Main memory database recovery strategies

Arlino Magalhães[1], Angelo Brayner[2], José Maria Monteiro[2]

[1]Information Systems Course
Universidade Federal do Piauí (UFPI)
Av. Universitária, s/n – Teresina – PI – Brasil

[2]Computing Department
Universidade Federal do Ceará (UFC)
Av. Mister Hull, s/n – Fortaleza – CE – Brasil

arlino@ufpi.edu.br, {brayner, monteiro}@dc.ufc.br

Abstract. *Main Memory Database (MMDB) technology has been an efficient alternative for a real-time and high-performance mission-critical applications. However, MMDBs are more vulnerable to failures due the memory volatility. Although the recovery component looks similar in disk- and memory-oriented systems, these systems differ drastically in how they implement their architectural components. This tutorial aims to provide a thorough review of MMDBs recovery techniques. To achieve this goal, the tutorial reviews the main concepts of database recovery and MMDB implementations. Only then, we present the MMDB recovery techniques and discuss the recovery strategies of a representative sample of modern MMDBs.*

Resumo. *A tecnologia bancos de dados em memória tem sido uma alternativa eficiente para aplicações de missão crítica em tempo real e de alto desempenho. No entanto, os bancos de dados em memória são mais vulneráveis a falhas devido à volatilidade da memória principal. Embora o componente de recuperação pareça semelhante nos bancos de dados em disco e em memória, esses sistemas diferem drasticamente em como eles implementam seus componentes arquiteturais. Este tutorial tem como objetivo fornecer uma revisão completa das técnicas de recuperação de bancos de dados em memória. Para atingir esse objetivo, o tutorial revisa os principais conceitos de recuperação de banco de dados e implementações de bancos de dados em memória. Só então, serão apresentadas as técnicas recuperação de bancos de dados em memória e discutidas as estratégias de recuperação de uma amostra representativa dos bancos de dados em memória modernos.*

1. Tutorial identification

1. **Title**: Main memory database recovery strategies
2. **Authors**:
   - Angelo Roncalli de Alencar Brayner (presenter)
     Federal University of Ceara
     brayner@dc.ufc.br
   - José Maria da Silva Monteiro Filho
     Federal University of Ceara
     monteiro@dc.ufc.br

- Arlino Henrique Magalhães de Araújo (presenter)
  Federal University of Piaui
  arlino@ufpi.edu.br

3. **Type**: advanced.
4. **Presentation language**: portuguese.
5. **Audio-visual equipment required**: a notebook and datashow.

2. Tutorial overview

Most of the current application scenarios, such as trading, real-time bidding, advertising, weather forecasting, social gaming, etc., require massive real-time data processing. Main Memory Database technology has proved to be an efficient alternative for a real-time view of operational data (OLTP-style) and high-performance mission-critical situations due to the high throughput rates and low latency provided by these systems. This is because, in such systems, the database resides in Random Access Memory (RAM). Additionally, the development of new memory technologies has provided a larger storage capacity with lower costs. To illustrate our claim, memory storage capacity and bandwidth are growing at a rate of 100% every three years. In the meanwhile, RAM costs are falling by a factor of 10 every five years. Moreover, other recent hardware/architecture improvements can potentially provide better performance to MMDBs with low overhead [Magalhães et al. 2021b, Arulraj and Pavlo 2017].

The fact that the database resides in volatile storage influences the design approaches adopted by MMDBs, such as query processing, concurrency control, recovery after crashes, data storage, and indexing. For this reason, these systems are designed to optimize access to main memory instead of secondary memory, as with traditional disk-resident systems [Zhang et al. 2015, Faerber et al. 2017].

MMDBs provide very high IOPS (Input/Output Operations Per Second) given that the primary database is handled in volatile storage. However, the database residing in a volatile memory makes these systems much more sensitive to system failures than conventional disk-resident database systems. The recovery mechanism is responsible for restoring the database to the most recent consistent state before a system failure has occurred. In this way, after a system crash, the recovery manager loads the last valid checkpoint (a prior database backup copy) and then starts to execute all actions recorded in the log file forward from the checkpoint record [Malviya et al. 2014, Mohan et al. 1992, Härder and Reuter 1983].

