dx.doi.org/10.7437/NT2236-7640/2026.01.002
Notas Técnicas, v. 16, n. 1, p. 17–24, 2026

Cluster HPC Planck: Infraestrutura, Provisionamento via Ansible e Avaliação de Escalabilidade Paralela

M. P. de Albuquerque,* P. J. Russano,† and P. M. Russano‡

Laboratório de Computação e Inteligência Artificial (LAB-IA) — Coordenação de Tecnologia (COTEC)

Centro Brasileiro de Pesquisas Físicas (CBPF/MCTI)

G. N. Santos§

Instituto Federal do Rio de Janeiro (IFRJ), Rio de Janeiro, Brazil

Submetido: 18/09/2026 *Aceito: 21/10/2026*

Resumo: Esta nota técnica descreve a implantação, a configuração e a validação do Cluster HPC Planck, um sistema de computação de alto desempenho (*High-Performance Computing*, HPC) implantado pelo Laboratório de Computação e Inteligência Artificial (LAB-IA), da Coordenação de Tecnologia (COTEC) do Centro Brasileiro de Pesquisas Físicas, e destinado a atender aos grupos de pesquisa em física teórica vinculados à Coordenação de Física Teórica (COTEO). O cluster é composto por quatro servidores Supermicro equipados com processadores AMD EPYC 9965, totalizando 1.536 núcleos físicos e cerca de 3 TB de memória. A infraestrutura foi integralmente provisionada por meio de código (Ansible), garantindo reprodutibilidade, e disponibiliza aos usuários um portal web (Open OnDemand) com ambientes interativos (Jupyter e VS Code), além de acesso por linha de comando. O gerenciamento de recursos é realizado pelo escalonador Slurm, com políticas de cota por perfil de usuário, e a autenticação é integrada ao serviço LDAP institucional. A capacidade do sistema foi avaliada por estudos de escalabilidade paralela em múltiplos nós — nas modalidades forte e fraca — e pela execução de uma simulação representativa de física estatística (modelo de Ising bidimensional), cujo resultado reproduziu o valor teórico da temperatura crítica. Todos os experimentos foram repetidos e os resultados apresentam a respectiva medida de dispersão (desvio padrão). Os resultados confirmam que o cluster é funcional, escalável, seguro e adequado ao uso científico. A estabilidade do sistema foi ainda comprovada por um teste de operação contínua de 24 horas, com mais de 27 mil execuções e disponibilidade de 99,92%.

Palavras chave: computação de alto desempenho (HPC); computação paralela; Slurm; MPI; provisionamento automatizado (Ansible); física computacional.

I. INTRODUÇÃO

A pesquisa em física teórica depende, de forma crescente, de simulações numéricas e do processamento de grandes volumes de cálculo. Problemas como simulações de mecânica estatística, integração numérica de sistemas de N corpos e métodos estocásticos exigem capacidade computacional que ultrapassa a de estações de trabalho convencionais. Nesse contexto, a computação de alto desempenho (HPC) — a agregação de múltiplos nós de computação operando de forma coordenada — torna-se um instrumento essencial de pesquisa [1].

O objetivo desta nota técnica é triplo: (i) documentar a arquitetura de hardware e software adotada no Cluster HPC Planck; (ii) descrever o modelo de provisionamento automatizado, as políticas de governança de recursos, as medidas de segurança e as interfaces de acesso disponibilizadas aos usuários; e (iii) apresentar a avaliação de desempenho realizada, tanto do ponto de vista de escalabilidade quanto da

* marcelo@cbpf.br
† pjrussano@cbpf.br
‡ pmiguel@cbpf.br
§ genildo.santos@ifrj.edu.br

execução de um problema científico real, com a metodologia empregada em cada caso.

O cluster foi implantado pelo Laboratório de Computação e Inteligência Artificial (LAB-IA), da COTEC, para atender aos grupos de pesquisa da Coordenação de Física Teórica (COTEO) do CBPF. Seu projeto seguiu uma hierarquia de prioridades: estabilidade, segurança, gerenciamento racional de recursos, reprodutibilidade e desempenho. Essa ordem norteou as decisões técnicas descritas nas seções seguintes, privilegiando um sistema robusto e reproduzível, no qual o desempenho bruto é explorado dentro de um arcabouço de governança e segurança.

II. ARQUITETURA E MÉTODOS

A. Infraestrutura de hardware

O Cluster Planck é constituído por quatro servidores Supermicro de dois soquetes, montados em rack. Um dos nós acumula as funções de nó mestre (controle e serviços), enquanto os demais atuam como nós de cálculo; o nó mestre também participa do processamento quando há capacidade disponível. Cada nó dispõe de dois processadores AMD EPYC 9965, com 192 núcleos por soquete, resultando em 384 núcleos físicos por nó e 1.536 núcleos no agregado do cluster (3.072 *threads* com o *multithreading* simultâneo). A Tabela I resume as especificações; as Figuras 1 e 2 mostram as vistas frontal e traseira do rack, e a Figura 3, a arquitetura lógica do sistema.

