The Performance Of Erasure Codes Used In FT-MPI

Liu Xiaoguang [1], Wang Gang [1], Zhang Yu [1], Li Ang [1]
[1] Nankai-Baidu Joint Lab, College of Information, Nankai University, Tianjin, 300071, China

liuxg74@hotmail.com

ABSTRACT: Today, the scale of High performance computing (HPC) systems is much larger than ever. Some HPC systems consist of thousands or even tens of thousands of processors. The larger scale leads to a challenge that how to deal with process failures. The most important programming tool for HPC is MPI (Message Passing Interface). There are some existing methods to deal with fault-tolerance, such as MPICH-V, StarFish, MPI/FT and so on, using the MPI context. Most of them do the checkpoint on disk. In this paper, some erasure codes, which used in RAID systems usually, are applied to deal with the fault-tolerance in-memory. Based on fault-tolerance-MPI (FT-MPI) platform, RAID4, RAID5, RDP and X-Code are implanted to do the checkpoint in-memory. The experimental results show that RDP is feasible for double-fault-tolerance in-memory.

KEYWORDS: MPI; fault-tolerance; RDP; X-Code

I. INTRODUCTION

Today, the scale of High performance computing (HPC) systems is much larger than ever. Some systems consist of thousands or even tens of thousands of processors. According to the data from Top500 in June, 2003, most systems (54.8%) include 129-512 processors. But in November, 2007, most systems (53.6%) includes 1025-2048 processors. The fastest system in 2007 consists of more than twenty thousands processors[1].

A challenge to the systems which include so many processors is the ability to deal with hardware failures. Concluding from the current experiences on the top-end machines, a 100,000-processor machine will experience a process failure every few minutes. In order to solve this problem, some fault tolerance mechanisms must be considered in both hardware and software design.

The most important programming tool for HPC is MPI (Message Passing Interface). Combining MPI and fault tolerance, users can improve the reliability of applications. Usually, this MPI is called fault-tolerance-MPI (FT-MPI). There are three kinds of fault-tolerance implementation for MPI. Firstly, fault-tolerance mechanism can be implemented by global checkpoint. After failure occurring, the MPI applications are restored to recent checkpoint. Secondly, fault-tolerance mechanism also can be implemented based on event-log. At last, fault-tolerance mechanism can be implemented using replications.

Cocheck is a fault-tolerance MPI based on checkpoint[2]. StarFish, another fault-tolerance MPI, is similar to Cocheck[3]. Cocheck is developed using Codor, but StarFish is developed on a distributed system.

Log-based fault-tolerance MPI has two log Strategies, one is optimistic log, the other is pessimistic log. MPI-FT[4] is a fault-tolerance mechanism implemented on LAM-MPI[5]. MPI-FT can use optimistic or pessimistic log according to user's requirements. MPICH-V2[6] use Checkpoints Scheduler to maintain consistence.

In this paper, some error-correcting codes, which used in RAID systems usually, are applied to deal with the fault-tolerance in-memory. Some typical construction methods of single and double erasure codes, such as RAD4, RAID5, Row-Diagonal Parity (RDP)[7] and X-code[8], are implemented on FT-MPI.

II. ARCHITECTURE OF FT-MPI

There are three main parts in FT_MPI, detection, notification and recovery. The architecture of FT-MPI is shown in figure 1.

FT-MPI system is built in a hierarchical structure. MPI Library layer is the API to applications. The second layer is responsible for data transfer. FT-MPI runtime library (FTRTL) is responsible for interaction with operation system through Harness library. FTRTL can receive failure notification from both communication library and Harness library. According to the source of communications, FTMPI can determine the reason of failure.

III. FAULT-TOLERANCE IN-MEMORY USING ERASURE CODES

Based on the FTMPI platform, some erasure codes are used to implement fault-tolerance mechanism in-memory.

A. Single erasure codes

In this paper, RAID 4 and RAID 5 are used as single erasure codes for fault-tolerance in-memory.

