Cloud Computing in the 1970s: The Discovery of Hash Based Relational Algebra

Kjell Bratbergsengen

Cloud Computing in the 1970s: The Discovery of Hash Based Relational Algebra

Kjell Bratbergsengen

Department of Computer and Information Science (IDI)
The Norwegian University of Science and Technology (NUST/NTNU)
kjellb@idi.ntnu.no

Abstract. Cloud computing was not a known term in the 1970s. However, this is about work done from 1975 to 1990, which with today's terminology partly could have been termed, cloud computing. This article is about how we discovered hash-based methods for doing relational algebra, searching for efficient algorithms to run on a future parallel computer. We found the algorithms; however, Norsk Data never built the parallel computer. We built four parallel computers ourselves, but first we implemented the hash based relational algebra algorithms in TechRa, a system for mono computers, and obtained excellent results. Being able to establish a lab, developing machines, software and algorithms led to a very rich research period from 1985 to 1990, involving many PhD and master students. We had the world record in both sorting and relational algebra on our parallel computers for a while. We commercialized our research in 1989 with many positive effects, but with negative effects to the university activities in the field.

Keywords: Database research, history of research projects, parallel computing

1 The Database Group

A small database group was formed at NTH and SINTEF in 1972 lead by Kjell Bratbergsengen, then assistant professor at NTH. The group built several file and database systems right from its beginning: AKUFIL [1], INDSEQ [2], Ra1 [3], Ra2 [4], and a backend database machine Ra3 [5]. AKUFIL was a specialized file system for storing matrices. INDSEQ was an indexed sequential file system used in an internal project CASCADE led by Arne Sølvberg. Inspired by a similar system on our UNIVAC 1107 computer, we used block splitting on overflow (at both data and index levels); our architecture was really close to what is known as B*-trees today. RA1 was used to store the first BIBSYS database and it served BIBSYS for more than fifteen years until it was followed by ADABAS on a new IBM computer. RA2 was developed for Kongsberg Vaapenfabrik (KV); the first application was their bill of material database. Later, KV in Scandinavia also marketed RA2. RA2 featured sequential, indexed sequential and hashed access to records at the first level. Below each first level record, there could be a general tree of records of different types.

The database group was very familiar with the CODASYL work. It existed a Norwegian system SIBAS, based on the CODASYL proposed standard. SI developed

SIBAS and Norsk Data marketed it; there was no room in Norway for more systems of this type. Our market was lighter and faster systems.

The database group started as a common group for NTH and SINTEF. Bratbergsengen was employed by the university, the first two members; Knut Hofstad and Kjell Wibe were both employed by SINTEF. They operated very integrated the first ten years, however, as the SINTEF group grew and financing needs alike, they became more independent of each other.

2 The ASTRA Project

In 1970, Edgar F. Codd published his seminal paper "A Relational Model of Data for Large Shared Data Banks." This was a totally new way of organizing data, but it was regarded as useless caused by very long execution times. The model favored programmer productivity. Finding methods to get reasonable execution times seemed impossible, and became a great challenge. Relational algebra is the "instruction set" of relational database systems. Instructions included selection, projection, join, intersection, union, difference, division and aggregation. Operands were tables or files if you like, so are also results. Algorithms for doing relational algebra were normally nested loop algorithms or sort based. These methods were terrible for large operands. A number of research projects tried to find effective ways to implement the relational model the next fifteen years [6–12]. Because of the time-consuming algorithms, all the listed projects involved building some type of new hardware, either parallel disks or machines with smart or especially associative memory. The list of projects is not complete.

Norsk Data saw a market for selling more hardware, and together we worked out a research proposal to the Norwegian Science Foundation – the ASTRA project in 1977. We should develop a relational database system for parallel computers. The database data was to be stored in main memory. Norsk Data engineers had figured out that main memory would reach the same price as disk in quite a few years. Main memory storage and parallel machines should solve the problems with long execution times. However, Norsk Data engineers were not equally acquainted with disk technology and disks became cheaper too. Eventually this led to the end of the project. However, the main result of the project was the discovery of efficient methods for doing relational algebra. It is easy to parallelize selection. All the other relational algebra operations are all based on finding records with matching values of their operation keys – a set of attributes specified for the operation. In the seventies, nested loop or sorting for larger operands was used to find records with equal key values. Both methods are hard to parallelize.