Componente	Especificação
Nós de computação	4 servidores Supermicro (2U): 1 nó mestre + 3 nós de cálculo
Processadores	2× AMD EPYC 9965 por nó (192 núcleos por soquete)
Núcleos	384 núcleos físicos por nó; 1.536 no total (3.072 threads com SMT)
Memória	∼754 GB por nó; ∼3 TB no agregado
Rede	Gigabit Ethernet (1 GbE)
Armazenamento	RAID10 NVMe (∼7 TB) no nó mestre, exportado via NFS 4.2
Sistema operacional	Ubuntu Server 24.04 LTS (kernel 6.8)
Escalonador	Slurm (accounting, fair-share e QoS por perfil)
Autenticação	LDAP institucional (SSH via `nslcd`; portal via Dex/OIDC)
Bibliotecas	OpenMPI 4.1.6; OpenBLAS; GCC/GFortran
Ambientes	Python 3.13, R 4.6.1 e Julia 1.10 (via Miniforge)
Provisionamento	Ansible (infraestrutura como código, reproduzível)

Tabela I. Especificações do Cluster HPC Planck. Fonte: elaborado pelos autores (2026).

B. Pilha de software e gerenciamento de recursos

O sistema operacional adotado é o Ubuntu Server 24.04 LTS. O gerenciamento de recursos e o escalonamento de tarefas são realizados pelo Slurm, que aloca núcleos e memória aos trabalhos (*jobs*) submetidos pelos usuários, mantendo filas e arbitrando o uso concorrente.

Figura 1. Vista frontal do rack: os quatro servidores Supermicro do Cluster Planck. Fonte: elaborado pelos autores (2026).

Figura 2. Vista traseira: cabeamento de rede, alimentação redundante e interfaces de gerenciamento. Fonte: elaborado pelos autores (2026).

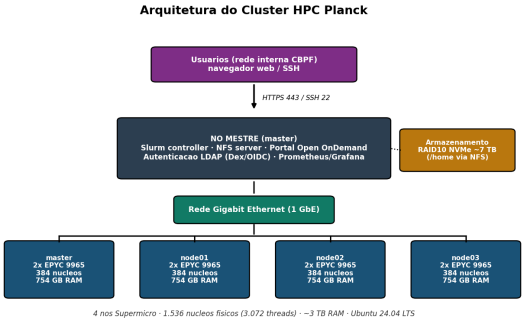

Figura 3. Arquitetura lógica do cluster: nó mestre, rede de interconexão, nós de cálculo e armazenamento compartilhado via NFS. Fonte: elaborado pelos autores (2026).

Sobre o Slurm foram configuradas políticas de contabilização (*accounting*), justiça de uso (*fair-share*) e qualidade de serviço (QoS). As cotas de recursos são definidas por perfil e associadas automaticamente aos usuários conforme seu grupo no diretório LDAP institucional: um processo de sincronização periódica atribui a cada usuário a QoS correspondente ao seu perfil (aluno, pesquisador ou sênior). A Tabela II apresenta as três classes de QoS efetivamente configuradas [2].

Os limites são expressos em núcleos lógicos: a QoS baixa (perfil aluno) permite até 384 núcleos lógicos, a normal (pesquisador) até 764, e a alta (sênior) até 2.000 —

QoS (perfil)	Prior.	Cota (núc. lóg.)	Jobs sim.	Tempo máx.
baixa (aluno)	10	384	10	2 dias
normal (pesquisador)	50	764	20	7 dias
alta (sênior)	100	2.000	40	7 dias

Tabela II. Classes de qualidade de serviço (QoS) configuradas. As cotas são expressas em núcleos lógicos (o cluster dispõe de 3.072 lógicos / 1.536 físicos). Fonte: elaborado pelos autores (2026).

cerca de dois terços da capacidade do cluster. Cada classe também define um número máximo de trabalhos simultâneos por usuário e um tempo máximo de execução. Requisições que excedam a cota do perfil permanecem em fila até que haja disponibilidade, sem comprometer o uso pelos demais. O escalonamento é organizado em duas partições: `debug`, para testes rápidos (limite de 30 minutos), e `compute`, para produção (limite de 7 dias).

