

# Avaliação de Desempenho do *Middleware* de Código Aberto PORTICO para Simulações Distribuídas de Tempo Real Utilizando HLA

Cap QCO João Rodolfo de Oliveira Rosa e Prof. Dr. Álvaro Luiz Fazenda  
Exército Brasileiro e Universidade Federal de São Paulo

**Resumo** — Este estudo examina o potencial de uso da Infraestrutura de Tempo de Execução (RTI, do inglês *Runtime Infrastructure*) de código aberto, PORTICO, no domínio da simulação distribuída de tempo real. O RTI é um componente central da Arquitetura de Alto Nível, HLA (sigla do inglês *High Level Architecture*), padrão do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) para a integração de simuladores. Foram realizados experimentos a fim de avaliar o desempenho da comunicação em diferentes modos, conforme as especificações da IEEE HLA, e verificar os limites de tempo real. O objetivo foi determinar a viabilidade do PORTICO como RTI de código aberto, considerando suas restrições e comparando-o com um RTI comercial (*pRTI*) sempre que possível. A avaliação focou nos principais serviços definidos na especificação IEEE HLA da Interface de Federado, incluindo a interação entre federados e a atualização de atributos de objetos. Os resultados mostram que o RTI de código aberto avaliado é viável nas condições semelhantes aos testes realizados, levando em conta a composição da federação, ações e parâmetros de configuração. No entanto, alguns dos experimentos visaram testar os limites do RTI, o que pode influenciar a análise da sua viabilidade em situações fora do escopo dos testes.

## I. INTRODUÇÃO

Este estudo avalia a viabilidade do PORTICO como *middleware* para integração de simuladores em tempo real. O RTI é um produto consolidado no mercado, com implementações comerciais e de código aberto sendo o componente central da Arquitetura de Alto Nível (HLA).

Baseado no trabalho de [Moritz Gütlein et al., 2020], foi verificada a robustez do PORTICO, que não incide em custos de licenciamento de uso e assegura domínio tecnológico, comparado com uma solução comercial. O objetivo foi avaliar as potenciais vantagens e desvantagens de uma ferramenta de código aberto em relação a um *software* comercial estabelecido. Os resultados encontrados apontam para a viabilidade do *middleware*.

Embora a tecnologia tenha sido desenvolvida nos anos 90, o estudo da HLA ainda é relevante, pois o emprego de ferramentas de simulação, com o objetivo de reduzir custos e alcançar vantagem competitiva, é um dos principais aspectos da transformação digital das organizações.

## II. ARQUITETURA DE ALTO NÍVEL

- Padrão adotado pelo mercado para integrar simuladores na área militar, mantido pela Organização para Padronização da Interoperabilidade em Simulação (SISO, sigla para *Simulation Interoperability Standards Organization*) e documentado pela IEEE em normas específicas, sendo as principais 1516-2010, 1516.1-2010 e 1516.2-2010. Desenvolvido sob demanda do Departamento de Defesa dos Estados Unidos, possibilita a interoperabilidade entre simuladores e facilita o reuso de seus componentes.
- Seus principais componentes, representados na Figura 1, são: o RTI, que implementa os serviços necessários para a troca de mensagens entre as aplicações integradas; os federados, que são as aplicações integradas (simuladores ou quaisquer outros dispositivos físicos a serem integrados por meio de uma rede de computadores); o FOM, ou Modelo de Objeto da Federação (*Federation Object Model*), documento que padroniza os dados compartilhados entre os federados; e a Federação, ambiente integrado formado pelo RTI, o FOM e os federados.

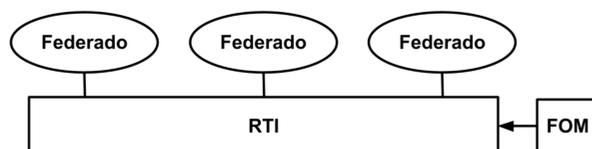


Fig. 1. Topologia Arquitetura de Alto Nível

## III. IMPLEMENTAÇÕES DE RTI: PORTICO E *pRTI*

- PORTICO: software de código aberto e desenvolvido com financiamento do Escritório de Simulação em Defesa Australiano, com início do projeto em 2005. Utiliza a biblioteca *JGROUPS* da linguagem de programação Java permitindo uma implementação descentralizada. Utiliza um arquivo de dados de inicialização do RTI (*RID - RTI Initialization Data*) para configurar seus parâmetros de execução, como o tamanho dos pacotes a serem compartilhados. Não possui uma interface gráfica.
- *pRTI*: desenvolvido e comercializado pela empresa *Pitch Technologies* [Björn Möller et al., 2012] e possui uma estrutura centralizada, com o RTI atuando como nó central. Disponibiliza uma interface gráfica para melhor usabilidade, mas não permite a customização de seus parâmetros de execução. A versão disponível para testes permite a integração de até 2 federados.

## IV. METODOLOGIA DE PESQUISA

- Foram desenvolvidas aplicações Java representando os federados e executando tarefas essenciais para os experimentos, como envio e recebimento de mensagens, atualização do tempo e dos valores de atributos.
- No experimento de escalabilidade, utilizou-se o teste de interação entre os federados, compartilhando um *array* de inteiros com 10 KB de tamanho. Foram realizadas 10.000 iterações, sendo cada iteração uma mensagem enviada. O número de federados foi aumentado de 2 até 16, dobrando esse número a cada rodada de 10 medições.
- Para a estabilidade, atualizou-se o valor do atributo de cada federado. Variou-se o número de iterações de 2.500, 5.000 e 10.000. Verificou-se a média de tempo em cada um dos testes.
- Para os testes foi computado o tempo médio por iteração o qual foi comparado aos valores definidos por [Jakob Nielsen, 1994]. As medições em nanossegundos foram, ao final, convertidas para segundos.
- No comparativo entre o PORTICO x *pRTI* foram 10 rodadas de 2.500 iterações, sendo cada iteração uma de troca de mensagens (um *array* de inteiros com 10 KB) entre os federados.
- Toda a comunicação entre os federados ocorreu de forma *broadcast*.

