LaTeX Author Guidelines for 8.5×11-Inch Proceedings Manuscripts

Miguel Belém
83531

Tiago Gonçalves
83567

Vítor Nunes
83576

Abstract

In this project we will apply two different aproaches to solve the problem of having a distributed tupplespace. These aproaches have different tradeoffs considerations regarding time spent executing a request, network congestion and time spent recovering from a fault. For this work we consider a perfect failure detector, in which if a server stop responding to requests, then is because it crashed.

1. Introduction

A tuple space consists in distributed collection of tuples. A tuple contains fields that are some how related. A basic tuple space needs to be writable, readable and deletable.

Challenges. At a first glace seems easy to implement a tuple space in a distributed way however problems like consistency, data loss and performance rise.

To overcome this challenges we propose two diferent implementations of a distributed tuple space which solve the challenges listed above.

The State Machine Replication (SMR) which uses a coordination mechanism envolving one leader to be responsible for coordination.

And a Xu and Liskov implementation that discards the need of a leader to coordinate client's requests.

2. State Machine Replcation

The State Machine Replication is based on primary backup approach. Given one primary and several secondary nodes, clients will perform requests to the primary node only.

When the first server is switched on, we start by searching for alive servers requesting AREYOUTHEMASTER. If no reply is given then the server becomes the leader. When others servers join the process repeats but they will receive the reply containing the URL path of the master.

2.1. Clients request

In our solution when the client starts we begin with a list of all URL's servers, sends the request to all servers and only the leader will start processing, others simply discard the request.

2.2. Fault tolerance

This implementation has an obvious **week point**: if the master crashes the system will stop repling to requests. Our solution to this problem consists of using an heartbeat that is sent from the secondary nodes to the primary node every random seconds between 3 to 10 seconds. This ensures that in the worst case the system will took ten seconds to recovers (assuming instant propagation of messages) . The randomization process is used to optimize network bandwith and prevent heartbeats flooding on the master.

Every node in the SMR contains an identifier. When a master crashes the remaining node with the smallest ID will be elect as the new master. It starts by informing all secondary nodes about its leadership change. By that time all secondary nodes start sending heartbeets to the new master.

Another problem is when a **secondary node crashes**. In this case the availability of the system is not affected since the primary node continues to receive client's request. Nevertheless if that same node resurrects then it will be unconsistent with others. To fix this problem, every node has a log.

A log is used to keep track of every request made by the master **that was already executed.** When a secondary node crashes and recovers it will have no such tuple in the tuple space, then request to the master a copy of his own log. The master suspend the client's requests processing and replies to the new secondary node. As soon the new node is ready the master continues to processing client's request.

3. Xu and Liskov

De acordo com esta implementação já não se recorre a um master que garante a consistência do sistema coordenando todos os pedidos. Em vez disso, cada Front end de cada cliente vai propagar o pedido para todos os servidores vivos, e cada um desses servidores vai tratar de o executar.

3.1. Clients request

Na nossa solução, quando um cliente quer realizar uma lista de um ou mais pedidos, vai inicialmente criar um Front End, que vai procurar quais os servidores vivos, e com isso é criada a View do cliente, que tem um ID. Quando um pedido é enviado para os servidores, é anexado ao mesmo o ID da view do cliente, o que permite ao servidor saber se o cliente está ou não sincronizado com os servidore, caso não esteja, é porque o sistema sofreu uma modificação e quando a mesma é terminada, a nova View de servidores vivos é enviada ao cliente.

3.2. Fault tolerance

Esta abordagem ao não apresentar um master, tem a vantagem de que quando um nó crasha, seja ele qual for, o sistema não precisa de parar. Isto é possível pois não existe uma unidade central da qual todo o sistema dependa, e que caso a mesma crashe, todo o sistema é comprometido. Apresenta uma desvantagem em relação ao SMR que é a de precisar de parar o sistema por breves instantes quando um novo nó se liga ao sistema. Isto é feito pois as Views precisam de ser atualizadas para passar a conter o novo nó, e não se pode perder pedidos que estejam a ser feitos enquanto as Views são atualizadas.

Quando um nó crasha o cliente vai atualizar a sua view, e informar todos os seus membros para atualizarem a sua view. Quando um nó rescuscita, vai começar por informar todas as máquinas vivas de que precisa do seu espaço de tuplos e vai pedir para que atualizem as suas views. Cada máquina ao receber este pedido vai recusar pedidos dos clientes que apresentem uma view antiga e vai dar o seu tuplespace ao novo nó. O novo nó vai fazer uma interseção de todos os tuplespaces (a interseção garante que ele apenas vai guardar os pedidos que já foram executados em todas as máquinas). Em seguida esse tuplespace é propagado para todo o sistema, com isto garantimos que nunca se perdem pedidos, pois mesmo que um pedido já tenha sido enviado para uma máquina mas não para outra, o mesmo vai ser repetido. Após todos os tuplespaces terem sido atualizados, o novo nó sinaliza as outras máquinas para que mandem a sua nova view ao cliente, repetindo o mesmo todos os pedidos da mesma operação que foram enviados numa view antiga.