O compartilhamento de dados entre os nós é feito por um sistema de arquivos em rede (NFS, versão 4.2): o diretório pessoal dos usuários (`/home`) reside em um volume RAID10 de discos NVMe no nó mestre e é montado de forma transparente em todos os nós de cálculo. A montagem foi ajustada para desempenho, empregando múltiplas conexões paralelas por ponto de montagem (`nconnect`) e blocos de leitura e escrita de 1 MiB, além da opção `_netdev`, que assegura a montagem apenas após a rede estar disponível.

Além do armazenamento compartilhado, cada nó dispõe de uma área de trabalho temporária local (`/scratch`), destinada a dados intermediários de alto volume. Para cada trabalho submetido, o Slurm cria automaticamente, por meio de um script de prólogo, um diretório privado nessa área, com permissões restritas ao usuário; ao término, a área é liberada. Esse mecanismo evita a sobrecarga do sistema de arquivos em rede com escritas intensivas e temporárias. O Quadro 1 apresenta o núcleo do prólogo.

Quadro 1. Script de prólogo do Slurm que cria a área de scratch local por trabalho.

```
#!/bin/bash
# cria area de scratch local e privada para o job
SCRATCH="/scratch/job.${SLURM_JOB_ID}"
mkdir -p "${SCRATCH}"
chown "${SLURM_JOB_USER}:" "${SCRATCH}"
chmod 700 "${SCRATCH}"   # acesso so do dono
```

Para computação paralela distribuída, o cluster dispõe do OpenMPI, que implementa o padrão *Message Passing Interface* (MPI) e permite que processos em nós distintos cooperem na resolução de um mesmo problema. As bibliotecas de álgebra linear (OpenBLAS) e os compiladores (GCC, GFortran) completam o ambiente de desenvolvimento.

C. Provisionamento automatizado (Ansible)

Toda a configuração do cluster foi implementada segundo o paradigma de infraestrutura como código, utilizando a ferramenta Ansible. Cada componente foi descrito em módulos reutilizáveis e versionados (*roles*), agrupados em famílias funcionais [3]:

- base do sistema e armazenamento: `base`, `nfs-server`, `nfs-client`, `scratch-local`;
- escalonamento: `slurm-controller`, `slurm-compute`, `slurmdbd`, `slurm-qos`, `slurm-ldap-sync`;
- autenticação: `ldap-client`, `munge`;
- ambientes científicos: `miniforge`, `openmpi-config`, `codeserver`, `apptainer`;
- portal e monitoramento: `openondemand`, `prometheus`, `grafana`;
- segurança: `firewall-master`, `firewall-nodes`, `ssh-hardening`, `fail2ban`, `apache-hardening`.

Essa abordagem garante reprodutibilidade: a partir dos mesmos arquivos, o cluster pode ser reconstruído de forma consistente, e qualquer alteração é aplicada de maneira controlada e idempotente. Reduz-se, assim, a dependência de procedimentos manuais e o risco de divergência de configuração entre os nós.

D. Interfaces de acesso ao usuário

O acesso ao cluster é oferecido por um portal web (Open OnDemand), que permite ao usuário trabalhar diretamente pelo navegador, sem necessidade de configuração local. Após a autenticação institucional, o pesquisador seleciona a aplicação desejada e a dimensão dos recursos, e o portal submete o trabalho ao escalonador de forma transparente. Duas aplicações interativas foram disponibilizadas: o Jupyter Notebook, com núcleos para Python, R e Julia, voltado à análise de dados e prototipagem; e o VS Code, voltado ao desenvolvimento, compilação e submissão de código, notadamente em C++ com MPI. Em ambos, o tamanho do recurso respeita a cota do perfil do usuário [4, 5]. A Figura 4 mostra a tela de autenticação do portal; as Figuras 5 e 6, a seleção de recursos para os ambientes Jupyter e VS Code; e a Figura 7, o fluxo de utilização, do acesso autenticado à execução e ao monitoramento do trabalho.

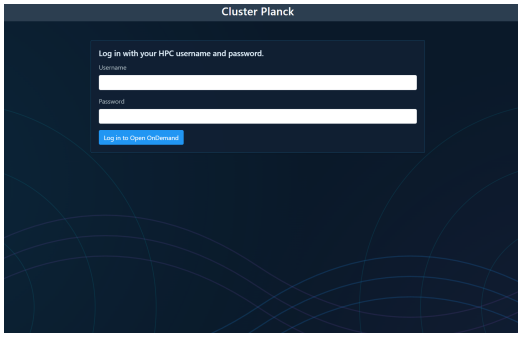

Figura 4. Tela de autenticação do portal, integrada ao LDAP institucional. Fonte: elaborado pelos autores (2026).

