Fast Map Reduce Algorithms for Exact-Repair Reconstruction of Big-Data in Cloud Storage

Xue Qin
Department of ECE
University of Texas at San Antonio
San Antonio, TX, USA
qinxue1107@gmail.com

Brian Kelley
Department of ECE
University of Texas at San Antonio
San Antonio, TX, USA
Brian.Kelley@utsa.edu

***Abstract*—In distributed storage for big data systems, there is a need for exact repair, high bandwidth codes. The challenge for exact repair in big-data storage is to simultaneously enable both very high bandwidth repair using Map-Reduce, simple coding schemes and to combine this with maximally distance separable (MDS) exact repair for the rare, but exceptional outlier error patterns requiring optimum erasure code reconstruction. We construct the optimum fast bandwidth repair for a big-data source, $\mathcal{A}$, using MAP-Reduce, exact repair reconstruction. The algorithm combines MDS, with a second fast decode algorithm in a cloud environment. We investigate cloud experiments for optimum fast bandwidth reconstruction for 1-Exabyte Big Data in the cloud and demonstrate cloud results for Poisson error rate arrival models.**

I. Introduction

Distributed storages systems in cloud data centers, Data as a Service (DaaS) systems and other big data systems increasingly require fast bandwidth node reconstruction in the event of disk failures. For instance, open-source frameworks such as OpenStack deploy Swift DaaS and Glance image storage. In many instances, there is a need to both achieve optimum guarantees of data recovery in the event of node failure *and* fast bandwidth repair, *two countervailing objectives*. Typically, an erasure code, converts k information symbols using a generator matrix into an N-symbol codeword [1].

We propose the use of maximum distance separable (MDS) coding jointly with a fast reconstruction algorithm for correction (see [2]), *but reformulated in a cloud environment for Big-Data*. The specific pattern of the errors in memory determines whether we can apply fast bandwidth or the optional MDS reconstruction. Given a Poisson error arrival patterns into the big data system, a core concept is this—control the mean number of errors injected into the big data set by applying fast erasure decoding in real time in the cloud at a sufficient rate, $1/\mathrm{T}_{per}$ so that slower MDS reconstruction occurs with low probabilistic guarantees. We define the effective bandwidth achieving this objective, $\mathrm{EffBW} = 1/\mathrm{T}_{per}$. We apply the fast algorithm with probability, $q = f(T_{per}, \lambda)$, thereby enabling mean computational latency of the reconstruction to become

$$\mathrm{D}_{\mu} = (1 - \mathrm{q}) \times \mathrm{L} \times \mathrm{T} + \mathrm{q} \times \mathrm{T} \ll \mathrm{L} \times \mathrm{T} \qquad \text{EQ-1}$$

In Table 1, T and $\mathrm{L} \times \mathrm{T}$ are the computational latency of the fast bandwidth reconstruction and (L-times) longer MDS latency, respectively. A key optimization goal is the determination of the mean upper bound time T_{per} so that the combined rate $2/3$ simple regenerative code with MDS [2] in a cloud environment is dominated by the simple regenerative code latency (e.g. $\mathrm{D}_{\mu}, \ll \mathrm{T}_{per}$). We define our optimality objective as the upper bound correction period, T_{per}, enabling the mean computational delay of the reconstruction system, D_{μ}, to be 1% of T_{per}. Figure 1 illustrates that the fast algorithm delay occurs with high probabilistic guarantees, greatly decreasing $E[\mathrm{D}_{\mu}]$.

Table 1: Cloud application parameters.

