

Implementação de um Cluster Beowulf utilizando o framework Warewulf

**Ludivan Bento Brandão, João Paulo de Brito Gonçalves,
Paulo José Pereira de Oliveira**

Instituto Federal do Espírito Santo - Campus Cachoeiro de Itapemirim
Caixa Postal 727 - CEP: 29.300-970 - Cachoeiro de Itapemirim - ES - Brazil

ludivanldv@gmail.com, jpaulobg@ifes.edu.br, paulojoseo@gmail.com

***Abstract.** The evolution speed and use of computers has been growing very rapidly with the passing of the years, with this also comes the need of processing for quickly answers for evolutionary needs, such as financial or scientific calculations. With Clustering even old machines can be maintained in use, improving the computational performance of a academic or corporate environment through the sharing of resources within a local area network. In view of this, the objective of this work is to construct a Cluster following the Beowulf standard, achieving a performance gain using common computers.*

***Resumo.** A velocidade de evolução e uso de computadores vem crescendo muito rapidamente com o passar dos anos, com isso, também vem a necessidade de processamento para respostas rápidas às necessidades evolutivas, tais como cálculos financeiros ou científicos. Com a clusterização é possível manter máquinas antigas em uso, melhorando a performance computacional de um ambiente corporativo ou acadêmico por meio do compartilhamento de recursos dentro de uma rede local. Tendo em vista isso, o objetivo desse trabalho, é implementar um cluster Beowulf, conseguindo com isso um ganho de performance utilizando computadores comuns.*

1. Introdução

É indubitável que a tecnologia possui uma grande importância na forma de economia atual, sendo necessário por conta disso que as empresas se mantenham o mais informatizadas possível, o que implica em um caráter cada vez mais informacional no perfil de empregos, que por consequência demonstra também uma necessidade tecnológica que dê suporte a toda essa evolução e mudança na forma de trabalho, e que se dá desde o começo da era do conhecimento [Lastres et al. 1999].

Em 2011 foi verificado que no Brasil 27% do PIB vem de micro e pequenas empresas, sendo que o fator decisivo para diferenciar micro e pequenas empresas de uma grande se dá pela sua receita auferida ou número de pessoas ocupadas na empresa [SEBRAE 2014]. Visando seguir os avanços tecnológicos, uma pequena empresa poderia se beneficiar do uso de clusterização para ganho de processamento e economia na aquisição de novos computadores.

Cluster é uma palavra de origem inglesa que significa “aglomerado” em português é usada para se referir a um aglomerado de coisas semelhantes [Priberam 2017].

Em computação, cluster é uma forma de processamento distribuído que se tornou popular devido a relação preço/performance, pois é uma forma atrativa de se construir um

supercomputador utilizando tecnologias relativamente baratas em uma rede de alta velocidade, tendo como objetivo um conjunto de computadores realizando um processamento como se fosse um só [Tanenbaum and Van Steen 2007].

Algumas das principais razões para a utilização de clusters são: a evolução dos computadores pessoais (com processadores cada vez mais velozes), a evolução das tecnologias de redes locais, normalização das ferramentas para arquiteturas de clusters e o fato de serem alternativas mais baratas que os supercomputadores proprietários tradicionais [Roncolato et al. 2018].

Além do uso em computação para cálculos científicos, a área de clusterização tem vivenciado uma renovação por conta do seu uso em áreas recentes que demandam intenso poder computacional, como Big Data, Data Science e Mineração de Criptomoedas. Em comum em todas estas áreas, está a necessidade de realizar processamento sobre grandes volumes de dados no menor tempo possível, tornando o uso de clusters de alto desempenho uma opção recomendada. Inclusive, tem-se observado a criação de clusters de máquinas virtuais para acesso remoto em plataformas de computação em nuvem, aliando-se a clusterização à outro novo paradigma da tecnologia.

