Analytical Modelling of Priority Commit Protocol for Reliable Web Applications

I. Awan
Department of Computing
University of Bradford, Bradford, UK
i.awan@scm.brad.ac.uk

M. Younas
School of MIS
Coventry University, Coventry, UK
m.younas@coventry.ac.uk

ABSTRACT
Web applications are vulnerable to failures and low performance due to the large population of users and the widespread distribution of Internet. Transaction technology provides Web applications with high reliability and improved performance. This paper presents a novel approach for the efficient commit processing of Web transactions. The proposed approach is based on the implementation of priority active network scheduling mechanism at each network node. It involves rigorous analysis of a network node with finite capacity to accommodate messages, bursty arrival process to represent incoming multi-class messages, and the employment of a priority scheduling mechanism to give preferential treatment to high priority messages. This analytical solution provides closed form expressions for calculating the queuing delay per class involved in the commit processing of Web transactions at each network node. The proposed approach significantly reduces the queuing delay for high priority messages such as commit, abort, and compensation of Web transactions. Consequently, performance of the commit processing of Web transactions is improved as response time of the nodes responsible for making decision is reduced.

1. INTRODUCTION
Active network technology allows network nodes to perform customised computation on the messages involved in processing of distributed applications. Thus part of the applications processing is accomplished at the network nodes. Active networks are used to improve the performance of distributed applications such as mixing sensor data, online auctions, and Web transactions [1, 2, 14].

Transaction management (TM) technology provides a facility wherein several operations can be treated as a single logical unit of work performed as a part of one application [3]. For example, an airline reservation application should either book the flight and transfer the money from customer’s account to travel agent’s account or must not perform any of these activities. Existing approaches use various transaction models and protocols in order to improve the performance and reliability of Web transactions [6, 7, 8]. However, they ignore the underlying network capabilities that can well be exploited to achieve improved performance of the Web transactions.

In this paper we propose a novel approach which is based on the employment of head-of-line (HoL) priority active network scheduling mechanism [11] at each network node. This approach is called priority commit protocol (PCP) and is built on our previous work [14]. The main purpose for employing HoL priority mechanism is to reduce the queuing delay involved at each node. In order to calculate this queuing delay, each node of an active network has been modelled as a queueing system with finite capacity. The arriving external traffic at each node is bursty as messages from various transactions can arrive simultaneously. This burstiness of traffic has been modelled by Compound Poisson Process (CPP). Each node may have multiple processors and hence can execute various transactions related messages simultaneously. This concurrent execution has been modelled using a generalised exponential (GE) distribution [10]. Based on such information, each node has been analysed as a GE/GE/1/N/HoL queueing system with HoL priority scheduling discipline. This analytical solution provides closed form expressions to calculate the queuing delay at each network node. PCP takes into account the actual queuing delay at each node of the active networks, which are involved in commit processing of the Web transactions. Using PCP, the queuing delay for high priority messages (such as commit, abort and compensate) can be reduced to the remaining service time of a low priority message in service, or the total service time required for all the high priority messages in front of it. The contributions of this paper are summarized as follows:

- Analysis of a GE/GE/1/N/HoL queue using maximum entropy (ME) methodology to calculate the queuing delay at each network node with bursty traffic under HoL priority service discipline.

- Performance of the commit processing is significantly improved by reducing the queuing time of the network nodes. This is crucial in the Internet environment where big queues can develop at the network nodes when huge number of transactions is being processed. PCP can be used to improve the performance of almost all of the existing commit protocols such as two-phase commit (2PC) protocol, presumed abort (PA) proto-
col, Web Services transactions, and Business Transaction Protocol (BTP) [7, 8].

- Real time transactions (such as online auctions and stock markets) can benefit from PCP as such transactions are required to meet deadlines and they can not tolerate longer delays caused by the network traffic.

The paper is structured as follows. Section 2 critically analyses the current research work on the Web transaction processing. Section 3 presents the proposed PCP. This section also describes the analysis of a GE/GE/1/N/HoL queue to model network nodes and illustrates the criteria for performance evaluation of PCP. Section 4 provides experimental results through the evaluation of PCP in comparison to existing approaches. Section 5 concludes the paper.

