Comparação entre Sistemas Gerenciadores de Recursos em um ambiente de alto desempenho utilizando uma aplicação científica

Gabriela Luisa Eckel[1], **Fernando Emilio Puntel**[2], **Adriano Petry**[1]

[1]Centro Regional Sul de Pesquisas Espaciais
Instituto Nacional de Pesquisas Espaciais
Santa Maria, RS, Brasil

[2]Colégio Politécnico da UFSM
Universidade Federal de Santa Maria
Santa Maria, RS, Brasil

gleckel@inf.ufsm.br, fernandopuntel@gmail.com, adriano.petry@inpe.br

Resumo. *Com o avanço da tecnologia e sua dependência no cotidiano, a computação de alto desempenho (HPC) passou a ser uma das principais áreas de pesquisa e investigação na computação. Os ambientes de alto desempenho são comumente utilizados em aplicações que necessitam processar uma variedade de informações e obter o resultado em um tempo limitado para a aplicação. Por conta disso, os ambientes de alto desempenho devem trabalhar de forma eficiente para atender as demandas da aplicação e obter os resultados de forma rápida. Para que um ambiente de alto desempenho trabalhe de forma otimizada, é fundamental a escolha de um sistema gerenciador de recursos (SGR) eficiente que atenda as necessidades do da aplicação em questão. Por esse motivo, o objetivo deste trabalho foi avaliar e comparar a performance de dois SGRs - SLURM e OAR - em uma aplicação científica real de previsão ionosférica, a fim de verificar qual deles atende melhor as necessidades e requisitos para essa aplicação. Os experimentos foram realizados em um cluster de uso dedicado localizando no Centro Regional Sul do INPE em Santa Maria, e possui 5 nós de processamento e um de controle. Foram realizados experimentos para quatro dias diferentes, e após avaliadas estatísticas referentes a taxa média de uso de CPU, memória e tempo total de execução. O SLURM apresentou melhores resultados na grande maioria das avaliações, incluindo um menor tempo total de processamento para os quatro dias simulados, quando comparado à solução com o OAR.*

1. Introdução

Atualmente, a HPC (computação de alto desempenho) oferece recursos computacionais necessários para aplicações que apresentam alta demanda de recursos computacionais, em áreas como mercado financeiro, previsão de tempo e simulação de fluídos físicos. O grande poder de processamento disponibilizado por ambientes de alto desempenho garante que o tempo necessário para obter os mais diversos resultados seja reduzido de maneira exponencial [Prabhu 2008].

O cluster é um exemplo de arquitetura de alto desempenho que fornece alta capacidade de processamento através da interconexão de computadores, normalmente chamados de nós, por meio de uma rede de alta velocidade, visando a distribuição do processamento requerido para solução de problemas, que na maioria das vezes não seria possível usando um computador convencional [Prabhu 2008]. Para [Sloan 2004], o cluster possui três componentes essenciais: uma coleção de computadores, uma rede de alta velocidade que interconecta os nós e um software que possibilita aos computadores compartilhar a execução de *job*s com outros nós.

O SGR (Sistema Gerenciador de Recursos) é responsável por realizar o gerenciamento dos *job*s, usuários e dos recursos computacionais distribuídos em um cluster, a fim de obter os resultados de forma rápida e eficiente. A escolha de um SGR é fundamental para otimização do uso dos recursos disponíveis, e alternativas como o SLURM e OAR são muito difundidas nestes ambientes. [Nicolas et al. 2016] [Yoo et al. 2003]

Este estudo busca avaliar e comparar os SGRs SLURM e OAR em uma aplicação científica real de previsão ionosférica, realizada diariamente no Centro Regional Sul de Pesquisas Espaciais do Instituto Nacional de Pesquisas Espaciais (CRCRS/INPE). A aplicação gera mapas de TEC (Conteúdo Eletrônico Total) para a região da América do Sul [Petry et al. 2014], que são disponibilizados em *http://www2.inpe.br/climaespacial/portal/tec-supim-previsao/*.