Este artigo está organizado da seguinte forma: na seção 2 será feita uma introdução ao Padrão Beowulf e ao framework Warewulf [Warewulf 2015]. Na seção 3 será descrita a implementação do cluster. A seção 4 tratará dos testes de desempenho realizados e resultados coletados e por fim, na seção 5 apresentaremos as considerações finais

2. Cluster Beowulf

Em 1983, a ARPA (atualmente conhecida como DARPA) lançou uma iniciativa para financiar a criação de computadores escaláveis e paralelos, diversas startups foram iniciadas e tentaram criar diferentes tipos de computadores paralelos, todos esses esforços falharam, mas em 1993, Tom Sterling e Don Becker lideraram um pequeno projeto com a NASA para construir uma potente estação de trabalho custando menos de US\$ 50.000. O projeto Beowulf era baseada na comodidade, computadores comuns e softwares disponíveis ao público, o projeto foi um sucesso, em 1994 foi montado um cluster de 16 nós, custando US\$ 40.000 e utilizando computadores intel 486, com isso, a clusterização Beowulf recebeu em 1997 o prêmio Gordon Bell pela relação performance/preço. [Sterling 2002]. Com isso, o Beowulf criou uma cultura de computação em cluster que permitia a qualquer pessoa capacitada a montagem de um cluster.

Um Cluster Beowulf tem como foco o ganho de processamento, podendo ser classificado como um cluster de alto desempenho (High Performance Cluster), são características deste tipo de cluster [Ridge et al. 1997]:

- a utilização de componentes não customizados (hardware comum)
- processadores dedicados
- uma rede privada (rede local por exemplo)
- uma base de softwares disponibilizados gratuitamente

2.1. Componentes

Um cluster padrão Beowulf pode ser visto como um sistema composto de 4 grandes componentes, dois de hardware e dois de software [Sterling 2002]. Os dois componentes

de hardware são: os computadores, que serão chamados de nós dentro do sistema, e a rede que conecta esses nós. Os dois componentes de software são: as ferramentas utilizadas para desenvolver aplicações paralelas como: MPI (Message Passing Interface), sockets e o ambiente de aplicações responsáveis por gerenciar os recursos para a computação (Middleware).

Todos os componentes são fundamentais para o funcionamento do cluster, já que o hardware provê a infraestrutura e o software provê a gerência do sistema, de forma que uma aplicação paralelizada utilizando MPI pode ser desmembrada em vários componentes que são alocados aos diversos processadores que compõem o cluster.

Em um cluster Beowulf existem dois tipos de nós, o nó mestre e os nós escravos, como pode ser visto na figura 1.

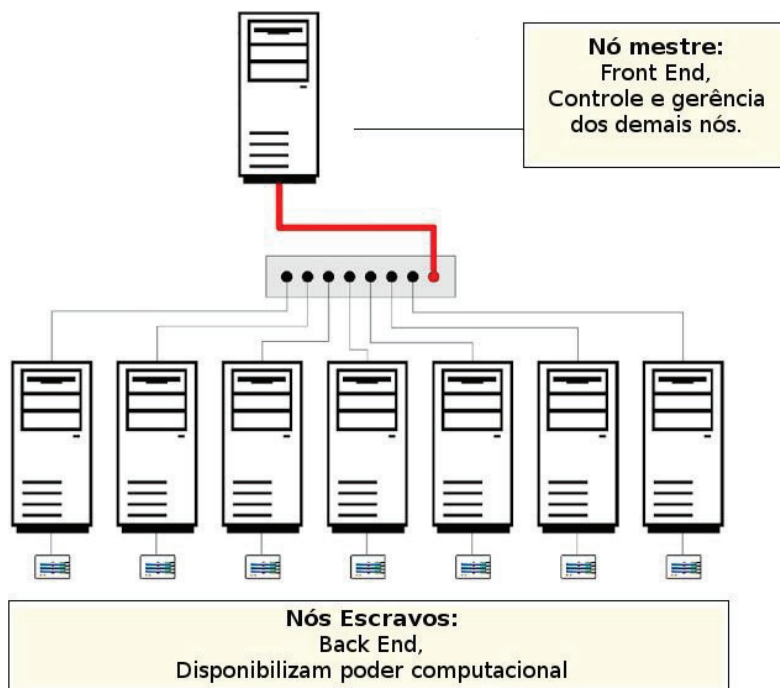


Figura 1. Adaptação estrutura de um cluster Beowulf. Fonte: [Gianni 2012]