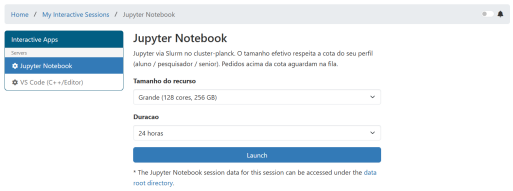

Figura 5. Portal Open OnDemand: seleção de recursos para o ambiente Jupyter (Python/R/Julia). Fonte: elaborado pelos autores (2026).

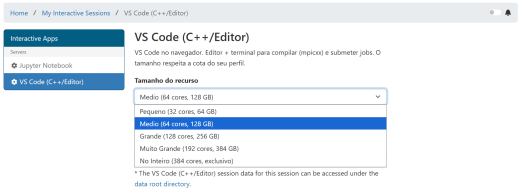

Figura 6. Portal Open OnDemand: ambiente VS Code, com opções de recurso até um nó completo. Fonte: elaborado pelos autores (2026).

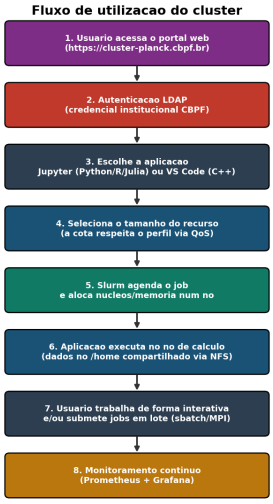

Figura 7. Fluxo de utilização: do acesso autenticado à execução do trabalho e ao monitoramento. Fonte: elaborado pelos autores (2026).

E. Monitoramento

O estado do cluster é monitorado de forma contínua por uma solução baseada em `Prometheus` (coleta de métricas) e `Grafana` (visualização). São acompanhados, por nó e de forma agregada, o uso de processadores e memória, o tráfego de rede, a temperatura dos processadores e o estado das filas e trabalhos do escalonador. A Figura 8 apresenta o painel de visão geral, com a ocupação do cluster, o estado dos nós, os recursos de CPU e as partições, capturado durante a execução de um trabalho.

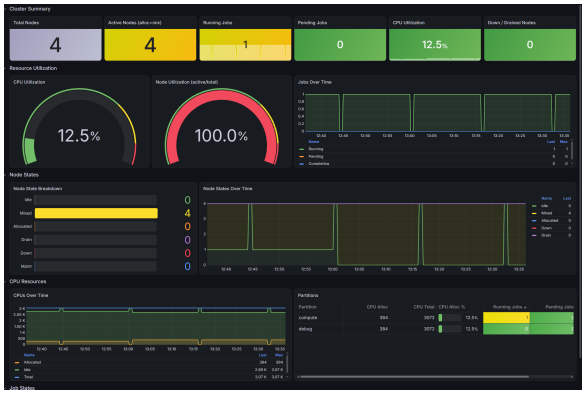

Figura 8. Painel de monitoramento (Grafana): visão geral do cluster, ocupação de núcleos, estado dos nós e partições, durante o processamento de um trabalho. Fonte: elaborado pelos autores (2026).

III. SEGURANÇA

Por atender múltiplos usuários e por expor um portal web na rede institucional, o cluster foi protegido por um conjunto de medidas de segurança em camadas, todas provisionadas via Ansible. O objetivo é restringir o acesso ao estritamente necessário, proteger os serviços internos e reduzir a superfície de ataque, sem comprometer a usabilidade [6].

A. Perímetro de rede e isolamento dos nós

Cada máquina do cluster possui um firewall (`ufw`) com política padrão de negar todo o tráfego de entrada, liberando apenas o necessário. No nó mestre, o acesso por SSH e ao portal web (portas 80, 443 e 3000) é permitido somente a partir da rede interna do CBPF; os serviços internos (NFS, Slurm) são liberados exclusivamente entre os quatro nós. Os nós de cálculo aceitam conexões apenas dos demais nós, de modo que nenhum usuário acessa um nó de cálculo diretamente — o acesso ocorre unicamente pelo escalonador. O Quadro 2 resume as regras do nó mestre.

Quadro 2. Resumo das regras de firewall do nó mestre.

```
# padrao: nega toda entrada; libera o essencial
22/tcp     ALLOW 152.84.0.0/16   # SSH (interna)
80,443/tcp ALLOW 152.84.0.0/16   # portal web
3000/tcp   ALLOW 152.84.0.0/16   # Grafana
(qualquer) ALLOW 152.84.60.31-34 # entre nos
```

B. Endurecimento do acesso SSH

O serviço SSH foi endurecido: o acesso direto do administrador (root) é proibido; apenas os grupos autorizados podem autenticar-se; e foram estabelecidos limites de tentativas e expiração de sessão ociosa. O Quadro 3 apresenta os principais parâmetros.