3.3. Printing your paper

Print your properly formatted text on high-quality, 8.5 × 11-inch white printer paper. A4 paper is also acceptable, but please leave the extra 0.5 inch (1.27 cm) at the BOTTOM of the page.

3.4. Margins and page numbering

All printed material, including text, illustrations, and charts, must be kept within a print area 6-7/8 inches (17.5 cm) wide by 8-7/8 inches (22.54 cm) high. Do not write or print anything outside the print area. Number your pages lightly, in pencil, on the upper right-hand corners of the BACKS of the pages (for example, 1/10, 2/10, or 1 of 10, 2 of 10, and so forth). Please do not write on the fronts of the pages, nor on the lower halves of the backs of the pages.

3.5. Formatting your paper

All text must be in a two-column format. The total allowable width of the text area is 6-7/8 inches (17.5 cm) wide by 8-7/8 inches (22.54 cm) high. Columns are to be 3-1/4 inches (8.25 cm) wide, with a 5/16 inch (0.8 cm) space between them. The main title (on the first page) should begin 1.0 inch (2.54 cm) from the top edge of the page. The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be 1-1/8 inches (2.86 cm) from the bottom edge of the page for 8.5 × 11-inch paper; for A4 paper, approximately 1-5/8 inches (4.13 cm) from the bottom edge of the page.

3.6. Type-style and fonts

Wherever Times is specified, Times Roman may also be used. If neither is available on your word processor, please use the font closest in appearance to Times that you have access to.

MAIN TITLE. Center the title 1-3/8 inches (3.49 cm) from the top edge of the first page. The title should be in Times 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Leave two blank lines after the title.

AUTHOR NAME(s) and AFFILIATION(s) are to be centered beneath the title and printed in Times 12-point, non-boldface type. This information is to be followed by two blank lines.

The ABSTRACT and MAIN TEXT are to be in a two-column format.

MAIN TEXT. Type main text in 10-point Times, single-spaced. Do NOT use double-spacing. All paragraphs should be indented 1 pica (approx. 1/6 inch or 0.422 cm). Make sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines

Figure 1. Example of caption.

between paragraphs. Figure and table captions should be 10-point Helvetica boldface type as in
Long captions should be set as in

Figure 2. Example of long caption requiring more than one line. It is not typed centered but aligned on both sides and indented with an additional margin on both sides of 1 pica.

Callouts should be 9-point Helvetica, non-boldface type. Initially capitalize only the first word of section titles and first-, second-, and third-order headings.

FIRST-ORDER HEADINGS. (For example, **1. Introduction**) should be Times 12-point boldface, initially capitalized, flush left, with one blank line before, and one blank line after.

SECOND-ORDER HEADINGS. (For example, **1.1. Database elements**) should be Times 11-point boldface, initially capitalized, flush left, with one blank line before, and one after. If you require a third-order heading (we discourage it), use 10-point Times, boldface, initially capitalized, flush left, preceded by one blank line, followed by a period and your text on the same line.

3.7. Footnotes

Please use footnotes sparingly[1] and place them at the bottom of the column on the page on which they are referenced. Use Times 8-point type, single-spaced.

3.8. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [1]. Where appropriate, include the name(s) of editors of referenced books.

3.9. Illustrations, graphs, and photographs

All graphics should be centered. Your artwork must be in place in the article (preferably printed as part of the text rather than pasted up). If you are using photographs and are able to have halftones made at a print shop, use a 100- or 110-line screen. If you must use plain photos, they must be pasted onto your manuscript. Use rubber cement to affix the images in place. Black and white, clear, glossy-finish photos are preferable to color. Supply the best quality photographs and illustrations possible. Penciled lines and very fine lines do not reproduce well. Remember, the quality of the book cannot be better than the originals provided. Do NOT use tape on your pages!

3.10. Color

The use of color on interior pages (that is, pages other than the cover) is prohibitively expensive. We publish interior pages in color only when it is specifically requested and budgeted for by the conference organizers. DO NOT SUBMIT COLOR IMAGES IN YOUR PAPERS UNLESS SPECIFICALLY INSTRUCTED TO DO SO.

3.11. Symbols

If your word processor or typewriter cannot produce Greek letters, mathematical symbols, or other graphical elements, please use pressure-sensitive (self-adhesive) rub-on symbols or letters (available in most stationery stores, art stores, or graphics shops).

3.12. Copyright forms

You must include your signed IEEE copyright release form when you submit your finished paper. We MUST have this form before your paper can be published in the proceedings.

3.13. Conclusions

Please direct any questions to the production editor in charge of these proceedings at the IEEE Computer Society Press: Phone (714) 821-8380, or Fax (714) 761-1784.

References

[1] I. M. Author. Some related article I wrote. *Some Fine Journal*, 99(7):1–100, January 1999.

[2] A. N. Expert. *A Book He Wrote*. His Publisher, Erewhon, NC, 1999.

[1] Or, better still, try to avoid footnotes altogether. To help your readers, avoid using footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence).