

# Uma Proposta de Monitoramento Hierárquico Utilizando Heartbeat em Sistemas de Computação de Alto Desempenho

Cleverson P. Silva<sup>1</sup>, Gustavo T. Santos<sup>1</sup>, João V. S. Mota<sup>1</sup>, Calebe P. Bianchini<sup>1</sup>

<sup>1</sup>Faculdade de Computação e Informática (FCI)  
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brasil

{10391119, 10391083, 10416614}@mackenzista.com.br

calebe.bianchini@mackenzie.br

**Resumo.** *Sistemas de Computação de Alto Desempenho (HPC) são fundamentais para aplicações científicas e industriais de larga escala, exigindo alta disponibilidade e tolerância a falhas. A biblioteca DeLIA oferece suporte a essas aplicações por meio de mecanismos como heartbeat, checkpoint e restart. Contudo, sua arquitetura atual adota um processo centralizado de monitoramento, que representa um ponto único de falha. Este artigo propõe uma nova arquitetura para os processos da biblioteca DeLIA, baseada em uma topologia em anel com heartbeat, eliminando a centralização e distribuindo as responsabilidades de monitoramento entre múltiplos nós.*

## 1. Introdução

Os sistemas de Computação de Alto Desempenho (HPC - *High-Performance Computing*) tornaram-se essenciais para o avanço de diversas áreas científicas e industriais, pois oferecem capacidade computacional significativamente superior à de servidores e computadores convencionais, viabilizando o processamento de grandes volumes de dados [Silva et al. 2022]. No entanto, devido à criticidade dessas aplicações, é imprescindível que esses sistemas apresentem alta disponibilidade e mecanismos robustos de tolerância a falhas, assegurando a continuidade das operações mesmo sob condições adversas [Netto et al. 2018].

Um dos principais desafios dos ambientes HPC é garantir a resiliência a falhas, ou seja, a capacidade de manter o funcionamento correto diante de problemas inesperados, como falhas de hardware, software ou conectividade [Chetan et al. 2005]. Para mitigar esses riscos, são empregadas arquiteturas tolerantes a falhas e técnicas de recuperação, como os mecanismos de checkpoint/restart e replicação de dados, que minimizam interrupções e preservam a integridade das execuções.

Neste contexto, este trabalho propõe uma arquitetura de monitoramento distribuído baseada em topologia de anel para a biblioteca DeLIA [Santana et al. 2024], que é uma biblioteca que implementa mecanismos de confiabilidade. Diferentemente da abordagem tradicional centralizada atualmente implementada na biblioteca DeLIA, a topologia em anel distribui as responsabilidades de supervisão entre múltiplos nós, eliminando pontos únicos de falha. Essa proposta visa aprimorar a resiliência do sistema, reduzindo a sobrecarga de comunicação e garantindo maior continuidade operacional em ambientes HPC críticos.

## 2. DeLIA

DeLIA<sup>1</sup> (*Dependability Library for Iterative Applications*) é uma biblioteca projetada para oferecer tolerância a falhas em aplicações paralelas iterativas que utilizam computação de alto desempenho. Ele permite a detecção de interrupções e a recuperação do estado global e local dos processos por meio de técnicas como *checkpoint/restart* e *rollback*. Desenvolvido para atender às necessidades de programas que sincronizam dados entre processos a cada iteração, a DeLIA combina flexibilidade e eficiência, permitindo configurar parâmetros como frequência de *checkpoints* e seleção de dados a serem persistidos. A biblioteca também é capaz de lidar com cenários de falhas abruptas (*fail-stop*) e interrupções *preemptivas*, garantindo a continuidade das operações com impacto mínimo no desempenho [Santana et al. 2024].

O principal objetivo da biblioteca DeLIA é abordar falhas do tipo *fail-stop*, caracterizadas por interrupções na execução de aplicações devido a falhas de hardware ou a problemas específicos de software, conforme discutido por [Herault and Robert 2015]. Essas falhas são particularmente críticas, pois resultam na interrupção abrupta da execução do sistema, comprometendo a integridade e a continuidade do processamento. Além disso, o DeLIA também foi concebido para tratar de cenários *preemptivos*, que correspondem a interrupções na execução das aplicações, mesmo quando estas não são provocadas diretamente por falhas técnicas.