2. Sistemas Gerenciadores de Recursos

2.1. SLURM

SLURM (Slurm Workload Manager) é um sistema de gerenciamento de cluster e agendamento de tarefas de código aberto, tolerante a falhas e altamente escalável para ambientes de alto desempenho computacional. Atualmente o SLURM é utilizado em 60% dos 500 melhores supercomputadores do mundo [Gvozdetska et al. 2019]. SLURM não requer modificações no kernel para sua operação e é relativamente independente. Como gerenciador de carga de trabalho em cluster, SLURM tem três funções principais. Inicialmente, aloca acesso exclusivo e/ou não exclusivo aos recursos dos nós de computação para os usuários por um período de tempo para que eles possam executar o *job*. Após isso, o SLURM fornece uma estrutura para iniciar, executar e monitorar o *job* (normalmente um *job* paralelo) no conjunto de nós alocados. Finalmente, arbitra a disputa por recursos, gerenciando uma fila de *job*s pendentes. [Yoo et al. 2003]

2.2. OAR

OAR é um gerenciador versátil de recursos e tarefas (também chamado de agendador de lotes) para clusters de HPC e outras infraestruturas de computação, como bancos de teste experimentais de computação distribuída. Foi desenvolvido no Instituto Politécnico Nacional de Grenoble, na França. A filosofia do projeto é que seja possível desenvolver um sistema complexo para o gerenciamento de recursos, sem que a eficiência e escalabilidade seja comprometida [Nicolas et al. 2016]. É composto por módulos que interagem principalmente via banco de dados e são executados como programas independentes. Portanto, formalmente, não há API (Interface de Programação de Aplicativos), a interação do sistema é completamente definida pelo esquema do banco de dados. Essa abordagem facilita o desenvolvimento de módulos específicos. De fato, cada módulo (como escalonadores)

pode ser desenvolvido em qualquer idioma com uma biblioteca de acesso ao banco de dados [Nicolas et al. 2016].

3. Sistema de Previsão Ionosférica

A ionosfera é a camada da atmosfera na qual existem elétrons livres e íons eletricamente carregados. Essa camada inicia em torno de 90 km e pode facilmente ultrapassar 1000 km de altitude da superfície da Terra. O sistema de previsão de dinâmica da ionosfera foi desenvolvido e é executado diariamente, com uma previsão de 24 horas à frente. O conteúdo total de elétrons na ionosfera interfere nos dados de posicionamento de sistemas globais de navegação por satélites (GNSS), portanto as simulações são de grande importância na prevenção de erros de posicionamento. A previsão baseia-se em uma modelagem físico-química da região, e sua interação com o campo magnético terrestre. Esse modelo matemático, escrito em linguagem Fortran, foi inicialmente desenvolvido na Universidade de Sheffield do Reino Unido e chamado SUPIM (*Sheffield University Plasmasphere-Ionosphere Model*), mas vem sendo substancialmente melhorado nos últimos anos. Além da execução do modelo matemático, também é preciso realizar um pré e pós-processamentos para realização da previsão ionosférica. O pré-processamento é responsável, dentre outras coisas, pela obtenção de dados necessários para realização da previsão ionosférica. Já o subsistema de pós-processamento, chamado DAVS (*Data Assimilation and Visualization System*), foi desenvolvido no INPE em linguagem C++, e é focado na interpolação dos dados oriundos do SUPIM em uma grade tridimensional homogeneamente espaçada [Petry et al. 2014].

4. Experimentos

4.1. Metodologia

Os experimentos foram realizados em um cluster de uso dedicado localizado no Centro Regional Sul de Pesquisas Espaciais do Instituto Nacional de Pesquisas Espaciais (CRCRS/INPE). O cluster utilizado possui 5 nós de processamento e um nó de controle. Cada nó de processamento possui 8 CPUs Intel Xeon 2.40 GHz, com 74 GB de memória RAM, sistema operacional CentOS 7.7 e os Sistemas Gerenciadores de Recursos SLURM e OAR instalados.

Para avaliação da performance dos SGRs foi escolhida a região da América do Sul para realização das simulações ionosféricas. A avaliação dos experimentos foi realizada separadamente para a execução do modelo SUPIM, que utiliza predominantemente recursos para cálculos matemático, e o subsistema DAVS, onde o processamento matemático e uso de memória são mais equilibrados. Para região escolhida, o SUPIM executa em 56 *job*s paralelos numa simulação, onde cada *job* representa uma longitude diferente da região escolhida (-35ºW até -85ºW). O DAVS executa 24 processos, correspondendo a cada hora simulada ao longo de um dia de previsão ionosférica. Todas as rodadas, tanto do SUPIM quanto do DAVS, foram executados de forma exclusiva em todos os nós do cluster, garantindo assim uma maior confiabilidade nos resultados [Puntel 2019].