2.1 Hash Based Relational Algebra Algorithms

The basic idea behind hash based relational algebra methods is quite simple. If two records have the same key value, their hash values would also be the same. If two hash values are different, their keys must be different. If two hash values are equal,

their keys *might* be equal, but not necessarily. To avoid testing too many unequal keys for equality, the hash value space is adapted so the probability of synonyms is small. Synonyms: two different key values giving the same hash value. We also use a hash formula on the primary key to compute the home node (the node where it is stored) of a record. Only one node is activated for storing, deleting, or modifying a record given that the primary key is known.

The first phase of all relational algebra operations (except selection and non-equijoin) is a redistribution of all the operand records to the target node determined by the hash value of the operation key. Then the operation is completed on the target node; the global task is split into N local tasks. N is the number of nodes on the parallel computer. The method requires a minimum of internal synchronization and features a built in load distribution mechanism, i.e. the hash based redistribution of operands.

Redistribution of operands required an internode network with high capacity and good scalability. The analysis of different networks was documented in [13]. We looked at different ring structures, records might move one or both ways, and the rings were intersected to reduce the average number of hops during redistribution. We ended up with an extreme variant, a ring of only two nodes, able to handle many nodes; the number of intersecting dimensions became larger. This is how we discovered the hypercube. The hypercube was chosen because it scaled perfectly and it had simple routing rules. The number of nodes in a hypercube is $N=2^D$ where D is the number of dimensions. We could also call D the diameter of the hypercube, which is the maximum number of hops to relocate a record. The average number of hops is D/2. In [13], we also described hash based relational algebra in very vague ways; we protected pour methods! We discussed patenting, but had no experience, no one to ask for advice, and no resources for financing.

The ASTRA project was quite ambitious; it also had a language development goal. The relational model was integrated into a modified SIMULA, the new programming language was named ASTRAL, see [14] and [15]. The database – or permanent data was declared in the outer block of the program. This block was never left conceptually. New programs were inserted as procedures into the persistent outer block. We managed to write a compiler for ASTRAL before the project was called off. Major contributors were Tore Amble and Tor Stålhane.

3 After ASTRA Came TechRa

As it became evident that Norsk Data would terminate the project because the assumptions about main memory price versus disk price did not hold, we realized that we had to build the parallel hardware ourselves.

However, this author got a request from a KV subsidiary – SYSSCAN Mapping, they needed at database system for storing geographic data. SYSSCAN Mapping pioneered digital maps. Kvatro AS (KV avdeling Trondheim) got a contract for building a brand new relational database system, with some added features for storing free text and especially sequences of x, y, z coordinates. The relational algebra was of course realized using hash-based methods. Roy Lyseng, fresh from NTH, was

employed by Kvatro to lead the project, which was mainly staffed by RUNIT people. Kvatro marketed TechRa successfully for more than twenty years. It was a lightweight and very efficient database system. The usefulness of hash based relational algebra methods on a mono computer system had been evidenced.

4 The Period of the Hypercube Laboratory (HCL)

How the methods worked on a real parallel computer had still to be tested. Realizing hash based relational algebra on a parallel computer moved the possible bottleneck from disk I/O or bus/memory system to the network connecting the nodes of the parallel machine. For some years, the author worked on a VLSI chip for setting up lines through the hypercube. However, no serious attempt to produce it was made, it simply was too expensive.

In 1983, we came across dual ported memory chips. Could we use them for communication between computers or nodes in a parallel computer? A test set up with three PCs was built during the Christmas holidays in 1983, and it worked. Dual port RAM was less efficient than using line switching between nodes; however, it was considered efficient enough for a demonstration system. After getting NFR support, a project was set up to build a parallel computer with eight nodes to demonstrate parallel algebra. David DeWitt and his group at University of Wisconsin had worked out a benchmark for relational database systems in 1983 [16]. The most popular test was the DeWitt join test, joining two tables; Table A containing 1,000 records each 182 bytes long and table B containing 10,000 records also 182 bytes long. We "promised" NFR to do the test in less than 3.0 seconds. We did it in 1.8 seconds. The same test was run on several computers: TechRa on ND540 used 17 seconds, 32 seconds on a VAX 11/750. Ingres – one of the leading systems at the time – used 156 seconds on a VAX 11/750 [17]. CROSS8 was the name of this first operational parallel computer. It was finished in 1985. It had a direct connection – one common dual port RAM of 1,024 bytes – between each pair of nodes, which explains its name. Each node was a single board computer with a SCSI disk. Each node had also a DPRAM in common with a supervising PC, which controlled the whole system. We experienced DPRAM to a very flexible and useful tool for developing new hard- and software. A curiosity: During start up, DPRAM was mapped in as boot memory, after start up the same area was used for transferring messages.