2. RELATED WORK

TM technology has widely been employed in various Web applications including, simple e-commerce applications of ordering a pizza [2, 5] or more advanced applications such as B2B and supply chain as in Web Services [6, 7, 9]. Existing approaches incorporate various transaction models and protocols so as to better meet the requirements of the Web applications. These include the classical ACID transactions and the extended transactions models such as open-nested transactions, compensating transactions, flexible transactions, conversational transactions and so on.

In order to implement the classical or extended transaction models, a variety of commit processing techniques have been defined such as two-phase commit (2PC), presumed abort (PA), implicit yes vote (IYV), presumed commit (PC) and optimistic commit protocols. 2PC and PA protocols have been employed both in classical [2, 5] as well as advanced Web transaction processing [4, 6, 7]. Variants of 2PC, such as PA, PC, ANPA, and optimistic commit protocols [2] are defined in order to improve the performance of 2PC by reducing the numbers of messages and forced-write operations, or employing the unilateral commit strategy as that of open-nested transactions [4].

Current approaches also use extended transaction models so as to manage complex Web applications (or business activities) that involve autonomous and heterogeneous business systems. Papazoglou [6] proposes a business transaction framework (BTF) for Web services and outlines requirements and characteristics of business transactions. WS-Transactions approach [7] is developed by IBM and Microsoft in order to manage effectively Web transactions in business activities. WS-Transactions and BTF aim to define a framework for providing transactional coordination of participants of services offered by multiple autonomous businesses that are based on Web services technologies. Further, OASIS [9] defines a protocol for Web services transactions, called Business Transaction Protocol (BTP).

The above approaches also incorporate compensating transaction model. Such model allows that component transactions of a Web transaction can be unilaterally committed without the commitment of their sibling component transactions. Advantage of compensating transaction model is to save system resources (e.g., early release of locks) and improve performance, as component transactions are not required to wait for the completion of overall Web transaction. However, the current compensation model has consequences for Web applications. For example, if the execution of compensating transaction for cancellation of a hotel room is delayed (e.g., due to heavy Internet traffic) then it may cause financial loss to the hotel management. This is because the room can not be booked for other customers until the booking (made by previous transaction) is cancelled. In order to fully benefit from the use of compensating transactions, current approaches must manage them properly. There must be a mechanism which optimise the execution of compensating transactions so that they are not affected by the heavy Internet traffic or long queuing delays. Network delay is one of the main performance issues in the Web environment. Unfortunately, majority of the existing approaches fall short of investigating into the underlying network infrastructure, which significantly affects the performance of Web transactions. To our knowledge, the sole approach in [2], uses active networks and proposes ANPA (active network presumed abort) protocol in order to improve the performance of the classical presumed abort (PA) protocol in the Internet. However, this approach has the limitation that the delay at each node has been considered at an abstract level and is same for all messages regardless of their importance (or priority). Bernstein et. al. [13] use transactions with queuing system in order to reliably communicate requests between clients and servers. However, this approach neither considers active networks nor the performance of transactions.

3. THE PROPOSED APPROACH

This section presents the proposed PCP. It also describes the analysis of a GE/GE/1/N/HoL queue to model network nodes and illustrates the criteria for performance evaluation of PCP.

3.1 Priority Commit Protocol

We propose to employ a priority scheduling mechanism using active networks in order to improve the performance of commit processing of Web transactions. The proposed approach models each node of an active network by a GE/GE/1/N/HoL priority queuing system with multi-class messages. This analysis will produce closed form expressions for the queuing delay at each node. The arrival process of the bursty incoming messages is represented by the CPP and transmission times of the messages are represented by GE distribution. We represent the total processing capacity of each network node by a single server and total capacity to accommodate messages in the queue is N. Service discipline at each node has been modelled by HoL priority mechanism [11] in order to give preferential treatment to control messages such as commit, abort, and compensate.