A Tabela 1 apresenta os requisitos de recursos computacionais do SUPIM e DAVS quando utilizados os SGRs SLURM e OAR. É importante destacar que a requisição de 0.5 de CPU do DAVS quando utilizado o SLURM deve-se ao fato de que os *job*s DAVS não necessitam de tanto poder de processamento e com isso foi calibrado até uma requisição

Tabela 1. Requisitos para execução do SUPIM e do DAVS

job	CPU	Memória (GB)
SLURM		
SUPIM	1	5
DAVS	0.5	6
OAR		
SUPIM	1	0
DAVS	1	6

que não iria prejudicar as execuções do DAVS, submetendo assim mais de um *job* por CPU. Esse ajuste foi possível após alterações no arquivo de configuração do SLURM. Já para o OAR não é possível realizar este tipo de configuração. A requisição de memória do SUPIM, quando utilizado OAR, foi zerada pois o banco de dados de recursos computacionais foi configurado para que todas as requisições fossem através de CPU, definida automaticamente pelo próprio OAR.

A fim de considerar a variabilidade nas previsões ionosféricas, uma vez que dependendo do dia simulado pode haver diferença das concentrações eletrônicas obtidas, o que pode levar a um maior ou menor tempo de processamento para convergência da simulação, foram escolhidos quatro dias diferentes para avaliação. O equinócio de outono (20 de março) e o solstício de inverno (21 de junho) para o hemisfério Sul no ano de 2019 foram os balizadores para definição das datas a serem utilizadas nas simulações, já que a variabilidade sazonal da ionosfera é conhecida e fortemente dependente da distância entre a Terra e o Sol, que afeta a quantidade de radiação ionizante que atinge a ionosfera. Além dessas datas, também foram escolhidos outros dois dias para simulação, que cobrissem o período entre tal equinócio e solstício, e que fossem dias equidistantes dessas datas balizadoras. Assim, os dias simulados foram: 20/03, 20/04, 21/05, 21/06, todos para o ano de 2019. Foram realizadas 15 simulações para cada um desses dias e para cada um dos Sistemas Gerenciadores de Recursos (SLURM e OAR), e coletadas as estatísticas em termos de utilização de CPU, de memória e o tempo total de execução das simulações. Para isso foram utilizadas as ferramentas: *mpstat* para medição da taxa de utilização de CPUs, *vmstat* para taxa de utilização de memória, e o tempo de execução foi medido no log da simulação. Em ambos os SGRs foi utilizado o algoritmo de escalonamento FIFO (*First In First Out*) e todos os *job*s foram configurados com a mesma prioridade na fila de execução. As informações de CPU e memória foram coletadas desde o inicio das simulações, e depois a cada 3 segundos e armazenadas em arquivo para cálculos estatísticos posteriores.

4.2. Resultados

Nesta subseção são apresentados os resultados obtidos após a execução dos experimentos. Nos gráficos são apresentados os resultados para os quatro dias escolhidos para simulações utilizando os dois SGRs. As colunas na cor laranja representam os resultados obtidos com o SLURM e as colunas na cor verde representam os resultados obtidos com o OAR.

A Figura 1 apresenta a taxa média de ocupação das CPUs quando o modelo SUPIM estava executando. Analisando essa Figura 1 podemos observar que o SLURM

