PERFORMANCE EVALUATION OF A DISTRIBUTED DATABASE - A BANKING SYSTEM CASE STUDY

Mbaye Sene [*,***] **Patrice Moreaux** [**]
Serge Haddad [***]

[*] *FST, Université Cheikh Anta Diop, Dakar, Senegal*
[**] *LISTIC, Université de Savoie, Annecy, France*
[***] *LAMSADE, Université Paris Dauphine, Paris, France*

Abstract This paper presents a case study of performance evaluation of a distributed database system made of a real banking system with several branches interconnected via a Wide Area Network (WAN). Due to the complexity of such systems, we use Stochastic Well Formed Petri nets (a high level stochastic Petri net model) to model its activities; the model obtained allows us to compute performance indices of the system. This is done by taking into account the dimensional parameters, the behaviour of the clients submitting requests and the number of clients, the WAN contention and the locality level of the database which enables us to consider the local and remote write requests. Synthesizing these various results, we propose configuration settings matching given workloads and system structures.

Keywords: Distributed database, Coloured Petri Net, Performance evaluation, Simulation, Stochastic Well formed Net.

1. INTRODUCTION

New applications are developed with more and more response time constraints; due to the limits of centralized architecture, distributed architectures have been introduced to enhance performances while providing reliable environments despite components or network failures. These systems are managed by distributed database management system (DDBMS) such as Oracle, Sybase, ..., which have in general three components (figure 1):

- the user process with the user interface, the data semantic control, the global process request and a global data scheme,
- the data process with the local data scheme, the database management system and the redo log manager,
- the WAN which interconnects these two nodes establishing a dialogue which can be in different form (RPC, packets or messages sending, etc.).

Each of theses components can play an important role in the performances of such large systems.

Many tools have been developed to solve the varieties of models used to evaluate the performances of these systems (queueing system, stochastic Petri nets, stochastic automata, etc.). In this paper, we study a banking system by means of Stochastic Well formed Petri Net (SWN) (Chiola *et al.*, 1993). The goal is to evaluate the benefits of partially distributing processing and data across the system. We show how the various parameters of the system may be mapped to SWN features and we present the results of our case study.

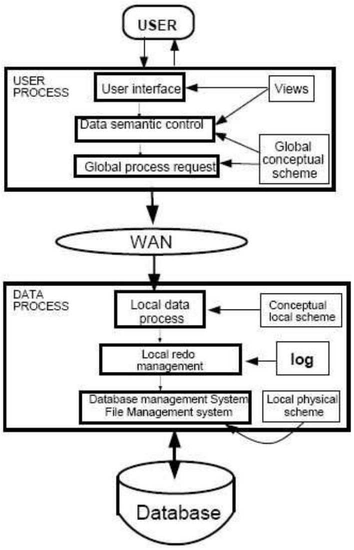

Figure 1. Distributed Database Management Systems (DDBMS) architecture

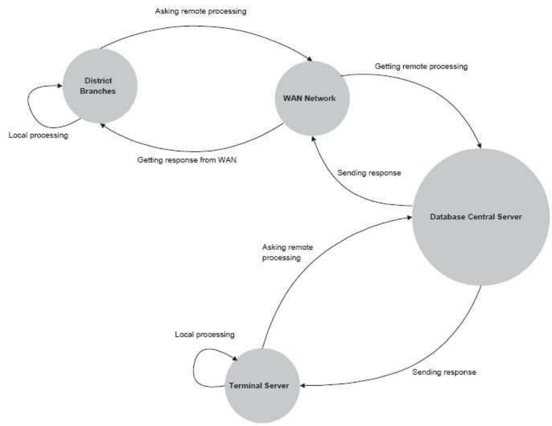

Figure 2. Structure of the banking system (with one branch)

The paper is organized as follows. In section 2, we describe the banking system. Section 3 presents the significant parameters of the system and the SWN model. The analysis method for SWN is explained in section 4 and results are given in section 5. Summary of this study and future work are provided in the last section.

2. THE BANKING SYSTEM

The banking system (figure 2) consists on N branches distributed in a star WAN. The whole database is stored in the central node whose server has two processors. Parts of the database can be stored in each of the $N-1$ branches divided into two categories: branches with slow server and branches with fast server.