MMDBs are implemented in a variety of ways. Therefore, recovery strategies should follow the architectural choice of the database system. This tutorial intends to present the several recovery strategies used in modern MMDBs. These strategies are a representative sample of the different MMDB recovery implementations. In order to achieve this goal, firstly, the tutorial reviews the main concepts of database recovery and MMDB implementations. This tutorial focuses on relational MMDB recovery. However, the recovery strategies presented are implemented similarly in other types of MMDBs. Besides, the tutorial introduces modern recovery techniques, such as instant recovery [Magalhães et al. 2021b, Magalhães 2021, Magalhães et al. 2021a, Sauer 2019].

2.1. Justification

Despite the efforts and publications, database recovery is not so well understood by the research community. In-memory database systems are not taught sufficiently in database courses or discussed enough in database textbooks. This tutorial aims to elucidate the main issues related to database recovery, especially those related to MMDB recovery.

2.2. Target audience

The target audience of this tutorial is students and professionals/developers who use traditional databases and seek alternatives for better performance through MMDBs. Additionally, this tutorial is also intended for academics interested in learning more about the main memory database recovery research area. The participants need to have an understanding of databases.

2.3. Related recent tutorials

The tutorials "*Modern main-memory database systems*" (VLDB 2016) and "*In-Memory Analytic DBMSs: Design and Lessons Learned*" (SBBD 2018) provided an overview of recent developments in MMDBs. The tutorials covered the key issues and architectural choices that must be resolved when building an MMDB.

In addition to presenting the architectural choices and implementation techniques for MMDBs, the tutorial proposed here intends to discuss the main concepts of traditional database recovery (implemented by disk-resident systems), detail how MMDBs implement persistence and recovery after failures and then present the main recovery strategies implemented by modern MMDBs. Besides, the tutorial covers modern recovery techniques, such as instant recovery.

2.4. Scope and structure

In order to achieve the goals mentioned in Sections 2.1 and 2.3, this tutorial intends to discuss the following topics:

1. **Introduction and motivation**: This section briefly introduces the main concepts of in-memory systems and does motivations to adopt MMDBs. The section shows the need to adopt MMDBs to support ultra-low latency service and real-time data analytics which is impossible to achieve using traditional disk-based processing/storage systems.
   This section presents some factors that make viable to build in-memory systems where a significant part, if not the entirety, of the database fits in memory. For example, memory storage capacity and bandwidth have been doubling roughly every three years, while its price has been dropping by a factor of 10 every five years. In addition, this section introduces some basic aspects of how MMDBs provide persistence and crash recovery.
2. **Database recovery**: This section aims to present the main concepts of DBMS recovery focusing on disk-resident systems. The section briefly reviews ACID properties and some buffer replacement protocols, features of database crashes, recovery methods, and our main section discussion, the ARIES recovery, a quite popular algorithm for recovering traditional databases.
   Furthermore, this section discusses some variant ARIES algorithms, modern recovery techniques, and high availability techniques. The main objective of this section is to provide a sufficient theoretical basis for the next sections.

3. **Main memory databases overview**: This section discusses the architectural choices and implementation techniques for MMDBs. MMDBs avoid the traditional design of disk-resident databases for performance reasons. The design approaches for MMDBs are diverse. The fact that the database resides in volatile storage influences the design approaches adopted by MMDBs, such as query processing, concurrency control, recovery after crashes, data storage, and indexing.
Besides, this section discusses some technologies that boosted the development of MMDBs. The emergence of new technologies boosted interest in MMDBs. Hardware/architecture solutions have been increasingly exploited for performance gain. These improvements offer promising alternatives for in-memory systems to reach their full potential. This section briefly presents the core technologies that leverage MMDBs, such as NUMA architecture, SIMD instructions, RDMA networking, hardware transactional memory, and non-volatile memory.
4. **Main memory databases recovery**: Recovery activities are the only way to recover an MMDB to the last consistent state before a crash. Systems can keep database copies for higher availability. However, high availability infrastructures are not immune to some sources of failures and can lead to a significant cost to the database infrastructure. Thus, recovery techniques are necessary to avoid failures and repair failed systems as quickly as possible.
In general, MMDB's durability and recovery seem like disk-based database systems. However, these systems are very different in several details. For example, ARIES-style recovery protocols are avoided in MMDBs for performance reasons. This section highlights how logging, checkpoint, and restart (recovery) processes are handled on MMDBs.
5. **Main memory databases recovery strategies**: This section describes the main features of recovery mechanisms delivered by well-known MMDBs, such as Hekaton, VoltDB, HyPer, SAP HANA, SiloR, TimesTen, PACMAN, Adaptive Logging, and FineLine. These strategies are a representative sample of the different MMDB recovery implementations.
6. **Main challenges and future directions**: This section intends to discuss some aspects related to challenges and future directions of research in MMDBs in order to provide guidance for other researchers.

