INTRODUCTION

Recent years have witnessed an increasing trend of the implementation of Distributed Database Management Systems (DDBMS) for more effective access to information. An important quality of these systems, consisting of n servers loosely connected via a communication network, is the ability to adjust to changes in workloads. To service increases in demand, for example, additional servers may be added to the existing distributed system and new data allocations computed. Conventionally, this requires a system shutdown and an exhaustive data reallocation. Such static methods are not practical for most organizations, for these methods result in high costs and in periods of data unavailability.

In a distributed database system, an increase in workload typically necessitates the installation of additional database servers followed by the implementation of expensive data reorganization strategies. We present the Partial REALLOCATE and Full REALLOCATE heuristics for efficient data reallocation. Complexity is controlled and cost minimized by allowing only incremental introduction of servers into the distributed database system. Using first simple examples and then, a simulator, our framework for incremental growth and data reallocation in distributed database systems is shown to produce near optimal solutions when compared with exhaustive methods.

Chapter IX

Algorithm Development, Simulation Analysis and Parametric Studies For Data Allocation in Distributed Database Systems

Amita Goyal Chin
Virginia Commonwealth University, USA
We present the incremental growth framework to address incremental expansion of
distributed database systems. Data is reallocated using one of two data reallocation
heuristics - Partial REALLOCATE or Full REALLOCATE. Both heuristics are greedy, hill-
climbing algorithms that compute new data allocations based on the specified optimization
parameter of the objective cost function. Due to their linear complexity, both heuristics can
be used to solve both small and large, complex problems, based on organizational needs. The
robustness of the heuristics is demonstrated first by simple, illustrative examples and then
by parametric studies performed using the SimDDBMS simulator.

The REALLOCATE algorithms in conjunction with SimDDBMS can be used to
answer many practical questions in distributed database systems. For example, in order to
improve system response time, a database administrator (DBA) may use SimDDBMS for
parametric evaluation. For example, the DBA may analyze the effect of upgrading CPU
processing capability, increasing network transfer speed, or adding additional servers into
the distributed database system. Furthermore, SimDDBMS may easily be modified to
evaluate heterogeneous servers, with different CPU processing capabilities. A DBA may
also use SimDDBMS to determine the impact and cost-benefit analysis of adding some
number, $s \geq 1$, of additional servers at one time.

RELATED WORK

The allocation of data amongst the servers of a distributed database system has
been widely studied as the “data allocation” or “file allocation” problem. The data
allocation problem generally attempts to strategically place relations at servers in the
distributed database system so as to yield maximum composite system performance (as
determined by the global objective function). Remote access to data is generally slower
and more costly than local access to data, because remote access results in network
delays and communication costs in addition to disk access delays and costs. A distribu-
tion of files that results in many remote accesses to data files will undoubtedly degrade
system performance. Burdening a server with large local processing demands may cause
bottlenecks in the system, also leading to degradation in system performance. There-
fore, data allocation algorithm(s) must consider both remote access costs and local
processing costs when computing the new data allocation. Finally, since the query
strategies implemented in the system also contribute significantly to the actual optimal
data allocation, these must also be considered.

Following the pioneering work in Porcar (1982), many researchers have studied the data
allocation problem (Daudpota, 1998; So, Ahmad, & Karlapalem, 1998; Tamhankar & Ram,
1998; Ladjel, Karlapalem, & Li, 1998). The single data allocation problem has been shown to
be intractable (Eswaran, 1974), which means that as the problem size increases, problem
search space increases exponentially (Garey & Johnson, 1979). Due to the complex nature
of the problem, some researchers (Cornell & Yu, 1990; Rivera-Vega, Varadarajan, & Navathe,
1990; Lee & Liu Sheng, 1992; Ghosh & Murthy, 1991; Ghosh, Murthy, & Moffett, 1992) have
resorted to integer programming methods in search for good solutions. Since optimal search
methods can only be used for small problems, heuristic methods are often used for solving
large data allocation problems (Apers, 1988; Blankinship, 1991; Ceri, Navathe, & Wiederhold,
1983; Du & Maryanski, 1988).

Researchers have studied both the static data allocation problem, in which data
allocations do not change over time, and the dynamic data allocation problem (Theel &
Relational Data Modeling for Geographic Information Systems
[www.igi-global.com/article/relational-data-modeling-geographic-information/51216?camid=4v1a](www.igi-global.com/article/relational-data-modeling-geographic-information/51216?camid=4v1a)

Improving the Understandability of Dynamic Semantics: An Enhanced Metamodel for UML State Machines
Eladio Dominguez, Angel L. Rubio and María A. Zapata (2004). *Advanced Topics in Database Research, Volume 3* (pp. 70-89).
[www.igi-global.com/chapter/improving-understandability-dynamic-semantics/4354?camid=4v1a](www.igi-global.com/chapter/improving-understandability-dynamic-semantics/4354?camid=4v1a)