Performance study of such system requires (see (Thomasian, 1998) for instance) to know details on different levels (resource allocation, database semantic, etc.). Operations performed by the system are summarized in figure 3. Note in particular that a client's request may be local or remote.

For concurrent processing, many technics such as 3-phases-commit in failure, quorum protocol, recovery protocol etc. have been developed. In this paper the system behaviour and the accurate settings of the banking system for given architectural parameters are studied; only the duplication based on the primary copy is studied, other technics are presented in (Oszu and Valduriez, 1998).

3. MODELLING THE SYSTEM

Since we study an actual system, accurate model and adapted tools must be used. Moreover, we have to devise a model of the system in a sufficiently detailed way.

3.1 Requests type and structural parameters

There are two types of activities: interactive activities and batch activities. Batch mode processing requires specific modelling with regard to the structure of the induced load. In the present work, we study only the interactive mode. Requests generated by the customers depend on three parameters LL, LRL and RRL in the range $[0, 1]$:

- the proportion of local requests compared to remote requests, is denoted by the parameter LL (*Locality Level*). If LL is equal to 0 then all the requests are remote: there is no local request. LL reflects the degree of replication of the database on branches.
- the proportion of local readings compared to distant readings LRL (*Local Read Level*): a local request is either a reading or a writing. If LRL is equal to 0, then all reading requests are remote.
- the proportion of local readings compared to distant writings RRL (*Remote Read Level*): a remote request is either a reading or a writing. LRL and RRL reflect the workload structure of the requests.

In interactive mode, there are three categories of parameters: those related to the servers (number of servers of each category , number of stations, number of database servers), those related to the structure of the system (duplication level, power processing of the servers) and those related to the wide area network (throughput). These parameters are summarized in table 1.

The WAN offers throughput ranging from 19.2 kbit/s to 10 Mbit/s. To take into account the WAN contention, we assume that no more than k requests of a given type can go through the WAN at the same time.

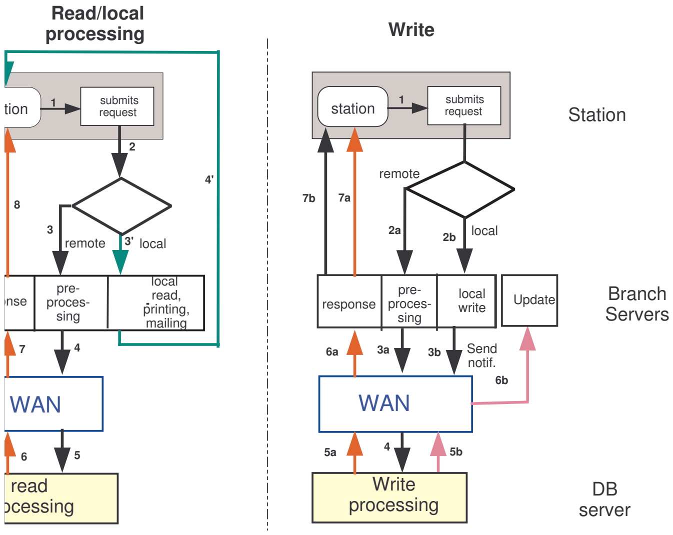

Figure 3. Read/write operations in a DDBMS system

	Name	Description	Values
Request profile			
	LL	Locality Level	0–1
	LRL	Local Read Level	0–1
	RRL	Remote Read Level	0–1
Population profil			
	NBA	Number of branches	8
	NBT	Number of stations	1–20
WAN dimensioning	WTHU	WAN throughput	19.2 Kbits/s - 10Mbits/s
	WCL	Number of requests in WAN	k