Priority scheduling mechanisms, employed at any node in a network, can play important role when distinct types of messages are being processed. These mechanisms give preferential treatment to some messages as compared to others and can significantly improve the overall performance of the system. Under a priority scheduling mechanism, distinct classes of messages are assigned different priorities. These priorities determine the order of service among classes. Scheduling priority mechanisms, such as head-of-line (HoL) [11], take into account that some services may tolerate longer delays than others (e.g., data versus control packets) and deal with the order with which messages are required to be processed. According to HoL mechanism, a message with
highest priority upon its arrival always get accommodation in front of the low priority messages in the queue. It only waits either for the remaining service time of the low priority message being processed or the sum of service times of all the high priority messages in front of it.

Generally there are two classes of messages in Web transaction, namely, data messages and control messages such as commit, abort, prepare, and compensate. Delay for the control messages can be high at nodes with larger queues and no service priority mechanism. This delay of the control messages can be reduced by employing a suitable priority scheduling mechanism at each node. It is crucial to reduce such delay as the control messages are required to be processed well in time. For example, delay in the compensation of a committed transaction may result in data inconsistency or financial losses, as described earlier. Keeping in view the simplicity of processing involved for a scheduling mechanism, we use HoL mechanism for the proposed PCP. Let each node in the network is equipped with a finite capacity buffer to store the incoming messages. The total time that a message spends in the node is the sum of the waiting time and the processing time. Waiting time for each message is the sum of processing times for all the messages in front of it. PCP enables control messages to avoid any queueing delay at each node.

3.2 ME Analysis of GE/GE/1/N/HoL Queue

This sub-section describes GE/GE/1/N/HoL priority queueing system which is used to analyse each network node involved in the processing of Web transactions.

Consider a stable single server GE/GE/1/N/HoL queue. For each class \( i (i = 1, 2, ..., R) \), let \( \lambda_i \) be the mean arrival rate, \( C_{si}\) the interarrival time SCV, \( \mu_i \) be the mean service rate and \( C_{si}^2 \) be the service time SCV. Let at any given time, \( n_i (0 \leq n_i \leq N) \), \( \sum_{i=1}^{R} n_i \leq N \), be the number of class \( i \) messages in the queue (waiting and/or receiving service), \( S = (n_1, n_2, ..., n_R, \omega) \) be a joint state, where \( \omega (1 \leq \omega \leq R) \) denotes the class of the current message in service and \( Q \) be the set of all feasible states \( S \). The form of the state probability distribution \( \{P(S), S \in Q\} \) of a GE/GE/1/N/HoL queue, can be characterized by maximizing the entropy functional

\[
H(P) = - \sum_{S \in Q} P(S) \log P(S),
\]

subject to prior information expressed in terms of the normalization and, for each class \( i (i = 1, 2, ..., R) \), the marginal constraints of server utilization, \( U_i(0 < U_i < 1) \), busy server probability, \( \theta_i(0 < \theta_i < 1) \) with \( n_i > 0 \), mean queue length, \( L_i(U_i \leq L_i < N) \) and conditional full buffer state probability, given that a class \( i \) message is in service, \( \phi_i(0 < \phi_i < 1) \), satisfying the flow balance equations, namely

\[
\lambda_i(1 - \pi_i) = \mu_i U_i, \quad i = 1, 2, ..., R,
\]

where \( \pi_i \) is the marginal blocking probability that an arriving message of class \( i \) finds \( N \) messages in the queue. By employing Lagrange’s method of undetermined multipliers, the ME solution can be expressed by

\[
P(S) = \frac{1}{Z} \prod_{i=1}^{R} g_i^{n_i(S)} \xi_i^{n_i(S)} \tau_i^{n_i(S)} y_i^{f_i(S)}, \quad \forall S \in Q,
\]

where \( Z \) is the normalizing constant, \( \{g_i, \xi_i, \tau_i, y_i, \forall i \in R\} \) are the Lagrangian coefficients corresponding to constraints \( \{U_i, \theta_i, L_i, \phi_i\} \) respectively and \( \{s_i(S), h_i(S), n_i(S), f_i(S)\} \) are suitably defined auxiliary functions (c.f., [12]). Aggregating (3) over all feasible states \( S \in Q \), and after some manipulation, a closed form expression for the aggregate joint state probability distribution \( \{P(n), n = (n_1, n_2, ..., n_R)\} \) of a stable GE/GE/1/N/HoL queue can be established (c.f., [12]). Utilizing the ME product-form solution, the mean marginal and aggregate delays, \( \{W_{ti}, i = 1, 2, ..., R\} \) and \( W_t \), respectively, can be obtained. In particular, the marginal mean delay can be clearly determined (via Little’s Law) by