We built three more parallel computers: HC16-186, HC16-386 and HC64-486 all with hypercube topology DPRAM message passing network. The first number (16, 64) is the number of nodes, and the last number denotes the Intel processor used. Each node was a single board computer designed and built by us. HC64-486 was never completed. It was up and running with four nodes; however, the HCL then closed down.

Shifting from direct communication in CROSS8 to indirect communication in the HCxx-systems was an issue. The CPU had to use some time to move records in transit between nodes. However, it turned out not to be a problem. HC16-186 and HC16-386 were fairly stable systems and they were used to improve the database algorithms, and a large number of parallel algorithms were developed and tested by our PhD and

master students. We had the world record in sorting for a period; in 1989, 100 MB was sorted in 180 seconds on HC16-186 [18].

In 1989, we were ready to commercialize our research, and we established the company HypRa AS with seven shareholders from the research group. The new company should develop a database for Telenor. It soon became clear that Telenor wanted a transaction-oriented system with extremely high reliability. Parallel algebra systems – which was really ready to go commercially at that time; it was not in Telenor's interest. The commercialization became a catastrophe to database research at the department. The lab disappeared, so did the people and there was no money to reestablish the loss. Despite being the founder and board chair of the company, I could not go along with it and I left HypRa in 1991. The high throughput, extremely reliable, transactional database for telecom operations required another ten years of research, but that is another (success) story. HypRa AS went through several transformations, Teleserve, Clustra, Sun Microsystems Trondheim, and was recently bought by Oracle. The database activities produced a large number of master candidates, very competent in database technology and parallel computing which strengthened Trondheim as the database city.

5 What We Achieved and Learnt

We discovered and demonstrated the usefulness of hash-based algorithms for doing relational algebra. These methods are part of all major data base systems today, both for mono machines and parallel machines. Our early experiences with parallel machines were taken further within HypRa and the companies that followed from HypRa; developing high capacity and highly reliable database systems.

The work on hypercube networks or more generally, high capacity highly scalable networks had not been continued after the three last prototypes. The reason: We had not pursued this track and other off the shelf hardware had been good enough. Work on hypercube networks or scalable networks could be reconsidered as we now have computers with thousands of nodes.

We established a permanent commercial activity on core database technology in Trondheim and Norway. We should have done more on publishing our work, especially the first years. That would have improved our worldwide standing, although we do not complain.

To protect our results we badly needed assistance. Today the universities in Norway have systems for this. However, it is surprising how long it takes to develop ideas before they are put to work. We discovered hash-based algebra in 1978, a vague publication was done in 1980, and a full publication of the methods and their results demonstrating their superiority came in 1984 [19]. Ten years later, at SIGMOD 1994, all the major database vendors of that time (IBM DB2, Sybase, and Oracle) indicated that they are now working on implementing hash-based methods in their systems.

Commercialization is very risky. It took away people, the lab, and financing. The database group was never reestablished to the same level. Partners from industry and research institutes had only an eye for protection of their commercial interests and they took no responsibility for continued research and production of more candidates

at the university. Continued activity at the department was considered as a threatening competition to the commercial offspring.

During the five intensive years at the Hypercube laboratory, we produced a large number of master candidates with hands on experience from data management and parallel computers. We covered a broad field from VLSI design to parallel programming and algorithms and advanced database technology: parallel core database algorithms, transaction handling and fault tolerance techniques on a parallel system. Access to bright candidates with relevant education was cited as a major reason for international companies to establish branch offices in Trondheim (Yahoo, Google).

Instead of moving projects away from the university, it would have been much better for continuity in research to keep the commercial activities within the department, or at least central parts of it. The same experience repeated itself when the FAST group moved away from our department.

Having a common goal, a laboratory and being able to build prototypes is of highest value. It requires financing to keep laboratory personnel and develop new hardware. The lab is bringing people together, a common working and *social* arena for graduate and undergraduate students, researchers and professors. Problems, hard to find from theory alone, popped up such as deadlock in the record redistribution algorithm. Crossing hypercube dimensions in arbitrary order caused deadlock. Crossing hypercube dimensions in any fixed order (all records obey this order) avoids deadlock. Of course, this is what we learnt in operating systems and transaction scheduling theory! In parallel systems, we must handle failures, the more nodes, the higher probability that one of them will fail. Working with cloud technology in the 1970s and 1980s was a very successful educational project. Many of the PhDs and masters from the project have key positions in advanced IT projects today.