2.2. Warewulf

Warewulf é um kit de ferramentas de gerenciamento desenvolvido originalmente para facilitar o desenvolvimento de grandes infraestruturas de clusters. Atualmente é uma solução que implementa um armazenamento de dados abstrato e orientado a objetos, com uma interface modular; sendo uma plataforma de código aberto.

Dentre os objetivos do projeto está: ser balanceado, prover à administradores, engenheiros, cientistas e provedores de serviços uma plataforma de gerenciamento que facilite seus requisitos, suporte a milhares de nós, estar oculto do sistema operacional ao invés de ser um componente integral, fácil de instalar e usar e estar aberto à contribuições do público [Warewulf 2015].

Com o Warewulf, a implementação de um cluster se torna mais simples, é possível automatizar a adição de nós, e a gerência do cluster se torna mais centralizada, uma vez que o sistema operacional provisionado aos nós escravos se encontra no mestre, e é provisionado aos demais nós de forma stateless.

2.2.1. Sistema Operacional e Aplicações

O sistema operacional selecionado para a implementação do Cluster foi o CentOS 6. A escolha do CentOS se deu pelo fato do framework Warewulf ser desenvolvido e testado em distribuições baseadas no Red Hat Enterprise Linux, além de que essas distribuições também são populares em servidores. A versão utilizada será a 6, o motivo da escolha foi por conta do warewulf ter sido desenvolvido e testado nessa versão.

As principais aplicações necessárias para a implementação foram: servidor MySQL [MySQL 2018], servidor web Apache [Apache 2018a], servidor DHCP [University 2018], servidor TFTP [TandemSystems 2018], módulos Perl [Apache 2018b], Network File System (NFS) [Jones 2018].

2.2.2. Hardware

Para a criação do cluster, foram utilizados 3 computadores disponibilizados pelo núcleo de pesquisa do Instituto Federal do Espírito Santo, Campus Cachoeiro de Itapemirim, suas especificações podem ser visualizadas na figura 2 e tabela 1:

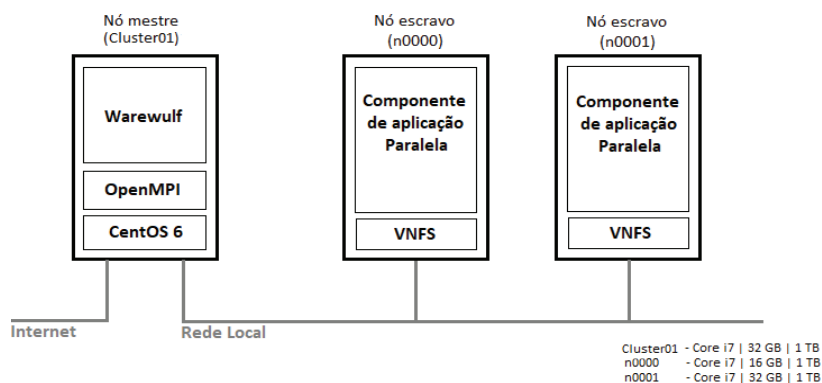


Figura 2. Estrutura do Cluster Implementado

| Máquina | Processador | Memória Principal | Memória Secundária |
|-----------|-----------------------------|-------------------|--------------------|
| Cluster01 | Intel Core i7-3370 3.40 GHz | 32 GB DDR3 | 1 TB |
| n0000 | Intel Core i7-4790 3.60 GHz | 16 GB DDR3 | 1 TB |
| n0001 | Intel Core i7-3370 3.40 GHz | 32 GB DDR3 | 1 TB |