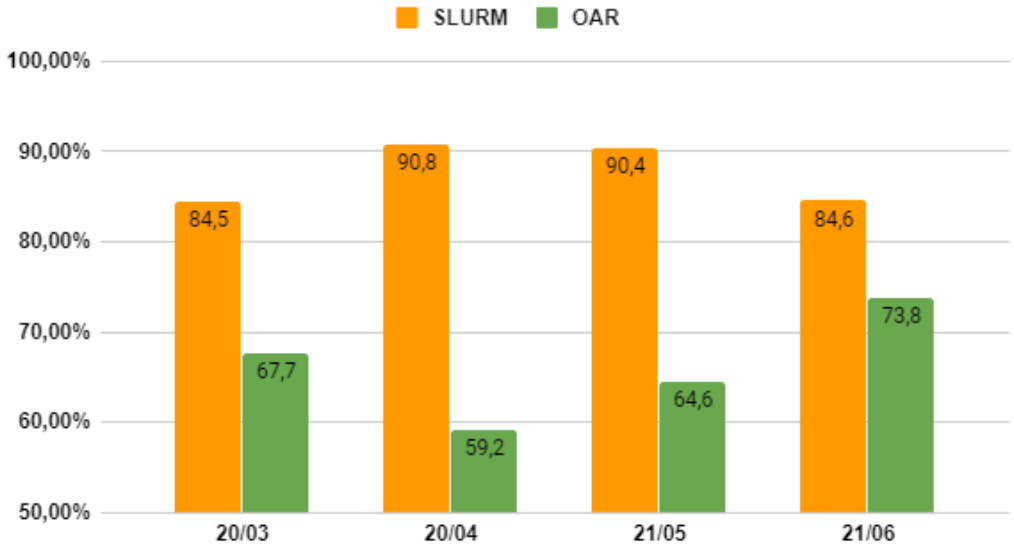

Figura 1. Taxa média de uso de CPU para execução do SUPIM

obteve um desempenho superior em todos os quatro dias. Isso deve-se ao fato de que o SLURM apresenta um sistema de submissão de *job*s mais eficientes que o OAR. No SLURM é possível submeter todos os *job*s de forma sequencial, mesmo que não tenha recursos suficientes para a execução, e o SGR fica responsável por realizar todo o escalonamento e submissão assim que um *job* concluir. Já na implementação que utiliza o OAR, um determinado *job* é submetido somente após o início de execução do anterior, e essas submissões ficam a cargo do script gerenciador do sistema de simulação.

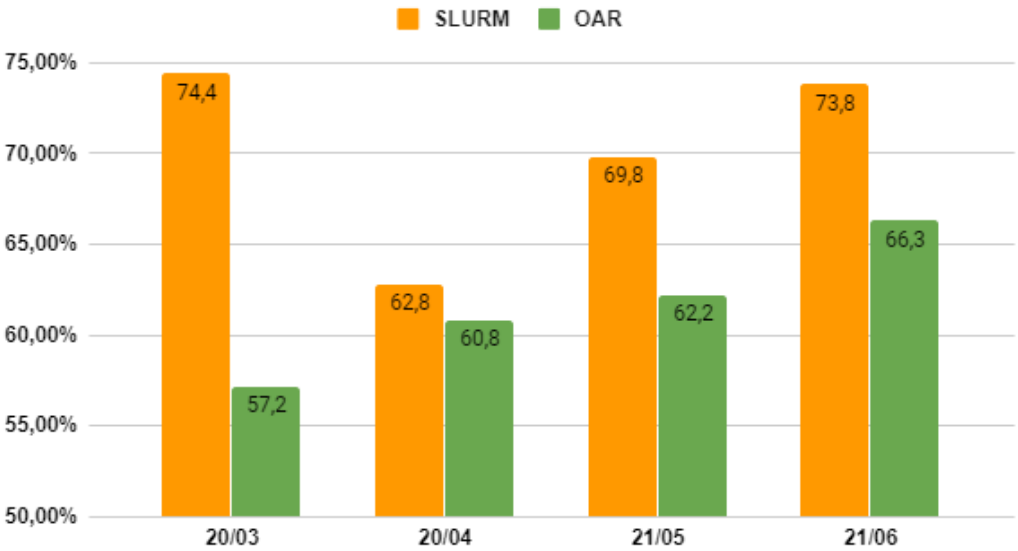

Figura 2. Taxa média de uso da CPU para execução do DAVS

Na Figura 2 podemos observar a média de ocupação de CPU durante a execução do DAVS. Neste cenário o problema é similar ao apresentado nas submissões dos *job*s do SUPIM, contudo como o número de *job*s é menor e não necessitam de tanto poder de

processamento como no SUPIM, as diferenças nas taxas de uso de CPU são menores. A maior diferença observada foi no dia 20/03. Isso deve-se ao fato de que os experimentos foram realizados utilizando uma aplicação científica real de previsão ionosférica, que depende de fatores externos para execução, o que pode acarretar variações importantes já que depende inclusive de dados externos para a execução.