Table 1. Parameters of the model

3.2 SWN model

High level Petri nets have contributed to the design and analysis of complex systems but because of the lack of structure on the expressions and on the arc function, it was difficult to use this concision in the model analysis. In order to feel this gap, various classes of high level Petri nets were introduced: regular Petri net, Well formed Petri Nets (WN), Stochastic Well formed Petri Net (SWN). In SWN, colour domains for every place and transition are Cartesian products of basic colour classes usually modelling the various fields of the entities of the system, or the entities themselves for transition colour domains. A basic class is a set of entities which have the same behaviour (class of servers, class of resources). This structure makes it possible to take into account the symmetry of the entities modelled. A symbolic reachability graph (SRG) may be computed where each node (called a symbolic marking) corresponds to a set of coloured markings with same behaviour up to colour symmetries in each basic colour class. Moreover, an arc of the SRG (called a symbolic firing) represents also a set of firings of transition instances, all enabled in any of the markings of the symbolic marking. With systems exhibiting many entities with symmetrical behaviour, the SRG provides a very reduced representation of the Reachability Graph (RG) of the coloured net. Finally, to increase ease of modelling with SWN, partitions of basic colour classes (called static subclasses) were introduced: tokens of the same colour class have the same qualitative behaviour but can belong to different static subclasses, some of their stochastic behaviour (firing rates, etc) being different.

In our model, we introduce two colour classes of tokens (B, T), and the field B is subdivided in two static subclasses: B_1 for slow branch servers and B_2 for the fast ones. A request can be of type: TRR (remote reading), TRW (remote writing), TLR (local reading), TLW (local writing) or TU(updating); these five static subclasses partition the full class T. Formally, we have:

- $B = B_1 \uplus B_2$
- $T = TRR \uplus TRW \uplus TLR \uplus TLW \uplus TU$

We have local and remote processing and the whole class T is:

$$T = TRR \biguplus TRW \biguplus TLR \biguplus TLW \biguplus TU.$$

The mean rate for a request t submitted by a client of branch x of transition ReqS is:

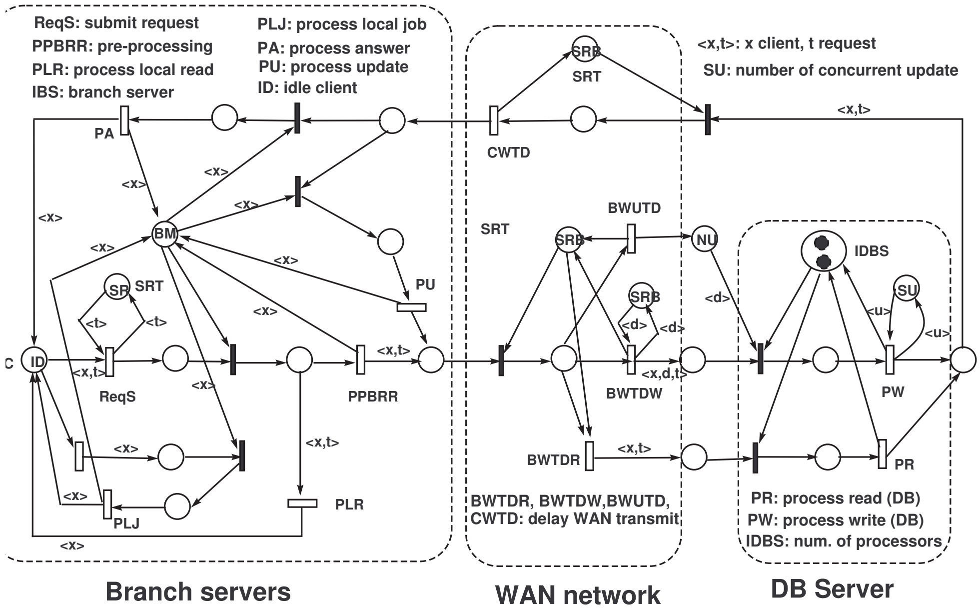

Figure 4. SWN of the complete banking system

$$\mu(ReqS(x,t)) = \begin{cases} LL.LRL.\mu & \text{if } t \in TLR \\ LL.(1-LRL).\mu & \text{if } t \in TLW \\ (1-LL).RRL.\mu & \text{if } t \in TRR \\ (1-LL) & \\ .(1-RRL).\mu & \text{if } t \in TRW \end{cases} \tag{1}$$

where μ is the basic rate obtained by measurement on the real system. The resulting SWN model od the system is given in figure 4.

4. ANALYSIS

We recall here the definition of performance criteria according to the SWN semantics by specifying how we can obtain them.

4.1 Analysis methods

We are interested in global performance indices such as mean response time, queue length at servers, queue length at WAN and utilization of the servers.

The response time $\overline{R}$ to a user's request is a function f depending on several variables:
$\overline{R} = f(WTHU, LL, LRL, RRL, NBT, NBA, MPLC, WCL, MPLA)$.