	Param.	Description
Inputs to Cloud App.	$\mathcal{A}$	Big memory source, a 1 Exabyte Big Data-set
	λ	Poisson error arrival rate in bits/sec/512 bytes of mem.
	k	MDS code rate, k/N
Run Time Parameters	P	Number of parallel databases, MDS system, that used as the parallel Hadoop Map-Reduce system
	N	The number of nodes in MDS system that used as the parallel Mapping for Hadoop Map-Reduce
	M	Node size memory in MDS system
	Tper	Time step period in seconds at which data bases errors are detected and memory is reconstructed
	(T, T x L)	(Hadoop Fast bandwidth correction latency, MDS correction latency is $L = N/3 + 2N^3/9$ times slower)

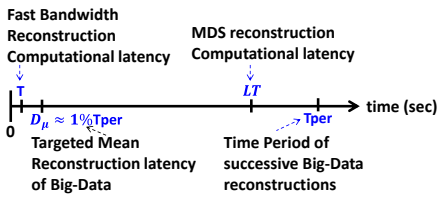

Figure 1: Timeline latency model.

Section II proposes a new Map-Reduce fast-bandwidth regenerative algorithm for Big Data on Clouds. Section III describes fast bandwidth simulation results in the Poisson error arrival model. Section IV outlines the variety of computer science processing protocols to be implemented on the NSF cloud.

II. Fast Bandwidth Cloud Regeneation of Big Data

Table 1 represents the set of parameters used in our cloud framework. Let's assume that our big data set over time has a Poisson error arrival rate in memory of λ $bit\ errors/sec/512\ bytes$ [3]. Let's define the optimum effective bandwidth of cloud reconstruction of the big data set,$\mathcal{A}$, as the value T_{per} that enables D_{μ} to be 1% of T_{per}

STEP 1: For a given λ and big data size, $\mathcal{A}$, we determine the optimum Map-Reduce parallelism P. Based upon λ and the size of $\mathcal{A}$, divide the set $\mathcal{A}$ into $\mathcal{A}$ /P memories.

STEP 2: Encode via MAP Reduce each of the $\mathcal{A}$ /P memories with a combined rate 2/3 simple regenerative code and MDS code. The MDS code further divides each of the $\mathcal{A}$ /P memories into N nodes based upon the (N,k) encoding for MDS.

1. For the fast bandwidth sparse encode procedure, form N encode data sets containing (x,y,s) triplets (see [**2**],[4]).
2. The (N,k) MDS encoding procedure divides a cloud database of size $\mathcal{A}/P$ into k sets of size $\mathcal{A}/(P \times k)$. In each system, the resulting source data is further encoded into N distributed memories via an (N, k) code with an erasure code rate, $r = k/N$. We therefore allocate $P \times N$ memories of size $\mathcal{A}/(P \times N)$ in a Map-Reduce framework. From this construction, we can tolerate $N - k$ disk failures in each of the P sets and still reconstruct any of the k information nodes in each set.

STEP 3: As Poisson errors arrive at the big data memory, For j $= 1,2, \ldots$ P

1. Apply reconstruction adaptively selecting fast bandwidth a probablity q or MDS a probablity $1 - q$ at so the mean computational latency of the reconstruction code is $1\% \times T_{per}$ ($\ll$ latency of MDS).
2. Place the N (x, y, s) encoded data sets in N separate Nodes for each j.

EndFor

We illustrate this cloud application protocol in Figure 2.

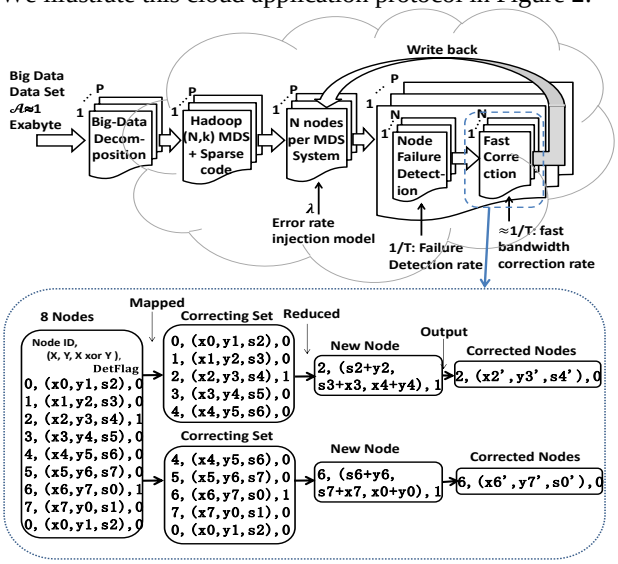

Figure 2: Cloud encode and reconstruction procedure for a given big data size, $\mathcal{A}$, and error arrival rate, λ, into memory.

