

Synapse meets Slurm: Proposta de um middleware para paralelização de algoritmos de otimização populacionais

Arthur M. P. Gabardo¹, Pablo A. Jaskowiak¹, Thiago P. Tancredi^{1,2}

¹Núcleo de Simulação e Otimização de Sistemas Dinâmicos (NSO)
Universidade Federal de Santa Catarina (UFSC) – Joinville, SC

²Laboratório de Simulação Naval (LaSiN) – Centro Tecnológico de Joinville
Universidade Federal de Santa Catarina (UFSC) – Joinville, SC

arthur.miguel@grad.ufsc.br, {pablo.andretta, thiago.tancredi}@ufsc.br

Resumo. *A Otimização Multidisciplinar possui um papel central na integração de áreas diversas em projetos de engenharia. Recursos computacionais podem, porém, ser um fator limitante em processos de otimização. Este artigo apresenta um middleware desenvolvido para integrar o software Synapse a clusters gerenciados pelo Slurm Workload Manager. O middleware facilita a execução de algoritmos de otimização populacionais (e.g., algoritmos genéticos) de maneira distribuída, favorecendo processos de otimização. Testes em um cluster heterogêneo permitiram validar a solução. Em experimentos preliminares, a solução desenvolvida apresentou speedups de até dez vezes em relação ao uso de workstations, abordagem até então suportada pelo Synapse.*

1. Introdução

Projetos de engenharia de sistemas complexos destacam-se por sua transversalidade, exigindo a integração de diversas áreas. Neste contexto, surge como elemento central a Otimização Multidisciplinar (MDO - *Multi-disciplinary Design Optimization*), proporcionando uma abordagem holística e integrada que transcende fronteiras tradicionais de análise, buscando o aprimoramento global de novos projetos [Caixeta 2011].

Um processo de otimização pode ser compreendido como a busca automática, sistemática e dirigida para obter uma solução de qualidade aceitável em um domínio de soluções [Martins and Ning 2021]. Seus dois componentes principais são: (i) uma ferramenta de análise, que fornece um modelo numérico/matemático do problema, permitindo mensurar a qualidade de uma solução e; (ii) o próprio algoritmo de otimização, responsável pela busca [Tancredi 2009]. É importante ressaltar que a caracterização de uma solução como aceitável irá depender de critérios do projeto (e.g. peso, custo, tempo). Além disso, o processo de busca é limitado por recursos computacionais e não há, em muitos casos, garantia de otimalidade [Martins and Ning 2021].

Neste contexto, o software *Synapse Engenharia Multidisciplinar* [Synapse 2023] proporciona um ambiente colaborativo e amigável ao usuário, viabilizando a otimização de projetos de engenharia mono e multiobjetivos. O Synapse possibilita integração com diversas ferramentas de planejamento de experimentos (DoE - *Design of Experiments*) e análise como AnSYS APDL e MATLAB. Dentre os métodos de otimização disponíveis no software, são de interesse deste trabalho os meta-heurísticos populacionais (e.g., algoritmos genéticos), visto que podem se beneficiar significativamente de processamento paralelo/distribuído, ao permitir que cada indivíduo (solução) seja avaliado isoladamente.

Apesar de empregar processamento paralelo, o Synapse não oferece suporte para processamento distribuído¹. O presente trabalho propõe o desenvolvimento e a validação de um *middleware* que permita ao Synapse tirar proveito de sistemas de processamento distribuído de alto desempenho (HPC – *High Performance Computing*) em *clusters* geridos com Slurm [Yoo et al. 2003]. Após alterações mínimas no Synapse, usuários agora podem lançar processos de otimização populacionais em que os indivíduos (soluções) são avaliados em um cluster, caso disponível. Tal funcionalidade permite o aumento da taxa de soluções examinadas, possivelmente beneficiando o processo de otimização.

2. Proposta

O software Synapse é desenvolvido em Delphi, com foco em Sistemas Operacionais Windows. Portar o software para Sistemas Operacionais GNU/Linux a fim de suportar processamento distribuído em clusters geridos pelo Slurm demandaria uma quantidade substancial de tempo, dado o nível de maturidade atual do software, que disponibiliza vários recursos (algoritmos/métodos) de otimização e interface gráfica amigável.

Para minimizar alterações no Synapse e ao mesmo tempo dar suporte a processamento em clusters, um *middleware* foi concebido/desenvolvido. A implementação usa *bourne again shell* (bash), em sua maior parte, com uma pequena porção de código desenvolvida em PowerShell (rotinas de autenticação, comunicação e transmissão de dados entre o host onde o Synapse está em execução e o cluster). A Figura 1 apresenta um diagrama de sequência da arquitetura proposta para o *middleware*. Toda comunicação entre Synapse e cluster se dá via *ssh* e é, a partir de então, interpretada pelo *middleware*.