Por meio da oferta de mecanismos de tolerância a falhas que englobam tanto interrupções *fail-stop* quanto cenários *preemptivos*, a DeLIA proporciona maior resiliência às aplicações baseadas no modelo BSP (*Bulk Synchronous Parallel*). Esses mecanismos permitem a detecção precoce de falhas e a recuperação eficiente, garantindo que a continuidade das operações seja preservada e que o estado global ou local da aplicação seja mantido, minimizando perdas no processo.

A DeLIA apresenta uma interface de programação de aplicativos (API) que viabiliza a incorporação de recursos avançados de tolerância a falhas no desenvolvimento de software. Por meio dessa API, os desenvolvedores podem invocar funções específicas da DeLIA, abstrair a complexidade inerente à implementação da biblioteca e configurar os principais parâmetros de funcionamento em um arquivo no formato JSON.

## 3. Topologia de anel

A estrutura de anel em sistemas distribuídos é uma topologia em que os nós são organizados de forma que cada nó tenha exatamente dois vizinhos: um anterior e um posterior, formando um anel fechado. Nessa configuração, as mensagens e os dados podem ser encaminhados de um nó para o próximo em uma direção definida (ou, em alguns casos, em ambas as direções), garantindo que as informações circulem por todo o sistema. Essa estrutura oferece simplicidade na comunicação e evita a sobrecarga de conexão centralizada [Tanenbaum and Steen 2007].

Especificamente, uma estrutura de anel em um sistema P2P (*peer-to-peer*) estruturado organiza os nós de forma lógica em um anel, com cada nó responsável por um intervalo específico de chaves ou dados. Esse arranjo facilita a busca eficiente e determinística de dados, com cada nó armazenando informações de roteamento que garantem

---

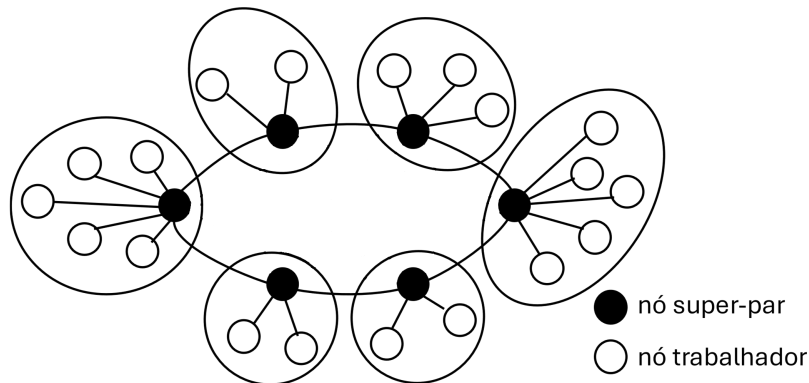
<sup>1</sup>O DeLIA pode ser obtido em <https://gitlab.com/lappsufrn/delia>

que as mensagens sejam encaminhadas de forma otimizada. Em uma variação com super-pares (*superpeers*), alguns nós no anel são promovidos a super-pares, que têm maiores responsabilidades, como o gerenciamento de grupos de nós regulares. Esses super-pares podem agir como pontos centrais para facilitar o roteamento e a organização, melhorando o desempenho geral do sistema e fornecendo maior robustez e eficiência na gestão de grandes volumes de dados e requisições [Tanenbaum and Steen 2007].

#### 4. Desenvolvimento do projeto

A versão atual da biblioteca DeLIA apresenta uma limitação importante relacionada à centralização do mecanismo de monitoramento baseado em *heartbeat*, no qual um único nó é encarregado da supervisão de todo o sistema. Essa arquitetura centralizada representa um ponto único de falha: caso o nó responsável pelo monitoramento sofra interrupções, a detecção de falhas é comprometida, afetando a continuidade das operações do sistema como um todo [Santana et al. 2024].

Para superar essa limitação, propomos o desenvolvimento de uma arquitetura hierárquica de monitoramento estruturada em uma topologia de anel com super-pares. Nessa abordagem, um subconjunto de processos do ecossistema DeLIA assume responsabilidades de monitoramento mútuo por meio do mecanismo de *heartbeat*, promovendo uma supervisão descentralizada. A Figura 1 ilustra a topologia proposta, evidenciando a organização dos nós e a atuação dos super-pares no anel.