\[
W_{ti} = \frac{L_t}{\hat{\lambda}_i}, \quad i = 1, 2, ..., R,
\]

where \( \hat{\lambda}_i = \lambda_i(1 - \pi_i) \) is the mean effective arrival rate of class \( i \) messages and \( L_t \) is the marginal mean queue length. The mean aggregate delay can be determined by

\[
W_t = \sum_{i=1}^{R} \hat{\lambda}_i W_{ti}, \quad \hat{\lambda} = \sum_{i=1}^{R} \hat{\lambda}_i
\]

Note that, as in earlier works (c.f., [10]), the Lagrangian coefficients \( \{g_i, \xi_i, x_i, y_i, i = 1, ..., R\} \) of the ME solution (3) for GE-type queues are largely invariant to the buffer size \( N \) and can be approximated via closed form asymptotic queueing theoretic expressions based on the ME solution of the corresponding infinite capacity GE/GE/1 queue at equilibrium. Moreover, using the flow balance condition (2) and closed-form expressions for the normalizing constant, \( Z \), the aggregate probabilities \( \{P(n), n = 0, 1, ..., N\} \) and the blocking probabilities \( \{\pi_i, i = 1, ..., R\} \), the Lagrangian coefficients \( \{g_i, i = 1, 2, ..., R\} \) can be recursively determined and can be seen in [12].

3.3 Performance Evaluation Criteria

We devise different mathematical expressions in order to calculate the commit and abort delays of a Web transaction. The following expression is defined to calculate the commit delay of a Web transaction using PCP:

\[
(3 \ast H - 2 \ast h) \ast (M - W_t) + 2 \ast W + A
\]

where \( H \) is the number of hops between the coordinator and participants of a Web transaction, \( h \) is the distance in terms of number of nodes between the coordinator and participants, \( M \) is the minimum delay (queueing + service time) for sending a message across a hop in the Internet, \( W \) is the minimum delay for a force-written log, and \( A \) is the message processing cost at active nodes, \( W_t \) is the queuing time for a message at any node and is equivalent to the processing time of the low priority message in service or the time required to serve all the high priority messages in front of it. \( W_t \) is very low for high priority messages (control messages) as, generally, the number of control messages is far less than the data messages at any node.

An expression for the abort decision can also be derived in a similar way and is given as follows:

\[
(3 \ast H - 2 \ast h) \ast (M - W_t) + W + A
\]

As described earlier, none of the current approaches give attention to queuing delays. All of them calculate the delay at each node based on non-priority scheduling mechanism such as first come first served. The most relevant approach, ANPA [2], also ignores such delays. According to ANPA,
4. EXPERIMENTAL RESULTS

Figure 1: Marginal utilisations for high priority messages

Figure 2: Marginal utilisations for low priority messages

the delay for a commit decision can be calculated using the following expression.

\[(3 * H - 2 * h) * M + 2 * W + A \]  \quad (8)

ANPA reduces the number of messages \( M \), between the participating systems of the commit protocol, thus improving the performance of PA protocol. Delay in PA protocol is calculated using the following expression:

\[3 * H * M + 2 * W \]  \quad (9)

4. EXPERIMENTAL RESULTS

This section first illustrates the credibility of the analytical solution by comparing its results against simulation. It then describes the significance of the proposed approach by comparing its performance against existing commit protocols of the Web transactions.

Simulation is an efficient tool for studying detailed system behaviour but it becomes costly, particularly as the system size increases. Analytical solutions, on the other hand, provides cost-effective and efficient tools for calculating system performance measures such as delay.

Numerical tests are carried out to verify the relative accuracy of the ME solution against simulation at 95% confidence intervals based on the Queueing Network Analysis Package (QNAP-2), using the same assumptions and input parameterization as the ones used for the analytic ME solution (c.f., Figs. 1-3). It is observed that the ME results are very comparable to those obtained via simulation. Moreover, it can be seen that the traffic burstiness (interarrival-time SCV) has an inimical effect, as expected, on the mean response time per class (c.f., Fig. 3). Results show that high priority messages face less mean response times as compared to the low priority messages.