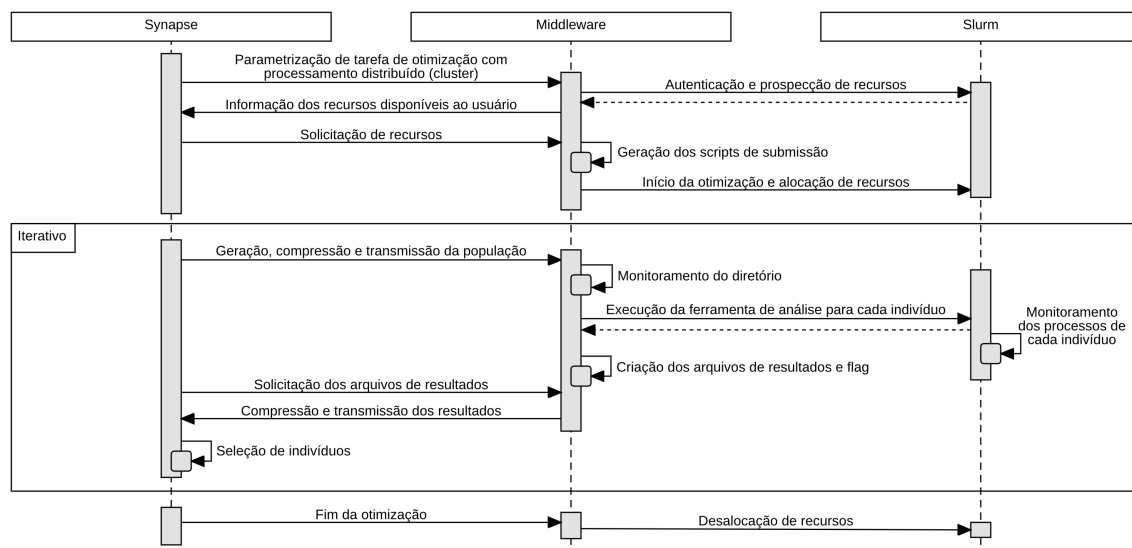


Figura 1. Diagrama de sequência do middleware proposto.

De forma breve, ao início de um processo de otimização, o *middleware* realiza a autenticação no cluster, o qual informa os recursos disponíveis ao usuário. Em seguida um *job* é submetido pelo Synapse ao Slurm. O *job* do *middleware* permanece em

¹Processamento paralelo refere-se à execução de múltiplos processos simultâneos conectados por um espaço de memória compartilhado. Processamento distribuído é caracterizado por um agrupamento de múltiplos computadores em rede, permitindo a execução ubíqua de tarefas [Bertsekas and Tsitsiklis 2015].

execução até o fim do processo de otimização usando os recursos requisitados de maneira exclusiva, sendo responsável por: (i) receber arquivos contendo indivíduos (soluções) que devem ser avaliados; (ii) alocar recursos adicionais e distribuir os indivíduos de acordo com a quantidade de recursos solicitada; (iii) monitorar e gerenciar uma fila interna ao middleware de indivíduos, realizando balanceamento de carga; (iv) compactar arquivos de resultado de uma geração do algoritmo genético e escrever arquivos de *flag*, indicando que os resultados estão prontos e que o Synapse pode transferi-los a partir do cluster.

Os scripts de submissão em lote são gerados automaticamente para cada um dos nós solicitados, e a requisição de alocação dos recursos é feita ao Slurm por meio do comando `sbatch`. Como a alocação é feita individualmente para cada nó, a análise dos indivíduos pode ocorrer mesmo com um número de nós disponíveis inferior ao requerido. Após a alocação dos nós, um diretório no caminho especificado pelo usuário é criado para armazenar os dados necessários para as simulações e seus resultados. Os jobs do middleware em execução nos nós de computação monitoram constantemente as mudanças realizadas no diretório e informam a quantidade de núcleos de processamento disponíveis.

Os indivíduos de uma população são gerados no computador do usuário por um dos métodos de DoE disponíveis no Synapse. Estes são então compactados e transmitidos para o diretório da simulação no cluster via `scp`, juntamente com um identificador do número da geração. A chegada deste arquivo ao cluster indica que há uma nova geração a ser avaliada. Após a extração dos indivíduos o arquivo `file_list` é criado, contendo em cada uma de suas linhas, uma fila de indivíduos a ser avaliada em cada nó. A geração e balanceamento de carga destas filas são realizados pelo middleware. Como o custo computacional da avaliação dos indivíduos em uma mesma geração é similar, a população é dividida de maneira proporcional ao número de núcleos de cada um dos nós. Os processos são distribuídos e executados como determinado no arquivo `file_list`, a quantidade de processos simultâneos é monitorada por meio da ferramenta `process status` (`ps`). Quando o tamanho da população é inferior ao número de núcleos, múltiplos threads podem ser usados para um mesmo indivíduo, caso haja suporte por parte da ferramenta de análise.