As this is frequently the case, the number of parameters is an intrinsic difficulty of such a study.

Our main goal is to determine the values of the parameters of the function f which minimize $\overline{R}$ or maintain it at a required value. We have basically three components in our models: the branches, the WAN and the central server. In the rest of the paper (unless explicitly mentioned), we will denote by TR the average global response time i.e. without distinguishing neither the request type, nor the branch from which the request is generated.

To study the behaviour of the system, we fix a configuration corresponding to the current situation called the *reference configuration.*

4.2 Performance indices of the model

Let us now recall how to determine the performance indices which are of interest in the context of SWN. Consider for instance transition $ReqS$ which models submitting requests of clients. Its mean throughput for $t \in T$ and $a \in B$ (i.e request type t from a branch b) is given by:

$$\overline{\chi}(ReqS,a,t) = \sum_{\substack{\widehat{\mathbf{m}} \in TSRS \\ \widehat{\mathbf{m}}[ReqS(a,t) >}} w(ReqS,a,t).\pi(\widehat{\mathbf{m}}).$$

$\widehat{\mathbf{m}}$ is the symbolic tangible marking and $w(ReqS,a,t)$ is the firing rate for static subclasses (a,t). Let us denote by P' the set of places visited by a request (a,t) starting from its generation and before the end of the receiving of its response, the mean number of requests $\overline{N}(a,t)$ being processed is

$$\overline{N}(a,t) = \sum_{p \in P'} \widetilde{\mathbf{m}}(p,a,t)\pi(\widehat{\mathbf{m}})$$

where $\widetilde{\mathbf{m}}(p,a,t)$ is the marking of place P for the static Cartesian product of the tuple (a,t) in $\widehat{\mathbf{m}}$ and $\pi(\widehat{\mathbf{m}})$ the steady probability for the net to be in marking $\widehat{\mathbf{m}}$. Using the Little law $\overline{N} = \overline{\chi}.\overline{R}$, the

	RL=0.01	RL=0.10	RL=0.50	RL=0.90	RL=0.99	Same RL				
LL	READINGS					WRITINGS				
0.01	833	1041	1223	1152	1086	381	496	628	536	425
0.10	833	1021	1091	1024	1038	431	642	802	732	494
0.50	734	774	1007	905	957	508	748	1375	896	567
0.90	667	556	634	697	779	415	524	598	636	465
0.99	696	556	494	548	630	400	441	480	436	367

Table 2. Response time (in ms) of readings/writing according to LL and RL

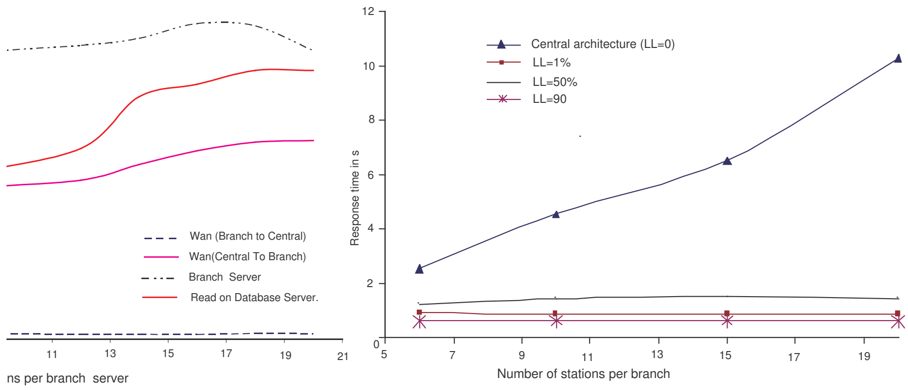

Figure 5. Queue length/Response time versus number of stations/branch

average response time for a request can thus be computed as:

$$\overline{R(a,t)} = \frac{\overline{\mathbf{N}}(a,t)}{\overline{\chi}(ReqS,a,t)}$$

Based on the SWN model theory, tools make it possible to compute automatically the quantities $\overline{N}$ and $\overline{\chi}$ from the definition of the SWN.

4.3 Computing the performance indices

For "small" systems, the tool *PERFSWN* (Haddad *et al.*, 2001) can be used to compute performance indices. However when the size and the complexity grow, the analytical resolution is out of reach of current tools because of the size of this generated chain. This is a general phenomenon and in most cases simulation technics are used instead. The main results which follow was thus obtained thanks to the new simulator WNSIM (Franceschinis *et al.*, 2001), (Franceschinis *et al.*, 2002).