Based on the analytical solution we have performed a number of experiments in order to evaluate the performance of PCP by considering different cases. We follow the current standard benchmark, for evaluating the performance of PCP. The current benchmark is used to determine the delay required to commit or abort a transaction [2]. Such delay is calculated using: (i) number of messages (ii) number of forced write operation (decision writing to disk) (iii) local processing time of a component transactions, (iv) message propagation and (v) delay at each node. We follow the current benchmark but also take into account the queuing delay, which is not considered by the current approaches. We use the following data in order to calculate the total delay for the commit and abort of a Web transaction: \( H = 5 \), \( W = 15 \text{ms} \), \( A = 0.4 \text{ms} \), \( M = 10 \text{ms} \). These values may vary depending on the nature of the Internet traffic and the load on computer systems [2],[14]. We use \( W_t = 5 \text{ms} \) as an average queuing time at each active network node calculated using expressions (4)-(5). Note that the value of \( h \) defines the distance between the participating systems such as the coordinator and participants. As in active networks, any node can make decision, the value of \( h \) varies over the total number of nodes between the coordinator and participants.

Using the expressions (6)-(9) (c.f., sub-Section 3.3), the performance of PCP has been evaluated in comparison to ANPA and PA protocols as these are the most relevant approaches. ANPA is chosen as it is based on the active networking mechanism. PA protocols is chosen as it is widely employed in existing Web transaction commit processing such as Web services transactions and OASIS Business Transaction Protocol [5],[6],[7].Table 1 presents the experimental results which are also shown in Figs.4-5.

<table>
<thead>
<tr>
<th></th>
<th>h = 1</th>
<th>h = 2</th>
<th>h = 3</th>
<th>h = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA (Commit)</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>ANPA (Commit)</td>
<td>160.4</td>
<td>140.4</td>
<td>120.4</td>
<td>100.4</td>
</tr>
<tr>
<td>PCP (Commit)</td>
<td>95.4</td>
<td>85.4</td>
<td>75.4</td>
<td>65.4</td>
</tr>
<tr>
<td>PA (Abort)</td>
<td>165</td>
<td>165</td>
<td>165</td>
<td>165</td>
</tr>
<tr>
<td>ANPA (Abort)</td>
<td>145.4</td>
<td>125.4</td>
<td>105.4</td>
<td>85.4</td>
</tr>
<tr>
<td>PCP (Abort)</td>
<td>80.4</td>
<td>70.4</td>
<td>60.4</td>
<td>50.4</td>
</tr>
</tbody>
</table>

It is obvious from Figs.4-5 that the commit and abort delays of Web transactions are significantly reduced using PCP as compared to ANPA and PA protocols. These results show that PCP improves the performance of Web transactions by reducing the network delays. PCP can therefore be used in the existing Web transaction commit protocols in order to achieve improved performance. PCP is generic in the sense as it is applicable to all the existing protocols. PCP does not demand any modification of the existing protocols such as reduction in the number of messages or other operations. On the other hand existing approaches require such modification in order to achieve improved performance. For example, PA protocol is defined to reduce the number of messages and
Distance between coordinator and participant
Total commit delay

Legend
PA
ANPA
PCP

Figure 4: Comparison of commit delay for various protocols

Figure 5: Comparison of abort delay for various protocols

forced-write operations of the 2PC protocols.

5. CONCLUSION

This paper presented a new protocol, called PCP, in order to improve the performance of Web transactions. PCP incorporates HoL scheduling mechanism at the network nodes in order to reduce the queuing delays for transaction-related messages. We have also presented the analysis of a GE/GE/1/N/HoL priority queuing system which is used to model each network node involved in the processing of Web transactions. This analytical solution is used in order to carry out a number of experiments to test the performance of PCP in comparison to existing approaches. Credibility of the analytical solution is also tested by comparing its results against simulation. A number of experiments have been performed so as to evaluate the performance of PCP. Experimental results show that PCP significantly improves the performance of the Web transactions.

6. ACKNOWLEDGMENTS

This work is supported by the EPSRC, UK under grant GR/S01658/01.

7. REFERENCES