Completada a simulação de todos os indivíduos, um arquivo de flag é criado. O Synapse realiza requisições periódicas para verificar a existência deste arquivo, assim que identificado, os resultados são compactados e transmitidos para o host do Synapse, onde são comparados e selecionados. A seguir, uma nova população é gerada, e o procedimento descrito ocorre de maneira iterativa até que o espaço de busca seja esgotado, um conjunto de soluções ótimas encontrado e o processo de otimização seja encerrado. Por fim, um arquivo sinalizando o término da otimização encerra os trabalhos alocados no cluster.

3. Resultados e Discussão

O middleware proposto foi implementado e integrado à última versão do *Synapse Offshore* (Figura 2), uma customização do Synapse destinada à otimização de sistemas de exploração de petróleo. Os testes e desenvolvimento foram realizados em um cluster heterogêneo com quatro nós de computação² e validaram a arquitetura proposta, que tem sido utilizada com sucesso nos projetos de pesquisa do Laboratório de Simulação Naval (LaSiN) da UFSC. Em uma comparação preliminar entre uma workstation com processador I9 12900K e o cluster mencionado, observou-se *speedups* de até dez vezes. Esse

²Três nós com 2 processadores Xeon E5-2630v3 e um nó com 2 processadores Xeon E5-2670v3.

valor é apenas uma estimativa empírica considerando os sistemas descritos. Experimentos futuros buscarão obter um valor de speedup seguindo o rigor estatístico apropriado.

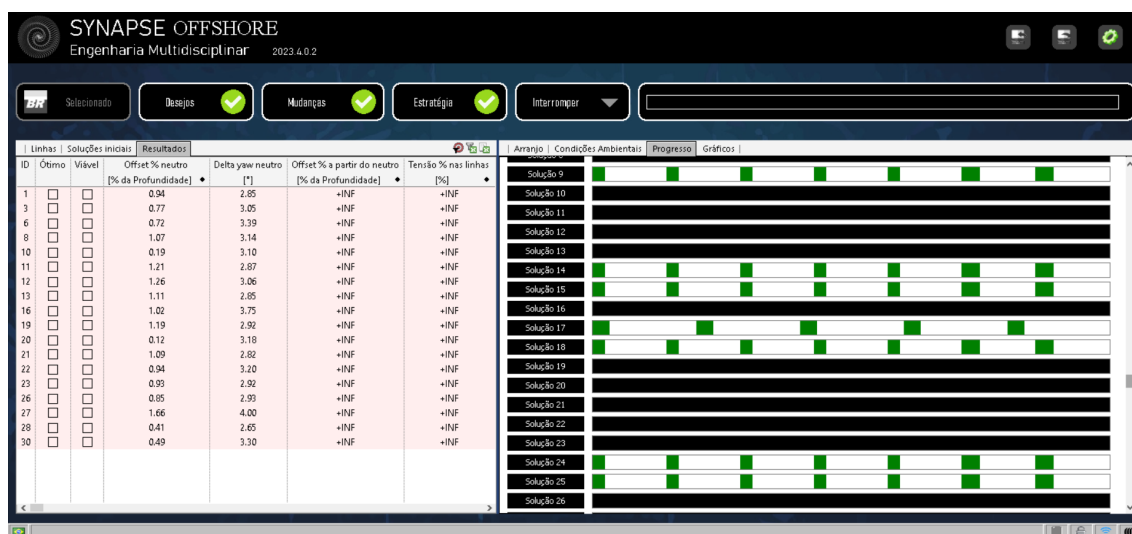


Figura 2. Tela de monitoramento do processo de otimização.

4. Conclusões

Os resultados obtidos demonstraram a viabilidade e vantagens do processamento distribuído de processos de otimização populacionais usando o Synapse em clusters gerenciados pelo Slurm. O middleware proposto possibilitou a execução de tarefas de simulação remotamente. Em trabalhos futuros, outras etapas do processo de otimização, como o planejamento de experimentos, seleção e mutação de indivíduos, também devem ser migradas para o cluster explorando vantagens do processamento distribuído e paralelo.

Agradecimentos

Os autores agradecem à Petrobras S.A. pelo apoio financeiro ao projeto.

Referências

- Bertsekas, D. and Tsitsiklis, J. (2015). *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont.
- Caixeta, P. R. (2011). *Otimização multidisciplinar em projeto de asas flexíveis utilizando metamodelos*. Doutorado em Engenharia Mecânica, EESC, USP, São Carlos.
- Martins, J. R. R. A. and Ning, A. (2021). *Engineering Design Optimization*. Cambridge University Press, Cambridge.
- Synapse (2023). Documentação Synapse Engenharia Multidisciplinar. Disponível em: <https://synapse.ufsc.br/>. Acesso em: 17 jan. 2024.
- Tancredi, T. P. (2009). *Otimização multidisciplinar distribuída aplicada a projetos de engenharia*. Doutorado em Engenharia Naval e Oceânica, POLI, USP, São Paulo.
- Yoo, A. B. et al. (2003). SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, pages 44–60, Springer. Berlin.