Quadro 3. Parâmetros de endurecimento do serviço SSH.

```
PermitRootLogin no          # root nao usa SSH
AllowGroups cbpf-users hpcadmin sudo
MaxAuthTries 4              # limite de tentativas
LoginGraceTime 30           # tempo p/ autenticar
ClientAliveInterval 300     # expira sessao ociosa
```

C. Proteção contra força bruta e do portal

A autenticação por senha foi mantida para viabilizar o uso das credenciais institucionais, dispensando a gestão individual de chaves. Para mitigar o risco, instalou-se a ferramenta `fail2ban`, que bloqueia temporariamente endereços com tentativas repetidas de acesso (cinco falhas em dez minutos resultam em bloqueio de quinze minutos), com os próprios nós isentos. No portal, adotaram-se HTTPS com política de transporte estrita (HSTS), ocultação da versão do servidor e autenticação institucional via OpenID Connect, com acesso restrito por grupo.

IV. RESULTADOS E VALIDAÇÃO

A avaliação foi conduzida em duas frentes: (i) estudos de escalabilidade paralela em múltiplos nós, nas modalidades forte e fraca; e (ii) uma simulação de física estatística que verifica a correção dos resultados. Todos os experimentos foram repetidos, e os valores apresentados são médias acompanhadas do desvio padrão. Salvo indicação, os testes de escalabilidade foram executados sobre três nós de cálculo, de modo que a comunicação entre processos atravessa a rede de interconexão.

A. Escalabilidade forte multinó

Na escalabilidade forte, mantém-se o tamanho total do problema fixo e mede-se a redução do tempo de execução ao se aumentar o número de processos. Empregou-se um cálculo de integração por método de Monte Carlo (estimativa de π), com carga total fixa distribuída entre um número crescente de processos (de 3 a 144), sempre repartidos igualmente entre três nós — de modo que a comunicação coletiva ocorre pela rede. Cada configuração foi repetida dez vezes. Definem-se o *speedup* $S = T_3/T_p$ (relativo à menor configuração) e a eficiência $S/(p/3)$. A Tabela III e a Figura 9 apresentam os resultados.

Observa-se que o *speedup* acompanha de perto o comportamento ideal, atingindo aceleração de 36,8 vezes com 144 processos. A eficiência paralela mantém-se elevada (77% a 81%), resultado consistente para uma interconexão Gigabit Ethernet, na qual a comunicação entre nós tem custo não desprezível. O desvio padrão, mostrado nas barras de erro, é inferior a 0,02 s em todos os pontos, o que corresponde a menos de 4% do tempo médio mesmo na configuração mais rápida (144 processos, 0,511 s), evidenciando a estabilidade das medições. Cabe registrar uma observação relevante: acima de aproximadamente 144 processos distribuídos, a comunicação coletiva do MPI sobre a rede Gigabit satura, degradando drasticamente o desempenho — o que delimita, na prática, a faixa de operação eficiente para cargas fortemente acopladas e reforça a rede como principal fator limitante do sistema.

Proc.	Tempo (s)	Speedup	Efic. (%)
3	18,785	1,0	100
6	11,596	1,6	81
12	5,806	3,2	81
24	2,906	6,5	81
48	1,457	12,9	81
72	0,986	19,0	79
96	0,760	24,7	77
120	0,606	31,0	77
144	0,511	36,8	77

Tabela III. Escalabilidade forte multinó: tempo, *speedup* e eficiência (média de 10 execuções). Fonte: elaborado pelos autores (2026).

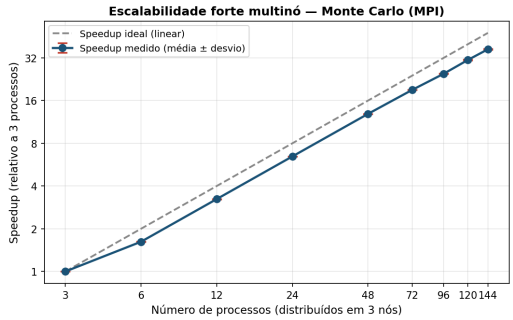

Figura 9. *Speedup* em escalabilidade forte multinó (3 a 144 processos em 3 nós), com barras de erro (desvio padrão de 10 execuções). Fonte: elaborado pelos autores (2026).

B. Escalabilidade fraca multinó

Na escalabilidade fraca, a carga por processo é mantida constante e o número de nós é aumentado, de modo que o trabalho total cresce proporcionalmente. O caso ideal corresponde a um tempo de execução constante. Empregou-se 16 processos por nó (carga fixa por processo), variando de 1 a 3 nós, com cinco repetições cada. A Tabela IV e a Figura 10 apresentam os resultados.