**Figura 1. Proposta de topologia em anel para o monitoramento *heartbeat***

Além disso, a seleção dos processos que atuarão como super-pares será feita de forma dinâmica com base em uma função logarítmica, de acordo com a equação  $f(x) = \log_2(x)$ , onde  $x$  representa o número total de nós. Essa estratégia permite promover de maneira eficiente um número reduzido de nós para funções supervisoras, mantendo o equilíbrio entre robustez e sobrecarga.

Essa nova arquitetura proporciona maior tolerância a falhas, pois garante que, mesmo na ocorrência de interrupções em nós supervisores, outros processos do anel sejam capazes de detectar a falha e ativar os mecanismos de recuperação da DeLIA. Dessa forma, assegura-se a continuidade das operações com impacto mínimo, reforçando a resiliência e a estabilidade do sistema [Santana et al. 2024]. A literatura indica ainda que essa abordagem reduz significativamente a quantidade de mensagens trocadas entre os processos, diminuindo a sobrecarga de comunicação e promovendo escalabilidade [Tanenbaum and Steen 2007].

## 5. Discussão e próximos passos

Este trabalho apresentou uma proposta de aprimoramento da biblioteca DeLIA por meio da implementação de uma arquitetura hierárquica de monitoramento baseada em topologia de anel com super-pares. A abordagem busca resolver uma limitação crítica da versão original da DeLIA, que utiliza um modelo centralizado de supervisão por meio de mensagens *heartbeat* sujeito a ponto único de falha.

Distribuindo o monitoramento e promovendo supervisores segundo uma função logarítmica, a arquitetura proposta aumenta a tolerância a falhas e assegura continuidade operacional, enquanto as mensagens de heartbeat, antes concentradas entre o nó principal e todos os processos, passam a ser repartidas entre os  $X$  nós supervisores, reduzindo a sobrecarga de comunicação e potencialmente melhorando a escalabilidade do sistema.

A proposta demonstra potencial para fortalecer a resiliência de aplicações paralelas iterativas em ambientes de HPC. Ainda pretende-se realizar experimentos com implementações reais da arquitetura proposta, utilizando-se de métricas como latência de detecção de falhas, tempo de recuperação e impacto no desempenho geral da aplicação, dentre outras que a literatura propõe e que também foram usadas para avaliação da própria biblioteca DeLIA [Tanenbaum and Steen 2007][Santana et al. 2024].

## Agradecimentos

Os autores agradecem o apoio da MackCloud<sup>2</sup>; e do projeto SPRACE – Processo nº 2018/25225-9, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP). Este trabalho foi financiado em parte pelo Fundo Mackenzie de Pesquisa e Inovação (MackPesquisa) – Projetos nros 231009 e 251005.

## Referências

- Chetan, S., Ranganathan, A., and Campbell, R. (2005). Towards fault tolerance pervasive computing. *IEEE Technology and Society Magazine*, 24(1):38–44.
- Herault, T. and Robert, Y. (2015). *Fault-Tolerance Techniques for High-Performance Computing*. Springer Publishing Company, Incorporated, 1st edition.
- Netto, M. A. S., Calheiros, R. N., Rodrigues, E. R., Cunha, R. L. F., and Buyya, R. (2018). Hpc cloud for scientific and business applications: Taxonomy, vision, and research challenges. *ACM Computing Surveys*, 51(1):1–29.
- Santana, C., Araújo, R. C., Sardina, I. M., Ítalo A.S. Assis, Barros, T., Bianchini, C. P., de S. Oliveira, A. D., de Araújo, J. M., Chauris, H., Tadonki, C., and de Souza, S. X. (2024). Delia: A dependability library for iterative applications applied to parallel geophysical problems. *Computers & Geosciences*, 191:105662.
- Silva, G. P., Bianchini, C. P., and Costa, E. B. (2022). *Programação Paralela e Distribuída com MPI, OpenMP e OpenACC para computação de alto desempenho*. CasaDoCodigo.
- Tanenbaum, A. S. and Steen, M. V. (2007). *Distributed Systems: Principles and Paradigms*. Pearson - Prentice Hall, 2 edition.

---

<sup>2</sup>Laboratório Multidisciplinar de Computação Científica e Nuvem, em <https://mackcloud.mackenzie.br>