Acknowledgments. Torgrim Gjelsvik was the first researcher at HCL. He did most of the practical work building new machines and writing the low-level software. Ole John Aske was the second researcher at HCL and he worked mainly with developing software. They made everything work and helped. PhD students Øystein Torbjørnsen, Eirik Knutsen, Petter Moe, and Olav Sandstå were also part of the HCL kernel. A large number of students did their project works and master thesis within the lab. Some developed the system further and many used the hypercube computers to get their first experience with writing and implementing parallel algorithms. I am very grateful to all of you, the ASTRA and TechRa projects participants and members of the database group. Thanks also to the Norwegian Science Foundation who financed both the ASTRA and the Hypercube projects.

References

1. Bratbergsengen K., Hofstad K., Nødtvedt E.: AKUFIL – et filsystem for lagring og gjenfinning av matriser. Brukerveiledning, Oppdrag nr. 140404.0 SINTEF 15.11.1972
2. Bratbergsengen K., Johansen T.: INDSEQ Et indeks-sekvensielt filsystem for Univac EXEC-8 og IBM System 360/370 OS, CASCADE arbeidsnotat nr. 23. 24. 4. 1972

3. Bratbergsengen K., Hofstad K.: RA1 – et vertsspråkbasert databasesystem. Metode og systemdokumentasjon, SINTEF prosjekt nr 140413, 1. 12. 1973
4. Bratbergsengen, K., Kongshaug, P., Kvitsand, S.: RA2 – Brukerveiledning, SINTEF prosjekt nr. 142515 13.1. 1977
5. Bratbergsengen, K., Hofstad, K., Wibe, K., Midtlyng, J.O.: Databasemaskiner – en riktigere måte for å omsette databaseprogramvare, Data no. 3 1977 og NordDATA 77, juni 1977
6. Ozkarahan, E.A., Schuster, S.A., Smith, K.C.: RAP An Associative Processor for Data Base Management, NCC 1975
7. Su, S.Y.V., Lipovski, G.J.: CASSM –A Cellular System for Very Large Databases, VLDB 1975
8. Leilich, H.O., Stiege, G., Zeidler, H.C.: A Search Processor for Data Base Management Systems, VLDB 1978
9. Langdon, G.G.: A Note on Associative Processors for Data Management, ACM TODS Vol 3, No 2 1978
10. DeWitt, D.J.: DIRECT – A Multiprocessor Organization for Supporting Relational Database Management Systems", IEEE Transaction on Computers, Vol C-28 No 6, 1979
11. Maller, V.A.J.: The Content Addressable File Store – CAFS, ICL Technical Journal Vol 1, No 3, 1979
12. Oliver, E.J.: RELACS An Associative Computer Architecture to Support a Relational Data Model, PhD Dissertation, Syracuse University 1979
13. Bratbergsengen, K., Larsen, R., Risnes, O., Aandalen, T.: A Neighbor Connected Processor Network for Performing Relational Algbra Operations, Paper presented at 5th Workshop on Computer Architecture for Non-Numeric Processing, March 11-14, 1980 Pacific Grove, California
14. Amble, T., Bratbergsengen, K., Risnes, O.: ASTRAL: A Structured and Unified Approach to Database Design and Manipulation, Proc. Database Architecture Conference, Venice, 1978
15. Bratbergsengen, K., Stålhane, T.: A feature analysis of ASTRAL, in Michael Brodie and Joachim Smidt The Relational Task Group - Feature Analysis of Relational Database Systems. Springer Verlag 1981
16. Bitton, D., DeWitt, D.J., Turbyfill, C.: Benchmarking Database Systems. A Systematic Approach, VLDB 1983
17. Bratbergsengen, K.: Algebra Operations on a Parallel Computer - Performance Evaluation, The 5th International Workshop on Database Machines, Oct. 5-8 1987 Karuizawa Japan
18. Baugstø, B.A.W., Greipsland, J. F.: Parallel Sorting Methods for Large Data Volumes on a Hypercube Database Computer, Lecture Notes In Computer Science; Vol. 368 Proceedings of the 6th International Workshop on Database Machines, France June 19-23, 1989
19. Bratbergsengen, K.: Hashing Methods and Relational Algebra Operations, VLDB 1984