Nós	Proc.	Tempo (s)	Desvio	Efic. (%)
1	16	5,230	±0,006	100,0
2	32	5,232	±0,003	99,97
3	48	5,236	±0,002	99,89

Tabela IV. Escalabilidade fraca multinó: tempo e eficiência (média de 5 execuções). Fonte: elaborado pelos autores (2026).

O tempo de execução mantém-se praticamente constante (variação inferior a 0,1%) ao se passar de 1 para 3 nós, resultando em eficiência de escalabilidade fraca de quase 100%. Esse comportamento indica que, para cargas com comunicação moderada, o cluster escala de forma excelente entre nós, com sobrecarga de rede desprezível. Em conjunto, os dois estudos delimitam o comportamento do sistema: escalabilidade forte próxima da ideal na faixa suportada pela

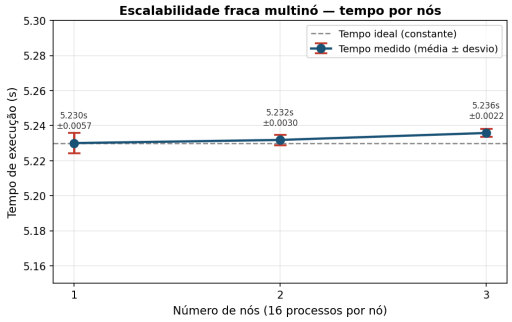

Figura 10. Escalabilidade fraca multinó: tempo de execução em função do número de nós, com barras de erro (desvio padrão de 5 execuções). Fonte: elaborado pelos autores (2026).

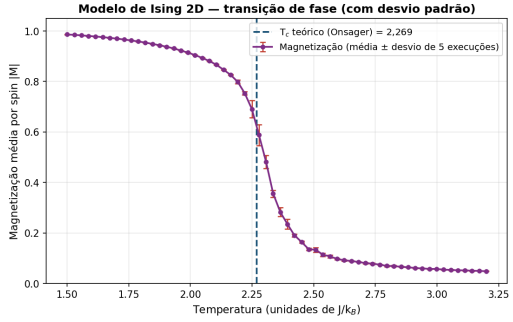

Figura 11. Modelo de Ising 2D: magnetização média por spin em função da temperatura, com barras de erro (desvio padrão de 5 execuções). A linha tracejada indica a temperatura crítica teórica de Onsager. Fonte: elaborado pelos autores (2026).

rede, e escalabilidade fraca praticamente perfeita para cargas de comunicação moderada. As definições formais de *speedup* e eficiência, bem como o limite teórico imposto pela fração sequencial de um programa, remontam à Lei de Amdahl [7].

C. Aplicação científica: modelo de Ising 2D

Para demonstrar a viabilidade do cluster em um problema científico real, e para verificar a correção dos resultados, executou-se a simulação do modelo de Ising bidimensional — um problema clássico da física estatística, cuja solução exata é conhecida, o que permite a validação por comparação com um valor teórico. O modelo descreve uma rede de spins que interagem com seus vizinhos: a baixas temperaturas o sistema se ordena (magnetização alta) e, acima de uma temperatura crítica, torna-se desordenado (magnetização nula). Para o modelo bidimensional, a temperatura crítica exata é $T_c \approx 2{,}269$ [8].

A simulação foi realizada pelo método de Monte Carlo, empregando o algoritmo de Metropolis, distribuído em 48 processos (16 por nó, em três nós) por MPI, cada um responsável por uma faixa de temperaturas — um padrão típico de varredura de parâmetros em física computacional. Cada temperatura foi simulada cinco vezes, permitindo obter a média e o desvio padrão da magnetização. A Figura 11 apresenta a curva de magnetização em função da temperatura, com as barras de erro correspondentes [9].

A curva reproduz nitidamente a transição de fase ferromagnética, com a queda acentuada da magnetização coincidindo com a temperatura crítica teórica ($T_c \approx 2{,}269$). Um aspecto físico notável é evidenciado pelas barras de erro: o desvio padrão é mínimo longe da transição, mas cresce de forma acentuada na região crítica (atingindo o máximo próximo a $T = 2{,}28$). Esse aumento das flutuações na vizinhança do ponto crítico é uma assinatura física esperada (as chamadas flutuações críticas), o que reforça a fidelidade da simulação. O conjunto completo da varredura foi concluído em poucos segundos, evidenciando a eficiência do processamento distribuído.

Os arquivos de resultados de todos os experimentos foram gerados automaticamente em formato CSV, contendo, para cada configuração, os valores medidos e a respectiva medida de dispersão. O Quadro 4 reproduz uma amostra do arquivo da simulação de Ising, evidenciando o crescimento do desvio padrão na região crítica.