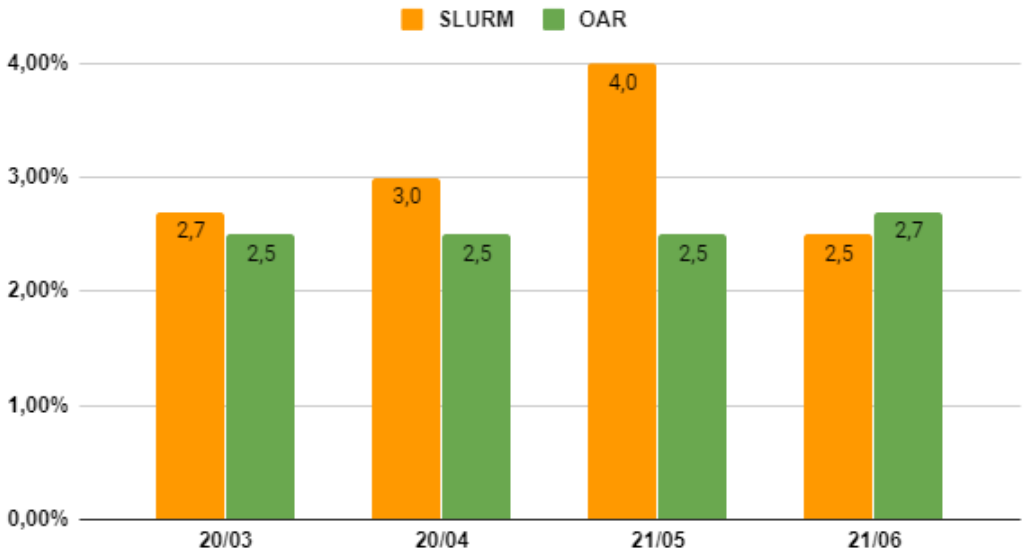

Figura 3. Taxa média de uso de memória para execução do SUPIM

Para avaliação da utilização de memória durante as simulações, a Figura 3 apresenta o valor médio utilizado durante a execução do SUPIM, e a Figura 4 durante a execução do DAVS. Ainda que as taxas de utilização de memória durante a execução sejam em média baixas, é possível observar maior equilíbrio na comparação entre os dois SGRs. Ainda assim, exceto pela simulação do dia 21/06, o SLURM apresentou taxas de uso de memória maiores.

A análise conjunta das Figuras 1 a 4 nos permite inferir que o SLURM apresentou, em quase todos os casos avaliados, uma maior taxa de utilização de recursos, tanto CPU quanto memória, quando comparado ao OAR. Espera-se, assim, que essa otimização na utilização de recursos resulte em menores tempos totais de execução da aplicação. Na Tabela 2 podemos avaliar o tempo total de execução médio requerido para as simulações executadas. Claramente, a execução do SUPIM é significativamente mais demorada que o DAVS, devido o processamento de uma gama de informações para realizar a previsão de toda ionosfera de forma tridimensional.

Na segunda coluna da Tabela 2, podemos observar o tempo utilizado para a execução da primeira parte da simulação ionosférica (SUPIM), onde podemos verificar que o SLURM usou significativamente menos tempo que o OAR para executar, em todos os dias simulados, em razão das taxas mais altas de uso de CPU apresentadas. Nesta etapa fica claro que o SLURM conseguiu aproveitar os recursos computacionais disponíveis de uma forma mais eficiente.

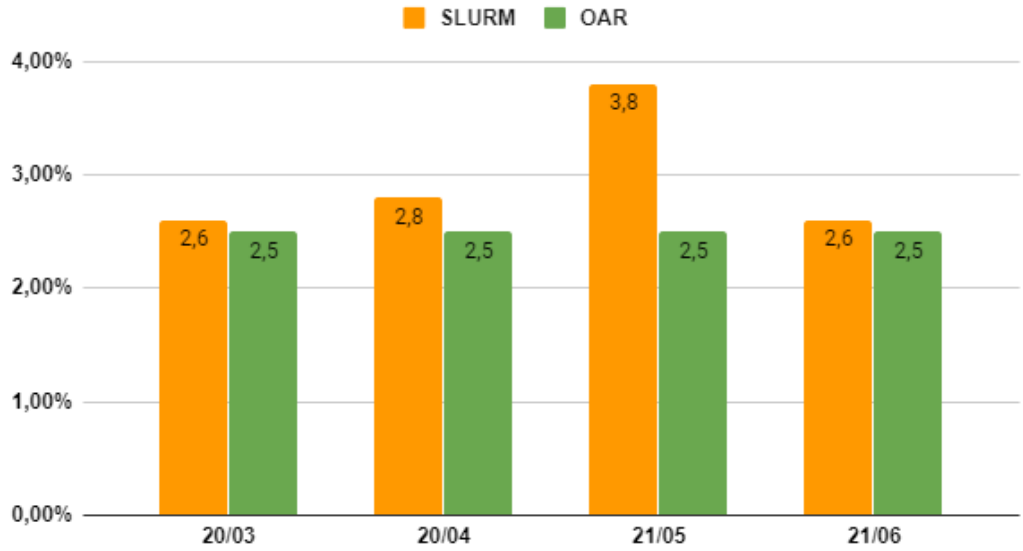

Figura 4. Taxa média de uso de memória para execução do DAVS

Na terceira coluna da Tabela 2 são apresentados os tempos de execução para o DAVS referente a cada dia de simulação. Nesta coluna é possível observar que o SLURM obteve desempenho similar ao OAR, e em dois dias de experimentos até obteve desempenho inferior ao OAR. Com isso, é possível observar que o OAR consegue trabalhar de uma forma mais otimizada quando a aplicação possui um número de *job*s reduzidos e com baixa requisição de CPUs.