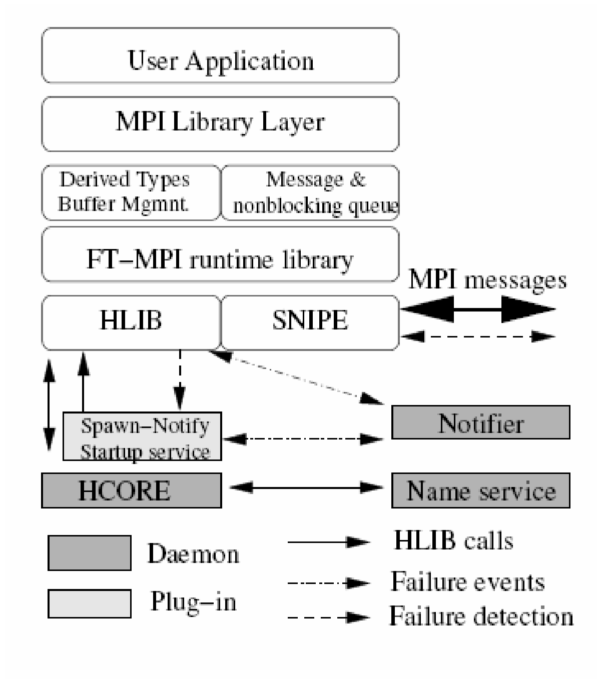
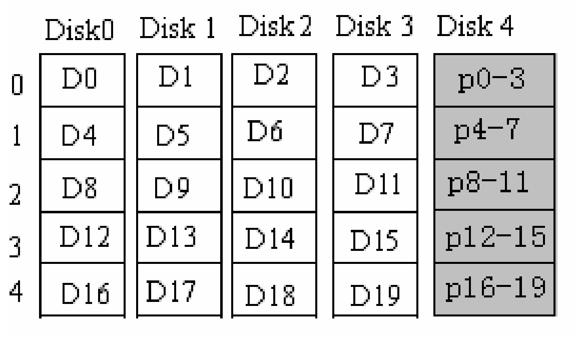


Figure 1. architecture of FT-MPI

As shown in figure 2, all parity data are stored in a singe disk. Using the idea of RAID 4, a process can be considered a disk figure 2.



	Disk0	Disk 1	Disk 2	Disk 3	Disk 4
0	D0	D1	D2	D3	p0–3
1	D4	D5	D6	D7	p4–7
2	D8	D9	D10	D11	p8–11
3	D12	D13	D14	D15	p12–15
4	D16	D17	D18	D19	p16–19

Figure 2. Data layout of RAID 4

Further more, the last process play a role of redundancy process like parity disk in figure 2. The processes are shown in figure 3. And the data recovery process is shown in figure 4.

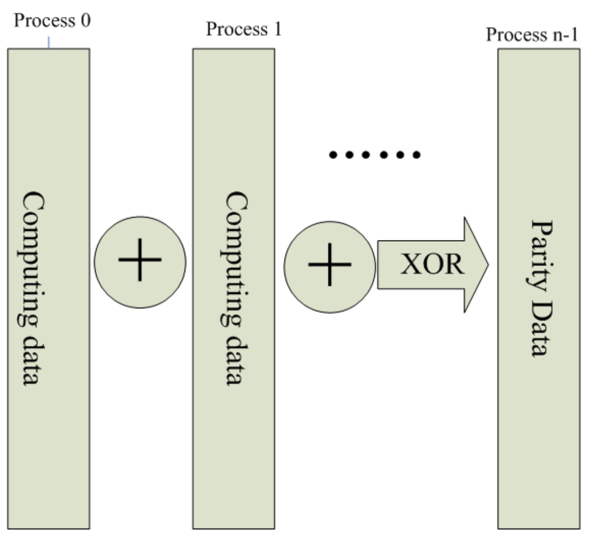


Figure 3. RAID 4 in FT-MPI

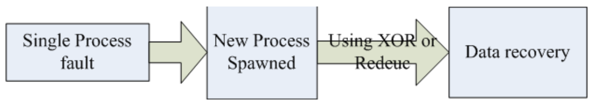
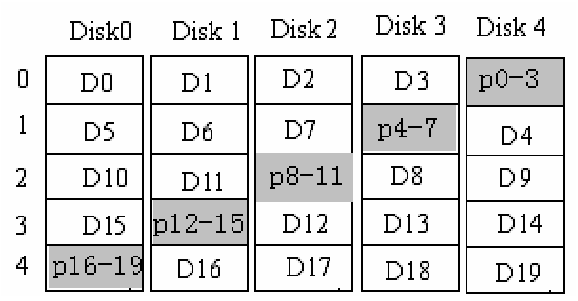


Figure 4. the data recovery of RAID 4

According to RAID 5 data layout, as shown in figure 5, the parity data are distributed to all disks. This data layout can improve system performance effectively.



	Disk0	Disk 1	Disk 2	Disk 3	Disk 4
0	D0	D1	D2	D3	p0–3
1	D5	D6	D7	p4–7	D4
2	D10	D11	p8–11	D8	D9
3	D15	p12–15	D12	D13	D14
4	p16–19	D16	D17	D18	D19

Figure 5. the data layout of RAID 5

Appling RAID 5 to FTMPI, the data in process is divided into stripes liking the stripes in RAID 5. Because all the jobs are done in memory, the integrity of data is not destroyed.