Quadro 4. Amostra do arquivo de resultados (CSV) da simulação de Ising, com desvio padrão.

```
temperatura,magnetizacao_media,desvio_padrao
1.5000,0.9864,0.0001
...
2.2203,0.7536,0.0066
2.2492,0.6908,0.0345   # regiao critica
2.2780,0.5883,0.0418   # desvio maximo
...
3.2000,0.0484,0.0008
(60 registros no total)
```

D. Estabilidade e confiabilidade em operação contínua

Além da escalabilidade e da correção dos resultados, avaliou-se a estabilidade do sistema — isto é, sua capacidade de operar de forma contínua e confiável, sem perda de processos nem degradação de desempenho ao longo do tempo. Diferentemente da escalabilidade, que mede o ganho de velocidade, a estabilidade mede a previsibilidade e a robustez da execução, aspectos essenciais para um ambiente de produção multiusuário [10].

Para tanto, executou-se um teste de resistência (*soak test*) de 24 horas ininterruptas: a mesma tarefa MPI distribuída (48 processos em três nós) foi repetida continuamente, registrando-se, para cada execução, o tempo de conclusão, a correção do resultado e o eventual insucesso. Ao final, foram realizadas 27.246 execuções. A Tabela V resume os indicadores obtidos.

O histograma da Figura 12 evidencia uma distribuição extremamente concentrada em torno de 1,4547 s, com coeficiente de variação de apenas 0,178%. Essa dispersão mínima demonstra que o sistema entrega desempenho altamente previsível e determinístico ao longo de dezenas de milhares de execuções — um forte indicador de estabilidade.

A Figura 13 confirma a ausência de degradação ao longo do tempo: o tempo de execução permanece constante durante todo o período, sem tendência de aumento

Indicador	Valor
Execuções totais (24 h)	27.246
Execuções bem-sucedidas	27.223 (99,92%)
Tempo médio de execução	1,4547 s
Desvio padrão	0,0026 s
Coeficiente de variação (CV)	0,178%

Tabela V. Indicadores do teste de estabilidade de 24 horas. Fonte: elaborado pelos autores (2026).

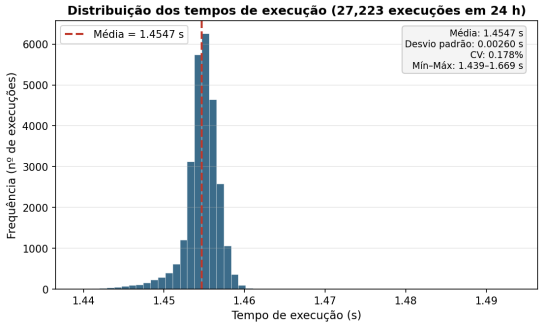

Figura 12. Distribuição dos tempos de execução das 27.223 execuções bem-sucedidas em 24 horas. Fonte: elaborado pelos autores (2026).

Figura 13. Tempo de execução ao longo das 24 horas. As linhas verticais indicam as 23 falhas de comunicação de rede. Fonte: elaborado pelos autores (2026).

ou deriva. Das 27.246 execuções, 23 (0,08%) resultaram em insucesso. A investigação dessas ocorrências revelou um padrão consistente: todas correspondem a falhas transitórias de comunicação de rede, registradas pelo escalonador com a mensagem "Communication connection failure to host", seguidas de recuperação automática — a execução imediatamente posterior a cada falha foi concluída normalmente. O Quadro 5 reproduz um trecho do registro do escalonador (`slurmctld`) que documenta essas falhas.

Essas falhas não decorrem de instabilidade do sistema de computação em si, mas da rede de interconexão Gigabit Ethernet, que ocasionalmente satura sob a carga de comunicação coletiva — a mesma limitação discutida na Seção V. A recuperação automática após cada evento demonstra a resiliência do sistema. Em síntese, o cluster apresentou disponibilidade de 99,92% em regime de operação contínua, com desempenho estável e previsível, sendo as raras interrupções atribuíveis exclusivamente à rede.

Quadro 5. Trecho do registro do escalonador (`slurmctld`) documentando as falhas de comunicação de rede.

```
[2026-09-09T14:22:31] error: slurmd error
  running JobId=967 :
  Communication connection failure
  to host node02
[2026-09-09T14:45:02] error: slurmd error
  running JobId=1330 :
  Communication connection failure
  to host node03
[2026-09-09T17:12:56] error: slurmd error
  running JobId=4467 :
  Communication connection failure
  to host node02
```

V. LIMITAÇÕES E TRABALHOS FUTUROS

Registram-se a seguir as principais limitações da configuração atual e as direções de evolução identificadas.