WNSIM is an event driven stochastic Discrete Event System simulator based on Generalized Stochastic Petri Nets (GSPN) or SWN. It allows the user to specify all parameters of the simulation run: size of the batches, length of the initialization phase, confidence levels and confidence intervals. Moreover, WINSIM simulation runs may be controlled by selected performance indices computation. In all our simulations we have taken a confidence level of 5% and a confidence interval of 0.95: given a simulation result r_s of a performance measure r, this means that the exact value r_e of r satisfies $Prob(r_e \in [0.95r_s, 1.05r_s]) \geq (1 - 0.05)$

5. RESULTS

In this paper, system performance is analyzed according to the request profile composed by the locality level and the proportion of the readings. We would like to forecast the behaviour of the system when the load (the number of stations per agency) increases. Finally, we examine the influence of the simultaneity of the updates on the performances.

5.1 Response times according to the request profile

Response times for readings and writings for various locality levels are reported in table 2.

For a given reading level (RL), TR decreases when the locality level increases since local readings are treated by the branch servers. In contrast, for a given LL, TR increases with RL until $RL = 0.50$ or $RL = 0.90$ depending on LL. This counter-intuitive result, since readings are less loading the system than writings, was not foreseeable a priori and can be highlighted only from computation of performance indices. With regard to writings, the evolutions are different. In addition, for a given rate of writings ($1 - RL$), the response time grows until $LL = 0.50$ and then decreases for any value of RL.

5.2 *Queue length and response time versus number of stations*

The number of simultaneous updates is limited (and modelled by the initial marking of the place *AU*). Figure 5 gives the queue length in the WAN and at the servers for various values of the number of stations. The Wan Branch to Central (resp. Central to Branch) curve corresponds to the Wan queue at the branches (resp. the server). The two other curves correspond to servers queues.

Response time for central and distributed architecture are given also in this figure.

We note that the queue length of the branch server queue is significantly smaller than the queue length of the WAN from branch to central node which hence appears as the bottleneck from the branch point of view. On the other hand, the central node sends large quantity of data to branch servers, so that when the WAN slows down, the queue length from central node to branches through the WAN increases. The read process in the central database server increases due to the small number of active processors.

6. CONCLUSION

In all configurations, the throughput of the WAN is a significant limiting factor to increasing the global performance of the system. We have seen that in WAN, from central node to branch, the queue length increases rapidly with the number of stations; so a more accurate configuration should increase the bandwidth of the WAN. The interest in a distributed system is moreover variable according to the request profile and is effective only if one authorizes more than one update in the WAN for a given branch. However, the power processing of the branches can be the bottleneck; hence, raising their power 3 to 4 times will make it possible to approximately maintain response time in the same order of magnitude to 20 stations. On a finer level, we highlighted the influence of the profile of request on the response time for various kinds of request. Finally we note that an analysis only based on qualitative features could not detect this behaviour. This shows the interest of the quantitative analysis of such systems.

Future work will study the impact of batch mode processing when introduced concurrently with interactive processing.

REFERENCES

Chiola, G., C. Dutheillet, G. Franceschinis and S. Haddad (1993). *Stochastic well-formed colored nets and symmetric modeling applications. IEEE TC.*

Franceschinis, G., R. Gaeta and C. Bertoncello (2001). *WNSIM: manual.* PEG, Dipart.di Informatica. Univ. di Torino (Italy).

Franceschinis, G., R. Gaeta and C. Bertoncello (2002). *GreatSPN: User's Manual (version 2.0.2).* PEG, Dipart. diInformatica. Univ. di Torino (Italy).

Haddad, S., P. Moreaux and M. Sene (2001). *Performance Evaluation with SWN: a technical contribution. Réseaux et systèmes répartis - CP Vol 13 N 6.*

Oszu, M. Tamer and P. Valduriez (1998). *Principles of Distributed Database Systems.* Prentice Hall International. 2nd edition.

Thomasian, A. (1998). *Performance Analysis of Database Systems. Proc. of the 10th Int.Conf. on CPE: Modelling Techniques and Tools.*