Tabela 1. Especificações das máquinas utilizadas para implementação do cluster
Fonte: autor

3. Implementação

Após a instalação dos softwares listados na seção 3.1 o Warewulf foi configurado no computador mestre, sendo assim, foi criada uma base de dados no servidor MySQL, para abrigar informações sobre os objetos utilizados pelo cluster (máquinas, arquivos, etc.). Após isto foi configurado o repositório do Warewulf e instalados os pacotes Warewulf Common, Warewulf Provision, Warewulf Provision Server e Warewulf VNFS; após isso os arquivos database.conf, database-root.conf e provision.conf foram configurados. O Warewulf foi implementado de forma stateless, sendo assim, foi criado um diretório chroot e a imagem VNFS, assim como o arquivo de bootstrap, para que a imagem do

sistema operacional fique disponível no computador mestre e possa ser acessado nos nós escravos.

Após a preparação e configuração dos arquivos do Warewulf, o sistema pode então receber a adição de nós. O processo de inclusão de nós ao cluster envolve inicialmente a criação de uma rede interna entre os computadores integrantes e a configuração de rede do computador mestre de forma manual. A inserção de novos computadores pode ser feita de forma manual ou automática, na qual é feita uma varredura na rede em busca de máquinas disponíveis. Nesta implementação em particular foi feita a inserção manual, por conta de existirem poucos computadores a serem incluídos.

4. Testes e Resultados

Com o cluster implementado, foram realizados alguns testes para validar a implementação e ganho do desempenho do mesmo. Para isso foi utilizado uma suite de ferramentas para testes de clusters de alta performance, desenvolvida pela NASA, o NAS Parallel Benchmarks (NPB) [NASA 2017].

4.1. NAS Parallel Benchmarks (NPB)

NPB é composto de pequenos conjuntos de programas criados para ajudar a avaliar a performance de supercomputadores paralelos e ele é derivado de aplicações de Fluidodinâmica computacional (CFD) consistindo de 5 kernels e 3 pseudo-aplicações [NASA 2017].

O componente utilizado para o teste foi a pseudo-aplicação Lower-Upper Gauss-Seidel solver (LU). Esse componente é baseado no método de Gauss-Seidel, que é uma técnica para resolver as n equações de um sistema linear de equações uma de cada vez em sequência [Weisstein 2002].

Nesse método, cada componente de uma nova iteração depende de todos os componentes computados anteriormente, e cada nova iteração depende da ordem que as equações são examinadas, se essa ordem é diferente, a nova iteração também é diferente.

O NPB possui quatro classes de teste:

- a. Classe S - Classe com o objetivo de testes rápidos.
- b. Classe W - Pequenas estações de trabalho.
- c. Classe A, B e C - Testes de problemas de tamanho médio, cada classe possui um acréscimo de tamanho de aproximadamente 4 vezes em relação a anterior.
- d. Classe D, E e F - Testes de problemas de tamanho maior, possui um acréscimo de tamanho entre cada classe, de aproximadamente 16 vezes de cada classe anterior.

Para esse artigo, foram realizados 9 testes utilizando a pseudo aplicação LU de classe C, com dimensão de $162 \times 162 \times 162$ e 250 iterações; os testes utilizaram 1, 12 e 24 processadores, sendo que os 3 computadores possuíam 12 núcleos físicos, e 24 quando acrescidos núcleos físicos e virtuais.