Rede de interconexão. A rede Gigabit Ethernet (1 GbE) é adequada ao modelo de uso predominante — trabalhos independentes, varreduras de parâmetros e cargas de comunicação moderada —, mas constitui o principal fator limitante para aplicações fortemente acopladas. Como quantificado na Seção IV A, acima de cerca de 144 processos distribuídos a comunicação coletiva satura. A adoção de uma rede dedicada de alto desempenho e baixa latência (por exemplo, 10/25 GbE ou InfiniBand) é a evolução de maior impacto previsível sobre a eficiência paralela em grande escala.

Nó mestre compartilhado. O nó mestre acumula os serviços de controle e também participa do processamento — decisão consciente que aproveita a capacidade de um servidor potente enquanto o cluster possui poucos nós. Com a expansão futura, prevê-se dedicá-lo exclusivamente aos serviços.

Caracterização de desempenho de pico. A avaliação priorizou escalabilidade e correção. A caracterização do desempenho absoluto de ponto flutuante, por meio de referências padronizadas como o HPL (*High-Performance Linpack*), constitui trabalho futuro.

Salvaguarda de dados (backup). O diretório dos usuários reside em volume RAID10, que oferece redundância contra falha de discos individuais, mas não substitui uma política de cópias de segurança. A implantação de um sistema de backup dedicado depende da aquisição de hardware de armazenamento específico, atualmente em fase de previsão, e figura entre as prioridades operacionais.

Otimizações adicionais. Incluem-se ainda o ajuste fino de afinidade de processos e topologia NUMA, a avaliação do impacto do *multithreading* simultâneo (SMT) e o contínuo refinamento das políticas de uso.

VI. CONCLUSÕES

Esta nota técnica documentou a implantação e a validação do Cluster HPC Planck, sistema de computação de alto desempenho implantado pelo LAB-IA (COTEC) e dedicado à pesquisa em física teórica dos grupos da Coordenação COTEO do CBPF. A infraestrutura reúne 1.536 núcleos físicos em quatro servidores AMD EPYC, integrados por um

escalonador Slurm com governança de recursos por perfil, autenticação vinculada ao diretório institucional, medidas de segurança em camadas e um portal web que oferece ambientes científicos (Jupyter e VS Code) diretamente pelo navegador. Todo o provisionamento foi realizado como código (Ansible), conferindo reprodutibilidade.

A avaliação de desempenho, conduzida em múltiplos nós e com repetições, demonstrou escalabilidade forte próxima da ideal (aceleração de 36,8 vezes, eficiência de 77% a 81%) e escalabilidade fraca praticamente perfeita (eficiência de quase 100%), delimitando com honestidade a faixa de operação eficiente do sistema em função da rede. A simulação do modelo de Ising 2D confirmou a correção científica dos resultados, ao reproduzir a temperatura crítica teórica e as flutuações características da região crítica. Em conjunto, os resultados demonstram que o cluster é estável, escalável, seguro e plenamente viável para o uso em pesquisa. As limitações identificadas orientam a evolução planejada, com destaque para a modernização da rede de interconexão e a implantação de uma política de salvaguarda de dados.

---

[1] STERLING, T.; ANDERSON, M.; BRODOWICZ, M. *High Performance Computing: Modern Systems and Practices*. Cambridge, MA: Morgan Kaufmann, 2018.

[2] ROTH, C. et al. Slur(m)-py: a slurm powered pythonic pipeline for parallel processing of 3d (epi)genomic profiles. *Epigenomics*, v. 17, n. 17, p. 1233–1247, 2025.

[3] BERTON, L. *Red Hat Ansible Automation Platform*. 1. ed. Delhi: BPB Publications, 2023.

[4] HUDAK, D. et al. Open ondemand: A web-based client portal for hpc centers. *Journal of Open Source Software*, v. 3, n. 25, p. 622, 2018.

[5] CHALKER, A. et al. Open ondemand: State of the platform, project, and the future. *Concurrency and Computation: Practice and Experience*, v. 33, n. 19, p. e6114, 2021.

[6] GUO, Y. et al. *High-performance computing security: architecture, threat analysis, and security posture*. Gaithersburg, MD, 2024. Disponível em: `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-223.pdf`.

[7] AMDAHL, G. M. Validity of the single processor approach to achieving large scale computing capabilities. In: *Proceedings of the April 18–20, 1967, Spring Joint Computer Conference (AFIPS '67)*. Atlantic City, New Jersey: ACM Press, 1967.

[8] ONSAGER, L. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, v. 65, n. 3–4, p. 117–149, 1944.

[9] METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, v. 21, n. 6, p. 1087–1092, 1953.

[10] DOMINGUEZ-TRUJILLO, J. et al. Lightweight measurement and analysis of hpc performance variability. In: *2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. [S.l.]: IEEE, 2020.