Extended Abstract

Contribution to the Design and Implementation of the Highly Available Scalable Distributed Data Structure LH*$_{RS}$

Rim Moussa

Thesis Goal

This thesis studied the design and implementation of the scalable and distributed data structure (SDDS) termed LH*$_{RS}$. LH*$_{RS}$ distributes scalable data on storage nodes using the distributed linear hashing schema LH*. In addition it tolerates the unavailability of any $k \geq 1$ unavailable nodes. The value of k can be large enough for any practical need. It can also scale transparently with the file, to prevent the reliability decrease, otherwise necessary. The scheme uses the parity calculus, to minimize the storage overhead. The calculus has no impact on the search time. We seek in this context to minimize the time of the update and recovery operations.

Thesis Contribution

Our theoretical results contribute to the definition of a particularly efficient parity calculus for our scheme. It uses a novel parity data calculus by a specific Reed Solomon Code (RS codes). The choice of RS codes results from the fact that are MDS and systematic codes. To tune the access performance in this context, we designed and implemented the prototype LH*$_{RS}$ Manager. Our system allows for the efficient record search and insert operations as well as for the recovery of unavailable data for any practical value of k.

We validated our choices by extensive experimental performance analysis. This was in particular the only way to estimate the actual performance of our scheme, as well as the contribution of various improvements, introduced over the time of our study. The complexity of the underlying implementation prohibited in practice any purely theoretical conclusions.

To obtain our results, we carried out throughout the thesis, cycles of design, implementation and experimental validation, as we detail in what follows. Our constant purpose was best access and recovery performance. The results contributed, on the one hand, to the evolution of the parity calculus for the scheme. The final parity matrix proposed for LH*$_{RS}$ has 1st column and 1st row of '1's. This leads to more frequent use of XORing than in the initial matrix of LH*$_{RS}$, [LS00]. In particular, for $k = 1$, our schema performs XOR encoding/decoding, and behaves as the most widely used RAID systems. We have further shown the utility of using the logarithmic matrix derived from the above one (for both encoding and decoding). Finally, while the initial scheme was foreseen for the Galois Field GF(2^8), we have shown that GF(2^{16}) should be used instead, as more efficient in our case.

We have further contributed progressively to the optimized design of main architectural components of our prototype. In particular, we progressively improved our network component. We provided it with a more efficient TCP/IP connections handler, where TCP/IP connections are passive OPEN [ISI81, MB00]. We have also designed the flow control and acknowledgements management strategy based on the principle of message conservation until delivery [JK88, GRS97, D01]. Finally, we added a dynamic addressing structure, updated through multicast probe for new data/parity servers.

As we already mentioned, we used the experimental performance analysis to prove the validity of our choices. We measured every improvement to find out its relative and absolute incidence on the response time. The final results show attractive access and recovery performance. Our testbed files with 125K records, recovered in almost half a second from a single unavailability and in about 1.2 seconds from a triple one. Our prototype recovers a data bucket group of size $m = 4$ from 1-unavailability at the rate (speed) of 5.66 MB/s of data. Next, we recover 2 data buckets of the group at the rate of 7.14 MB/s. Finally, we recover the group from 3-unavailability at the rate of 7.89 MBs. Individual search, insert and update times were at most 0.5 msec for a 3-available file. The performance is also due to the data storage in the distributed RAM. These results prove the efficiency of our scheme.

Future work should focus on the investigation of efficient erasure-resilient codes. It should also be possible to improve our server speed by using communication APIs calls in the background (by using I/O completion ports). We also foresee the study of applications of our scheme, to a high-availability DBMS design especially.

Publications

We have published four papers about our work [ML02, MS04, LMS04, LMS04*b*]. The latter publication, VLDB-04, corresponds also to the demo of our system.

In [ML02], we report the design and performance results of our first LH*$_{RS}$ prototype. The prototype implementation extends the one of [L00, M00], but improves the performances, essentially by replacing UDP by TCP/IP for parity update propagation during server split, parity bucket creation and bucket recovery. In this paper the servers' architecture is inherited from the work of [B00, D01], and basic encoding/decoding routines using Reed Solomon codes without any optimisations.

In [MS04], we describe the three components embedded to the SDDS2000 architecture. We report also performance results related to parity overhead, data recovery (server and record).

[LMS04] is a demo paper, where we summarize our LH*$_{RS}$ best settings, describing at a glance our RS encoding/decoding scheme as well as our server architecture. The demo outline shows the main functionalities of our prototype.

Finally, in [LMS04-*b*], we detail LH*$_{RS}$ fundamental theory with the new parity matrix (column and line of '1's). We highlight our encoding/decoding optimizations and report performance results. We also discuss the related work. This paper is submitted for a journal publication

Thesis Outline

The Thesis dissertation is divided into two parts, plus appendixes. The first part, *Etat de l'Art*, covers the state of the art of the networked data storage systems, and mechanisms of high availability. It consists of two chapters. The second part, *Conception & Implantation de LH*$_{RS}$*, describes LH*$_{RS}$ schema and our LH*$_{RS}$ manager as well as performance results.

The first Chapter of the Thesis, "*Introduction* ", discusses the basic issues and the motivation for the Thesis work, as well as our contribution.

In Chapter 2, "*Introduction aux Systèmes Répartis* ", we cover the basic features of networked data storage systems: hardware configuration, data distribution schemes, server architectural design, communication protocols, evaluation metrics and requirements. We also cover with special interest Scalable and Distributed Data Structures (SDDS), being one of our thesis fundamentals.

Chapter 3, "*Revue de Mécanismes de Haute Disponibilité* " reviews literature on high availability. It sketches recovery techniques from media failures [PGK88, HGK+94, B3M95, XB99, SS96, SS02, CEG+04, LMR98]. We explore and highlight in turn the pros and cons of each of these techniques.

Chapter 4, "*Fondements Théoriques de LH*$_{RS}$", details Reed Solomon codes, LH*$_{RS}$ schema, gives some hints to improve coding and decoding, and compares our adaptation of Reed Solomon codes to related work [R89, W91, SB92, BM93, WB94, BKL+95, SB95, R96, R97, P97, ABC97, IMT03, MTS04].

Chapter 5, “*Le Gestionnaire LH*$_{RS}$*”, describes the three components embedded to SDDS2000 server architecture, namely the TCP/IP connection handler [ISI81, MB00], the message acknowledgement strategy [GRS97, D01] and the dynamic addressing structure.

Chapter 6, “*Architecture opérationnelle du Gestionnaire LH*$_{RS}$*”, describes different scenarios, especially client manipulation (insert, update, delete, search queries), update parity propagation during the server split, parity server creation and servers recovery. We also discuss how the scenarios are mapped to the different architectures.

Chapter 7, “*Mesures de Performances de LH*$_{RS}$*”, reports the results of the conducted experiments in two different hardware configurations: a set of five 733MHz machines connected to a 100Mbps Ethernet network and a set of five 1.8GHz machines connected to 1Gbps Ethernet network. The performance factors of interest are the insert record time in a *k*-available LH*$_{RS}$ scheme, the record search time in both normal mode and degraded mode, the parity bucket creation time and *f* buckets’ recovery time. We compare performance results obtained in the two different configurations, SDDS2000 server architecture to our enhanced server architecture, and RS/ XOR coding/decoding in the two Galois Fields GF(2^8) and GF(2^{16}).

Finally, the last chapter “*Conclusion & Travaux Futurs*”, concludes the thesis dissertation and gives some future research directions.

The Thesis contains also *Appendix A & B* that present our prototype. We describe there the main functions offered by our client interface. We also describe the installation requirements, the structure of our object code for future extensions.