Tabela 2. Média de tempo para simulações em horas

	SUPIM	**DAVS**	**SUPIM+DAVS**
20/03/2019			
OAR	02:01:08	00:18:06	02:20:29
SLURM	01:25:10	00:19:20	01:44:56
20/04/2019			
OAR	02:33:35	00:19:12	02:52:38
SLURM	01:24:09	00:19:36	01:44:20
21/05/2019			
OAR	03:04:03	00:21:25	03:25:33
SLURM	01:44:50	00:19:15	02:05:02
21/06/2019			
OAR	02:25:25	00:17:38	02:43:32
SLURM	01:52:09	00:19:06	02:10:59

Já na última coluna da Tabela 2, nota-se que apesar dos tempos de execução mais equilibrados entre os SGRs na execução do DAVS, a predominância dos tempos do SUPIM, significativamente maiores, resultaram em tempos totais favoráveis à utilização do SLURM, que proporcionou melhorias consideráveis nos tempos requeridos, com redução média de 53 minutos.

5. Considerações Finais

Este trabalho teve com objetivo avaliar e comparar dois SGRs, OAR e SLURM, em um ambiente de alto desempenho utilizando uma aplicação científica. Para a avaliação foram analisados os dados de taxa de utilização de memória, taxa de utilização de CPU e tempo de execução utilizando cada um dos SGRs para quatro dias de simulações ionosféricas.

Após a realização de todos os experimentos e análises, constatamos que o SLURM obteve melhor resultado dos dados oriundos da execução do SUPIM. Contudo, mesmo o OAR obtendo resultados similares para tempo de execução no DAVS, o SLURM conseguiu concluir a execução da previsão ionosférica em menor tempo.

Quando analisamos a taxa de utilização de memória, as performances foram equilibradas, e o OAR se mostrou superior em um dia de simulação. E para CPU, SLURM foi superior em todos os dias, com destaque para a execução do SUPIM.

É importante destacar que os experimentos foram realizados em um ambiente real onde uma aplicação científica executa diariamente e mesmo o OAR apresentando resultados inferiores, quando comparado ao SLURM, o tempo total de execução da simulação de ambos SGRs é aceitável para disponibilização diária dos resultados. Como pesquisas futuras pretende-se testar os dois SGRs em um ambiente formado por nós com configuração heterogênea e com algoritmos de escalonamento distintos aplicados aos SGRs, onde será possível analisar outras métricas de desempenho de sistemas gerenciadores de recursos em um ambiente real.

Referências

Gvozdetska, N., Globa, L., and Prokopets, V. (2019). Energy-efficient backfill-based scheduling approach for slurm resource manager. In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, pages 1–5.

Nicolas, C., Joseph, E., and ZIRST, M. (2003-2016). Oar documentation-user guide. *LIG laboratory, Laboratoire d'Informatique de Grenoble Bat. ENSIMAG-antenne de Montbonnot ZIRST*.

Petry, A., de Souza, J. R., de Campos Velho, H. F., Pereira, A. G., and Bailey, G. J. (2014). First results of operational ionospheric dynamics prediction for the brazilian space weather program. *Advances in Space Research*, 54(1):22–36.

Prabhu, C. (2008). *Grid and cluster computing*. PHI Learning Pvt. Ltd.

Puntel, F. E. (2019). Análise de desempenho de algoritmos de escalonamento aplicados a um sistema gerenciador de recursos para execução de aplicação operacional de computação científica. Master's thesis, Universidade Federal de Santa Maria.

Sloan, J. D. (2004). *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI: A Comprehensive Getting-Started Guide*. O'Reilly Media, Inc.

Yoo, A. B., Jette, M. A., and Grondona, M. (2003). Slurm: Simple linux utility for resource management. In Feitelson, D., Rudolph, L., and Schwiegelshohn, U., editors, *Job Scheduling Strategies for Parallel Processing*, pages 44–60, Berlin, Heidelberg. Springer Berlin Heidelberg.