3. Brief professional biographies of authors

3.1. Angelo Brayner

Angelo Brayner received the MSc degree in Computer Science from the State University of Campinas (UNICAMP), Brazil, in 1994. In 1999 he receivedhis Ph.D. degree from the University of Kaiserslautern, Germany, working in the field of Transaction Management in Multidatabase Systems. He has been with the Federal University of Ceara, Brazil, since 2001 as full professor and leader of the CEARA (advanCEd dAtabase Research) research group. His current research interests include high-performance transaction systems, main-memory databases, and query processing in wireless sensor networks and mobile databases.

3.2. José Maria Monteiro

José Maria Monteiro received the MSc degree in Computer Science from the Federal University of Ceará (UFC), Brazil, in 2001. In 2008 he received his Ph.D. degree from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil, working in the field of Self-managed and Autonomic Databases. He has been with the Federal University of Ceará (UFC), Brazil, since 2010 as full professor and researcher at ARIDA (Advanced Research In DAtabase) research group. He has published more than 50 papers in international journals and conference proceedings and has coordinated several research and development projects. His current research interests include data science, big data and misinformation. José Maria has served as the Computer Science Department's Chair (2012-2015).

3.3. Arlino Magalhães

Arlino Magalhães has a Ph.D. degree in computer science from the Federal University of Ceara (2022). His master's degree was also taken at the Federal University of Ceara (2013). He graduated in computer science from the Federal University of Piaui (2004). Currently, he works as a professor in the Information Systems Course at the Federal University of Piaui. During his doctoral studies, he published some relevant articles in MMDB recovery research area, such as [Magalhães et al. 2021b], [Magalhães 2021], and [Magalhães et al. 2021a]. He has areas of interest in database and software engineering, working mainly on the following themes: self-tuning databases, cloud databases, and in-memory databases.

References

Arulraj, J. and Pavlo, A. (2017). How to build a non-volatile memory database management system. In Salihoglu, S., Zhou, W., Chirkova, R., Yang, J., and Suciu, D., editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1753–1758. ACM.

Faerber, F., Kemper, A., Larson, P., Levandoski, J. J., Neumann, T., and Pavlo, A. (2017). Main memory database systems. *Foundations and Trends in Databases*, 8(1-2):1–130.

Härder, T. and Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys (CSUR)*, 15(4):287–317.

Magalhães, A. (2021). Main memory databases instant recovery. In Bernstein, P. A. and Rabl, T., editors, *Proceedings of the VLDB 2021 PhD Workshop co-located with the 47th International Conference on Very Large Databases (VLDB 2021), Copenhagen, Denmark, August 16, 2021*, volume 2971 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Magalhães, A., Brayner, A., Monteiro, J. M., and Moraes, G. (2021a). Indexed log file: Towards main memory database instant recovery. In Velegrakis, Y., Zeinalipour-Yazti, D., Chrysanthis, P. K., and Guerra, F., editors, *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*, pages 355–360. OpenProceedings.org.

Magalhães, A., Monteiro, J. M., and Brayner, A. (2021b). Main memory database recovery: A survey. *ACM Comput. Surv.*, 54(2):46:1–46:36.

Malviya, N., Weisberg, A., Madden, S., and Stonebraker, M. (2014). Rethinking main memory OLTP recovery. In Cruz, I. F., Ferrari, E., Tao, Y., Bertino, E., and Trajcevski, G., editors, *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 604–615. IEEE Computer Society.

Mohan, C., Haderle, D., Lindsay, B. G., Pirahesh, H., and Schwarz, P. M. (1992). ARIES: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems (TODS)*, 17(1):94–162.

Sauer, C. (2019). Modern techniques for transaction-oriented database recovery. In Grust, T., Naumann, F., Böhm, A., Lehner, W., Härder, T., Rahm, E., Heuer, A., Klettke, M., and Meyer, H., editors, *Datenbanksysteme für Business, Technologie und Web (BTW 2019), 18. Fachtagung des GI-Fachbereichs ,,Datenbanken und Informationssysteme" (DBIS), 4.-8. März 2019, Rostock, Germany, Proceedings*, volume P-289 of *LNI*, pages 487–496. Gesellschaft für Informatik, Bonn.

Zhang, H., Chen, G., Ooi, B. C., Tan, K., and Zhang, M. (2015). In-memory big data management and processing: A survey. *IEEE Transactions on Knowledge and Data*, 27(7):1920–1948.