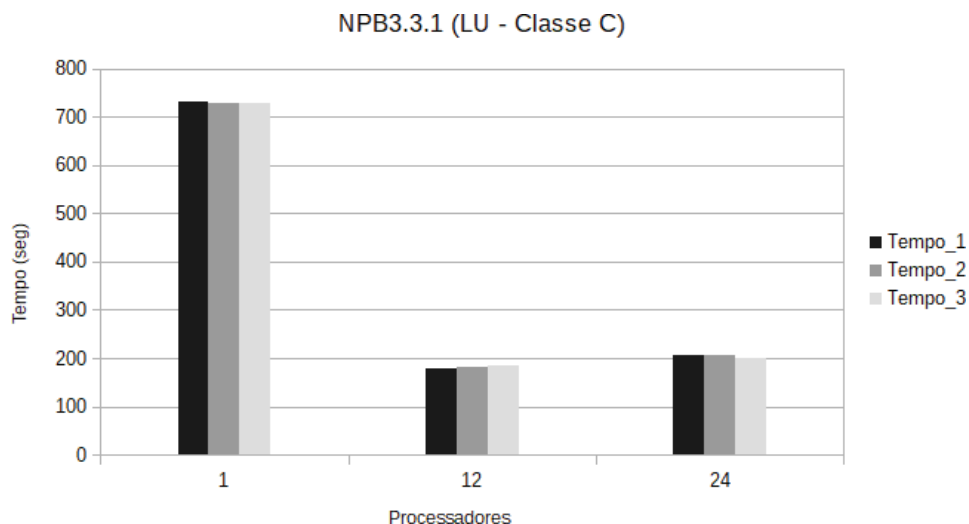


Figura 3. Resultados teste LU Classe C. Fonte: autor

| Número de Processadores | Tempo 1 | Tempo 2 | Tempo 3 |
|-------------------------|----------------|----------------|----------------|
| 1 núcleo | 0' 730'' 74''' | 0' 727'' 40''' | 0' 726'' 87''' |
| 12 núcleos | 0' 179'' 05''' | 0' 180'' 81''' | 0' 183'' 73''' |
| 24 núcleos | 0' 205'' 18''' | 0' 205'' 65''' | 0' 199'' 88''' |

Tabela 2. Resultados teste LU Classe C. Fonte: autor

4.2. Resultados

Cada teste foi realizado 3 vezes, e os resultados demonstraram o ganho de processamento com o acréscimo de processadores, como pode ser visto na tabela 2 e gráfico 1.

Esses resultados demonstram uma melhora do tempo de execução de aproximadamente 75,122% comparando a média do melhor resultado (12 núcleos paralelizados) e o pior resultado (1 único núcleo); um valor satisfatório, uma vez que em um teste LU de classe B, o supercomputador vetorial Cray C-90 e o supercomputador massivamente paralelo Cray T3D tiveram uma melhora de aproximadamente 92,043% comparando 16 núcleos com 1 núcleo (Cray C-90), e aproximadamente 65,691% comparando 256 e 32 núcleos (Cray T3D) [Bailey et al. 1992].

Entretanto, houve uma perda no cluster de aproximadamente 10,99% ao se comparar 12 núcleos com 24 núcleos; isso se dá, devido ao fato de que os computadores utilizados possuem 12 núcleos físicos, e mais 12 núcleos virtuais (Intel Hyper-Threading) [Intel 2018], por conta disso, nos testes executados o número de núcleos virtuais se mostrou irrelevante, tendo causado inclusive uma pequena degradação de desempenho, devido ao fato de que nem todas as aplicações se beneficiam da tecnologia de Hyper-Threading da intel [ARC 2018].

5. Considerações Finais

A implementação de um cluster seguindo o padrão beowulf, permitiu o reaproveitamento de máquinas comuns, com uma implementação utilizando apenas ferramentas de código aberto, tendo como custos apenas os dispositivos de hardware utilizados, como cabeamento, switch, etc.

A implementação foi mais simples graças a utilização do framework Warewulf, que simplifica e centraliza a implementação, gerência, e adição de nós ao cluster.

Para a validação do cluster foram realizados testes que tinham como objetivo verificar o ganho de desempenho. Os testes foram realizados por meio do software de benchmark Nas Parallel Benchmarks (NPB), utilizando a pseudo-aplicação LU (Lower-Upper Gauss-Seidel), uma aplicação baseada no método de Gauss-Seidel; sendo escolhido como tamanho do teste a classe C, sendo ele o maior problema de tamanho médio.