III. Fast Bandwdith Map-Reduce for Big Data Reconstruction on Clouds

Our goals is fast real time processing of Big-Data with reconstruction speed increased using with Hadoop parallel operation and Fast Bandwidth repare. The Map-reduce pseudo codes is defined as follows:

Map pseudocode:

Map

Define Map(Node_input N, detection_input D):
Correcting Set = [];
for each Node in Node_input N
if detection flag D(i) == 1, **then**
CorrectingSet = [Node((i-2)%N);
Node((i-1)%N); Node(i);
Node((i+1)%N); Node((i+2)%N)];
Error_Location = i;
endif
endfor
return Reduce (Error_Location, CorrectingSet);

Reduce pseudocode

Reduce

Define Reduce(Error_Location **key**, CorrectingSer **value**):
New_node = [];
New_node.x = xor(Node((i-2)%N).s , Node((i-2)%N).y);
New_node.y = xor(Node((i+1)%N).x, Node((i-1)%N).s);
New_node.s = xor(Node((i+2)%N).x , Node((i+1)%N).y);
Node(i) = New_node;
return (Error_Location, New_node)

The Cloud Simulation Results of Fast Bandwidth Reconstruction are illustrated in Figure 3.

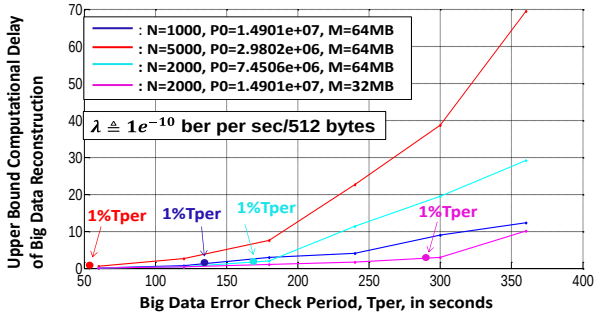

Figure 3: Upper Bound Computational Latency of Big Data Reconstruction Versus Big Data Error Check Period, Tper.

IV. Proposed NSF Comptuer Science Cloud Operation

We propose Hadoop Map-Reduced, Big-Data (1 Exabyte), fast exact repair reconstruction experiments in an NSF cloud that is simultaneous capable of MDS reconstruction.

Selected References

[1] G. Wang and Y. Zhao, "A Fast Algorithm for Data Erasure," ISI 2008 IEEE International Conference on Intelligence and Security Informatics, pp. 245-256, 2008.

[2] D.S. Papailiopoulos, J. Luo, A.G. Dimakis, C. Huang, and J. Li, "Simple Regenerating Codes: Network Coding for Cloud Storage," *The 31st Annual IEEE International Conference on Computer Communications: Mini-Conference,* pp. 2801-2805, 2012.

[3] Mielke, N.; Marquart, T.; Ning Wu; Kessenich, J.; Belgal, H.; Schares, Eric; Trivedi, F.; Goodness, E.; Nevill, L.R., "Bit error rate in NAND Flash memories," Reliability Physics Symposium, 2008. IRPS 2008. IEEE International , vol., no., pp.9,19, April 27 2008-May 1 2008

[4] Papailiopoulos, D.S.; Jianqiang Luo; Dimakis, A.G.; Cheng Huang; Jin Li, "Simple regenerating codes: Network coding for cloud storage," INFOCOM, 2012 Proceedings IEEE , vol., no., pp.2801,2805, 25-30 March 2012