) are selected from the NAS parallel benchmarks, because these two benchmarks mainly use large messages. CG and IS, Class A benchmarks were running on 2 and 4 processes. They used a small number of processes for CG and IS, because as the number of processes increases the message size decreases, which makes the benchmarks not bandwidth sensitive any more. In this work the multi stream design can improve the performance of CG by 11% and 8% on 2 and 4 nodes respectively, and improves the performance slightly.

In second work, the size of the implemented SoC-MPI library is 13584 Bytes which includes the MPI_Init, MPI_Finalize, MPI_Wtime, MPI_Send, MPI_Receive and MPI_Barrier functions for that the total amount of slices needed for the system with five Micro Blazes was 12343 Slices (90% of the available slices) and almost all Block-RAM (94%).

In third work, provide design that the Open Fabric Enterprise Distribution stack for the non-Infiniband SAN. For every sweep cycle OpenSM injects more and more DR-SMPs into the network, because of the cluster size increases, which is responsible for increasing the overhead on the SM-node and decreases the application performance.

The below table shows the interpretation about the studied papers

Paper No.	Paper Title	Observations				
		Kind of SAN	Application	Protocol Stack	Need For Impln	Drawbacks
1	MPI over uDAPL: Can High Performance and Portability Exist Across Architectures?(2006 IEEE)	Infiband	NAS parallel Benchmark	OFED	MPI impln portable across different networks and architecture	Less efficient and low BW
2.	SoC-MPI: A flexible Message Passing Library for Multiprocessor Systems-on-Chips(2008 IEEE)	System on Chip	Singular Value Decomposition	OFED	the use of many MPI functions can lead to a heavyweight library, which can become a problem in chip multiprocessors	Lack of synchronous sending, tuning library in terms of speed and size
3	Supporting OFED over Non-InfiniBand SANs(2010 IEEE)	Non IB	CDAC proprietary SAN	OFED	to support OFED software stack over non-IB SAN	Overhead of address translation, gives less performance

Fig.Comparative Analysis

V CONCLUSION AND FUTURE WORK

In this survey work, the comparative analysis of various implementations of the Message Passing Interface library for the different types of the system networks such as Infiniband, non-Infiniband or System on Chip architecture is elaborated precisely.This analysis is required for to study the MPI basics and to develop the system specific MPI library. Also, this gives the overall functionality of MPI while performing operations on different systems.

In future, it can be possible that by using the this detailed survey of MPI library design and implement the lightweight Message Passing Interface Libraries that provides the high performance over all types of messages .

VI REFERENCES

[1] Lei Chai Ranjit Noronha Dhabaleswar K. Panda, "MPI over uDAPL: Can High Performance and Portability Exist Across Architectures? ",Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06) 0-7695-2585-7/06 $20.00 © 2006 IEEE

[2] Philipp Mahr, Christian L¨orchner, Harold Ishebabi and Christophe Bobda, "SoC-MPI: A flexible Message Passing Library for Multiprocessor Systems-on-Chips", 2008 International Conference on Reconfigurable Computing and FPGAs**.**

[3] Devesh Sharma. Supporting OFED over Non-InfiniBand SANs. 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing.

[4] William Gropp and Ewing Lusk, "The MPI Communication Library: Its Design and a Portable Implement ation", 0-8186-498O-lB$43 .00Q 1994 IEEE.

[5]InfiniBand™ Host Channel Adapter Verb Implementer's Guide**,** Copyright Intel® Corporation 2003