Os resultados atestam a melhora no desempenho do cluster com a adição de mais núcleos de processamento executando em paralelo, comprovando que é possível implementar supercomputadores de uma forma muito mais acessível, utilizando computadores comuns e ferramentas de código aberto.

Como trabalhos futuros, podemos citar a adição de mais nós ao cluster para aumentar seu poder computacional e validar sua utilidade em projetos de Big Data ou Data Science.

Referências

- Apache (2018a). Apache http server project, <https://httpd.apache.org/>. Acesso em 03 de março de 2018.
- Apache (2018b). Welcome to the mod perl world, <https://perl.apache.org/>. Acesso em 03 de março de 2018.
- ARC (2018). Running mpi applications, <http://www.arc.ox.ac.uk/content/how-run-mpi-applications>. acesso em 02 de março de 2018.
- Bailey, D. H., Dagum, L., Barszcz, E., and Simon, H. D. (1992). Nas parallel benchmark results. In Proceedings of the 1992 ACM/IEEE conference on Supercomputing, pages 386–393. IEEE Computer Society Press.
- Gianni (2012). “cluster beowulf”, <http://nosequenombreponerlegianni.blogspot.com.br/2012/03/cluster-beowulf.html>. Acesso em 16 de janeiro de 2018.
- Intel (2018). Tecnologia hyper-threading intel, <https://www.intel.com.br/content/www/br/pt/architecture-and-technology/hyper-threading/hyper-threading-technology.html>. Acesso em 04 de março de 2018.
- Jones, M. T. (2018). Network file system (nfs), <https://www.ibm.com/developerworks/library/l-network-file-systems/index.html>. Acesso em 03 de março de 2018.
- Lastres, H. M., Albagli, S., et al. (1999). Informação e globalização na era do conhecimento. Rio de Janeiro: Campus, page 163.
- MySQL (2018). Mysql, <https://www.mysql.com/>. Acesso em 03 de março de 2018.
- NASA (2017). Nas parallel benchmarks, <https://www.nas.nasa.gov/publications/npb.html>. acesso em 04 de outubro de 2017.
- Priberam (2017). “cluster”, <http://www.priberam.pt/dlpo/cluster>. Acesso em 16 de janeiro de 2018.
- Ridge, D., Becker, D., Merkey, P., and Sterling, T. (1997). Beowulf: harnessing the power of parallelism in a pile-of-pcs. In Aerospace Conference, 1997. Proceedings., IEEE, volume 2, pages 79–91. IEEE.

- Roncolato, F. J. et al. (2018). Otimização de suprimentos de tecnologia da informação por meio da implementação de cluster computacional na uftm, <http://bdt.d.uftm.edu.br/handle/tede/75>. Acesso em 13 de março de 2018.
- SEBRAE (2014). “Participação das micro e pequenas empresas na economia Brasileira”, <https://m.sebrae.com.br/Sebrae/Portal%20Sebrae/Estudos%20e%20Pesquisas/Participacao%20das%20micro%20e%20pequenas%20empresas.pdf>. Acesso em 16 de janeiro de 2018.
- Sterling, T. L. (2002). Beowulf cluster computing with Linux. MIT press. TandemSystems (2018). Tftp protocol, <http://www.tftp-server.com/tftp-server-help/tftp-protocol.html>. Acesso em 03 de março de 2018.
- Tanenbaum, A. S. and Van Steen, M. (2007). Distributed systems. Prentice-Hall. University, I. (2018).
- What is dhcp?, <https://kb.iu.edu/d/adov>. Acesso em 03 de março de 2018.
- Warewulf (2015). Página Oficial, <http://warewulf.lbl.gov/trac>. Acesso em 16 de janeiro de 2018.
- Weisstein, E. W. (2002). Gauss-seidel method