

ZeroMQ Distributed Message Middleware

Uma breve abordagem

Eduardo Pedersetti José¹, Ivan Andreis¹, Leonardo S. Paula¹

¹Ciência da Computação – Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 15.064 – São Leopoldo – RS – Brasil

{eduardoxy, ivann.andreis, leonardopaula}@gmail.com

Abstract. *Abstract text.*

Resumo. *Texto do resumo.*

Middleware

A evolução das redes de computadores com o advento da internet facilitaram a ploriferação de aplicações distribuídas. Sabendo que as partes interessadas de uma aplicação distribuída podem executar em diferentes locais físicos diversas vantagens podem ser acrescidas à aplicações distribuídas, como tolerância à falhas (através de replicação) e aumento de performance através da paralelização de tarefas, por exemplo.

Analisando ambientes onde os Sistemas Distribuídos são executados, nota-se uma heterogeneidade entre os dispositivos que estão se comunicando. Podem existir, em um mesmo Sistema Distribuído, diferentes plataformas de hardware, tecnologias de rede, sistemas operacionais e linguagens de programação. Essas particularidades podem tornar o desenvolvimento de um SD grande desafio (Extraído do Livro Distributed Systems Architecture).

É notável, portanto, que para propiciar o desenvolvimento de uma aplicação distribuída se faz necessária uma infraestrutura adequada, que permita para o desenvolvimento e execução de um SD. Essa infraestrutura necessária é fornecida por um middleware. Um middleware é uma camada de software que reside entre a aplicação e a API de acesso à rede, sendo responsável por abstrair os detalhes de rede que podem ser ignorados pela aplicação.

A rede simplesmente fornece o acesso à camada de transporte, e o acesso à ela difere dependendo da tecnologia e da plataforma física utilizadas. O Middleware homogeniza o acesso às redes, oferecendo serviços genéricos às aplicações. Além de interligar diferentes domínios de tecnologias e encapsular as diferenças entre os sistemas. (Extraído do Livro Distributed Systems Architecture)(Inserir Figura Aqui.) Como o middleware é uma interface entre a API de acesso à rede e a aplicação, ele possui duas visões: a visão do programador da aplicação e a visão do programador do sistema.

O programador da aplicação vê o Middleware como uma ferramenta de auxílio para o desenvolvimento de aplicações distribuídas. Estes programadores não estão interessados como que o Middleware é implementado mas sim em como ele é implementado. O Middleware é uma caixa-preta para o programador da aplicação, com uma interface de acesso bem definida.

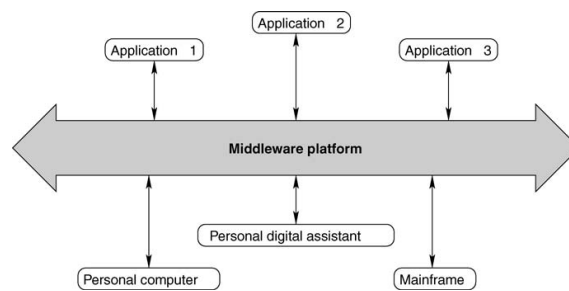


Figure 1. Arquitetura básica de um middleware

O programador de sistemas vai além. Para ele o Middleware é uma caixa-branca e seu interesse maior não está em como o esta caixa interage com o mundo externo, mas sim em como ocorrem os processos internos nela. A aplicação que utilizará o Middleware, para o programador de sistemas, tem uma importância menor. O ponto de referência comum à ambos, programador de sistema e de aplicações, é a interface fornecida pelo middleware que será utilizada pelas aplicações.

Um pouco sobre Message Queue

Descrição sobre message queue. Propósitos, problemas e soluções.

ZeroMQ

A definição formal do ZeroMQ é a seguinte: o ZeroMQ é uma biblioteca de mensagem feita para ajudar os desenvolvedores a projetarem aplicações distribuídas e concorrentes. Embora o ZeroMQ, chamado daqui para frente de ZMQ, seja similar à uma fila de mensagens (ou message queue), é importante deixar claro que ele não é um sistema de fila como o ActiveMQ, WebSphereMQ ou RabbitMQ. O ZMQ é diferente. O ZMQ fornece todas as ferramentas necessárias para que o desenvolvedor desenvolva o seu próprio sistema de fila de mensagens, ou seja, o ZMQ é uma biblioteca. O ZMQ é uma biblioteca desenvolvida nativamente na linguagem C e roda em diversas plataformas. Atualmente ele oferece suporte à mais de vinte linguagens de programação, entre elas estão C/C++, Ruby, PHP, Python, Node.js entre outras.

O ZMQ é uma biblioteca simples. Operações de entrada e saída assíncronas pode enfileirar mensagens em uma thread para tratamento de entrada e saídas. Estas threads trabalham de maneira assíncronas quando tratam o fluxo da rede, deixando o restante do trabalho para ser realizado pelo desenvolvedor. O ZMQ oferece uma interface simplificada para trabalhar com sockets se comparada às diretivas utilizadas na linguagem C, por exemplo.

Histórico

Aplicações

Operação

Concorrentes

Análise

Conclusão

Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

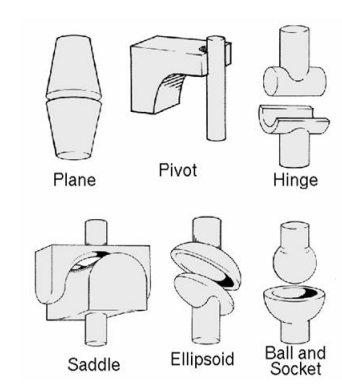


Figure 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 7.

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Table 1. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 \pm 5.22	7.01 \pm 6.84
Selection with in- depth movements	6.29 \pm 4.99	12.22 \pm 11.33
Manipulation with side movements	4.66 \pm 4.94	3.47 \pm 2.20
Manipulation with in- depth movements	5.71 \pm 4.55	5.37 \pm 3.28

References

- Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons Ltd.
- Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.