- Cada federado utilizou o sistema operacional Debian 11 em uma máquina virtual com 4 GB de RAM, gerenciadas utilizando a ferramenta *Xen Orchestra*. A virtualização foi realizada com o XCP NG Hypervisor. O hardware utilizado para o cenário foi um *Intel® Xeon® Silver 4114 CPU @ 2.20 GHz*. Parte do monitoramento dos testes foi feito com as ferramentas *top* e *bmon* para sistemas Linux. Foram utilizadas a versão 5.4.5.0 build 3195 do *pRTI Free* e a versão 2.1.0 do PORTICO.
- Os parâmetros de tempo seguidos foram: 0,10 s como limite para a percepção de tempo real pelo usuário e 1,00 s como limite para que, mesmo com uma percepção de atraso por parte do usuário, o sistema seja percebido como um sistema de tempo real, conforme definido por [Jakob Nielsen, 1994].

## V. RESULTADOS E DISCUSSÃO

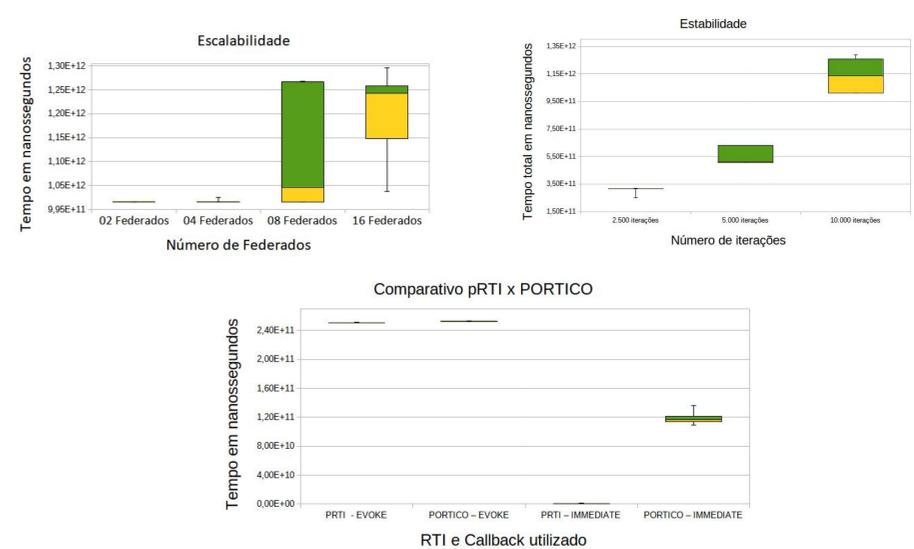


Fig. 2. Resultados dos experimentos de escalabilidade, estabilidade e comparativo entre PORTICO e *pRTI*.

- Para escalabilidade, observou-se que, até 4 federados, os valores foram regulares, com intervalo interquartilício estável. Com 8 e 16 federados, a variância aumentou, com desvio positivo para 8 federados e negativo para 16, mas ainda próximo de 0,10 s, conforme definido por [Jakob Nielsen 1994]. Para a estabilidade, a mediana foi próxima de 0,10 s para até 8 federados, com intervalo interquartilício reduzido, e aumentou a variância com 16 federados, mantendo a mediana próxima de 0,10 s.
- Considerando diferentes modelos de chamada, *EVOKE* e *IMMEDIATE*, não houve diferença relevante entre os dois RTI no modelo *EVOKE*. No entanto, o *pRTI* apresentou melhores resultados para o modelo de chamada *IMMEDIATE*, com tempo computado próximo de zero.

## VI. CONCLUSÃO

- Os resultados obtidos apontam para a viabilidade de um RTI de código aberto para simulações que envolvam limites de tempo real próximos aos definidos por [Jakob Nielsen, 1994].
- Além da motivação técnica ao utilizar um RTI de código aberto, também deve ser levado em conta fatores como a redução de custos e a independência de fornecedores.

## REFERÊNCIAS

1. M. Gütlein, W. Baron, C. Renner and A. Djanatliev, "Performance Evaluation of HLA RTI Implementations", 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Prague, Czech Republic, 2020, pp. 1-8, doi: 10.1109/DS-RT50469.2020.9213641.
2. J. Nielsen. 1994. "Usability Engineering". Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
3. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)–Federate Interface Specification," in IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000) , vol., no., pp.1-378, 18 Aug. 2010, doi: 10.1109/IEEESTD.2010.5557728.
4. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules," in IEEE Std 1516-2000 , vol., no., pp.1-28, 11 Dec. 2000, doi: 10.1109/IEEESTD.2000.92296.
5. T. Pokorny, "Portico - Runtime Infrastructure", Maio/ 2016. Disponível em <<https://github.com/openlvc/portico>>. Acesso em 31/08/2024.
6. B. Möller et al. "The HLA Tutorial. A Practical Guide For Developing Distributed Simulations", 2012. Disponível em: <<https://pitchtechnologies.com/hlatutorial/>>, Acesso em 31/08/2024.