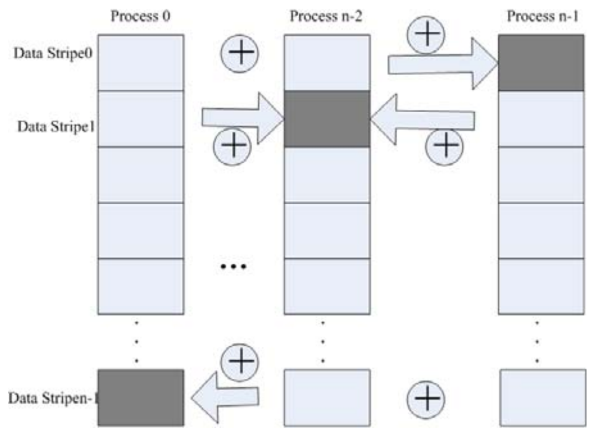


Figure 6. RAID 5 in FTMPI

The data recovery process is shown in figure 7.

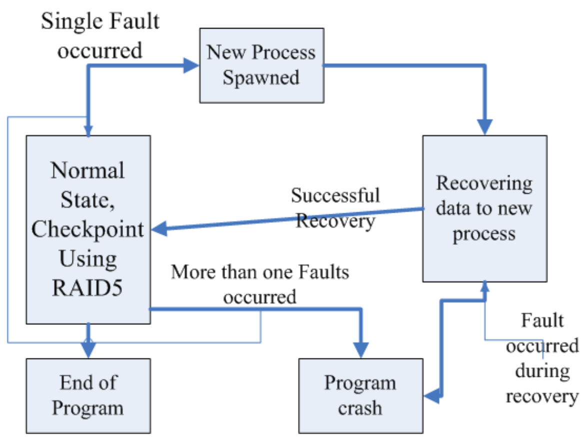

Single Fault occurred
New Process Spawned
Normal State, Checkpoint Using RAID5
Successful Recovery
Recovering data to new process
More than one Faults occurred
Fault occurred during recovery
End of Program
Program crash

Figure 7. the data recovery of RAID 5

B. *Double erasure codes*

RAID 4 and RAID 5 can only deal with single fault events. In order to handle double faults, some double erasure codes are implemented in FTMPI.

RDP is a kind of double erasure code. The parity data is gotten only using XOR operation. The data layout of RDP with 4 data disks is shown in figure 8.

DataDisk 0	DataDisk 1	DataDisk 2	DataDisk 3	Row Parity	Diag. Parity
0	1	2	3	4	0
1	2	3	4	0	1
2	3	4	0	1	2
3	4	0	1	2	3

Figure 8. data layout of RDP (4 data disks)

The implementation of RDP in FTMPI is shown in figure 9.

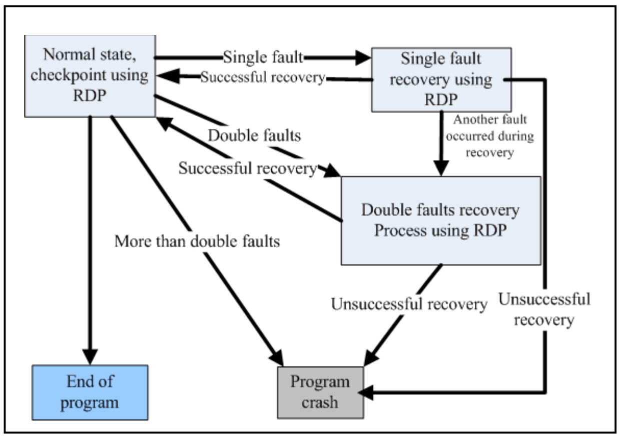


Figure 9. The data recovery of RDP

X-Code is another double erasure code used in FT-MPI. In X-Code, all data units are stored in a（n-2）*n matrix. The parity units are stored in a 2*n matrix. They construct a n*n matrix.

Considering   as the data in the ith line and jth column, the parity of X-Code can be described as follows,

$$C_{n-2,i} = \sum_{k=0}^{n-3} C_{k,<i+k+2>_n} \quad i=0,1,....,n-1 \quad (1)$$

$$C_{n-1,i} = \sum_{k=0}^{n-3} C_{k,<i-k-2>_n} \quad i=0,1,\square,n-1 \quad (2)$$

The data recovery process of X-Code is shown in figure 10.

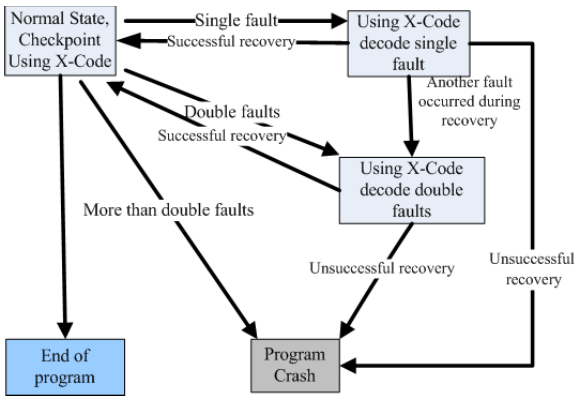


Figure 10. The data recovery of X-code

IV. Experimental results

The experiments to FT-MPI are run on some PC which connected by Gigabit Ethernet. The operation system is Redhat AS 4.0. The program with checkpoint is PCG, which is an iterative algorithm for linear systems.

A. The cost of checkpoint

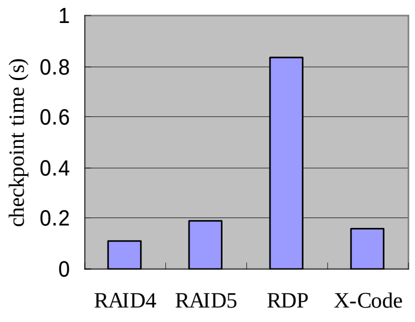


Figure 11. the cost of checkpoint

As shown in figure 11, the checkpoint time of RDP is higher than other codes obviously. The reason is RDP need more XOR calculation than others.

B. The runtime under single fault

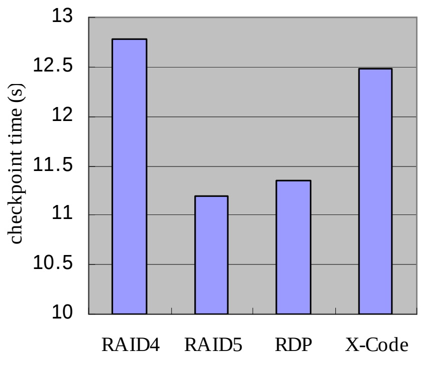


Figure 12. the runtime under single fault

Figure 12 shows the runtime of different erasure codes under single fault. The runtime includes executive time, checkpoint time and recovery time. As shown in figure 12, RDP and RAID5 have shorter runtime.

C. The runtime under double faults

Because only RDP and X-Code support double-fault-tolerance in FT-MPI, Figure 12 shows their runtime under double faults. According to the results, RDP is more effective than X-Code.

V. Conclusions

With the development of HPC, HPC systems need more and more nodes and processor. The larger scale will lead to the decline in system reliability. FT-MPI can tolerate hardware or software faults on program level. In this paper, some erasure codes are used in FT-MPI to improve program reliability. The experimental results show that RDP is feasible for double-fault-tolerance. In the future woks, the erasure codes for more than double faults will be tried in FT-MPI.

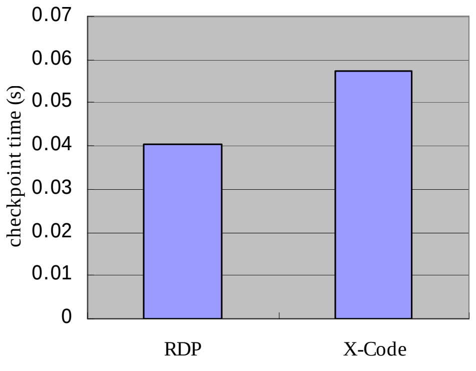


Figure 13. the runtime under double faults

Acknowledgment

This paper is supported partly by the National High Technology Research and Development Program of China (2008AA01Z401), SRFDP of China (20070055054), and Science and Technology Development Plan of Tianjin (08JCYBJC13000).

References

[1] http://www.top500.org/

[2] G. Stellner, "CoCheck: Checkpointing and Process Migration for MPI", Proceedings the International Parallel Processing Symposium, Honolulu, April 1996,pp 526-531

[3] A.Agbaria, R.Friedman, "Starfish: Fault-Tolerant Dynamic MPI Programs on Clusters of Workstations", the 8th IEEE International Symposium on High Performance Distributed Computing,1999.pp167-176

[4] S. Louca, N.Neophytou, etal, "MPI-FT: A portable fault tolerance scheme for MPI", Parallel Processing Letters, Vol. 10, No. 4 (2000) 371-382

[5] R. Graham, S. Choi, etal.,"A Network-Failure-Tolerant Message-Passing System For Terascale Clusters", International Journal of Parallel Programming, Vol.31(4) ,2003, pp285-303

[6] A.Bouteiller,F.Cappello etc., "MPICH-V2: a Fault Tolerant MPI for Volatile Nodes based on Pessimistic Sender Based Message Logging", SC2003, Phoenix, Arizona,2003, pp 25-35

[7] P.Corbett,B.English,etal., "Row-Diagonal Parity for Double Disk Failure Correction", FAST04,2004, San Francisco, CA ,pp1-14

[8] L.Xu, J. Bruck, "Highly Available Distributed Storage Systems", Workshop on Wide Area Networks and High Performance Computing, 1998: 307-330