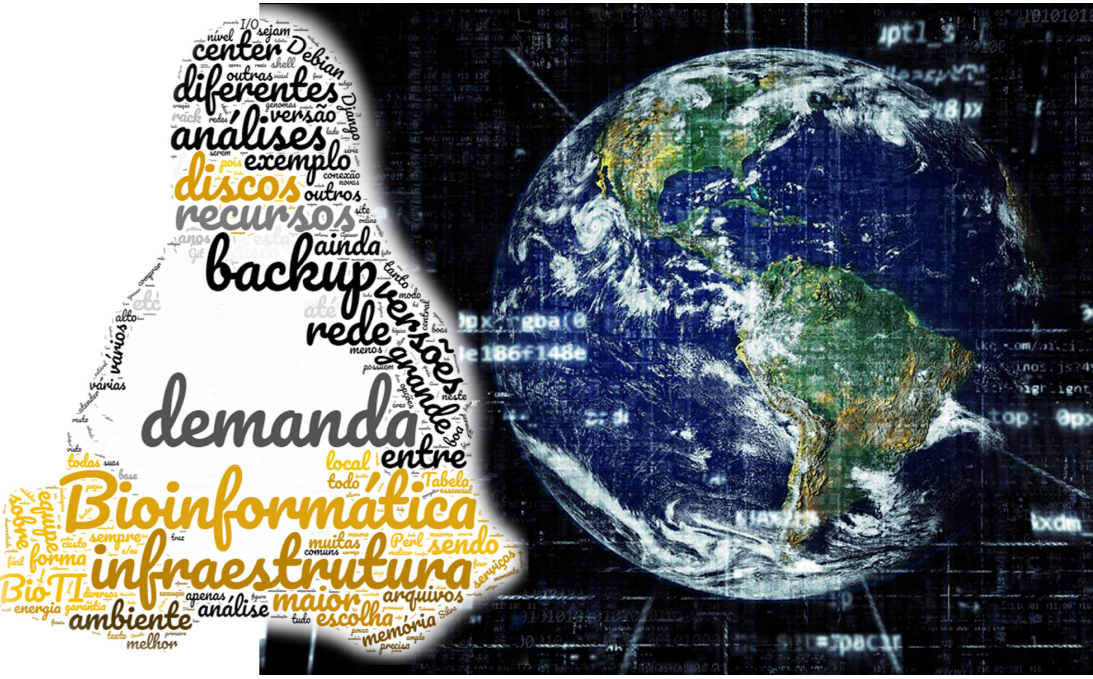
Documentos

ISSN 2177-4439
Dezembro, 2017 27

TI na Bioinformática: Implantação e Manutenção de Infraestrutura de BioTI – o Caso do LBB



Embrapa

ISSN 2177-4439
Dezembro, 2017

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Agroenergia
Ministério da Agricultura, Pecuária e Abastecimento

Documentos 27

TI na Bioinformática: Implantação e Manutenção de Infraestrutura de BioTI – o Caso do LBB

Eduardo Fernandes Formighieri
Marcelo Vicente de Paula
Marcelo Soares Souza
Andrei Stecca Steindorff

Embrapa Agroenergia
Brasília, DF
2017

Embrapa Agroenergia
Parque Estação Biológica (PqEB), s/nº.
Ed. Embrapa Agroenergia.
Caixa Postal 40315.
CEP 70770-901, Brasília, DF.
Fone: +55 (61) 3448-1581
Fax: +55(61)3448-1589
www.embrapa.br/fale-conosco/sac/

Supervisão editorial e revisão de texto: *Luciane Chedid Melo Borges*
Normalização bibliográfica: *Maria Iara Pereira Machado*
Editoração eletrônica: *Maria Goreti Braga dos Santos*
Imagem da capa: *Eduardo Fernandes Formighieri - https://www.wordclouds.com + imagem (CC0 Creative Commons) de https://pixabay.com.*

1ª edição
Publicação digitalizada (2017)

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Agroenergia

TI na Bioinformática: implantação e manutenção de infraestrutura de BioTI – o caso do LBB / Eduardo Fernandes Formighieri ... [et al.]. – Brasília, DF : Embrapa Agroenergia, 2017.

64 p. : il. – (Documentos ; v. 27).

Disponível em: http://www.embrapa.br/agroenergia/publicacoes

1. Bioinformática. I. Formighieri, Eduardo Fernandes. II. Série.

CDD 22. – 004

Autores

Eduardo Fernandes Formighieri
Engenheiro-agrônomo, doutor em Biologia Funcional e Molecular, pesquisador em Bioinformática da Embrapa Agroenergia, Brasília, DF.

Marcelo Vicente de Paula
Tecnólogo em Processamento de Dados, mestre em Gestão de TI, analista de TI da Embrapa Agroenergia, Brasília, DF.

Marcelo Soares Souza
Informata, consultor da Fundação Eliseu Alves, Brasília, DF.

Andrei Stecca Steindorff
Biólogo, doutor em Biologia Molecular, consultor da Fundação Eliseu Alves, Brasília, DF.

Apresentação

Neste documento, apresentamos o modelo de gestão do laboratório de bioinformática da Embrapa Agroenergia. Nele discutimos o modelo de planejamento de atividades, a gestão e a manutenção da infraestrutura. Esse laboratório é essencial para realizar análises dos dados levantados em estudos de genética e genômica de espécies vegetais e de microrganismos, trabalho essencial para o desenvolvimento de novas cultivares de plantas e estirpes de microrganismos fermentadores voltadas à produção de agroenergia. Esperamos que o documento sirva de orientação para infraestruturas laboratoriais semelhantes

Guy de Capdeville
Chefe-Geral da Embrapa Agroenergia

Sumário

TI na Bioinformática: Implantação e Manutenção de Infraestrutura de BioTI - o Caso do LBB

Eduardo Fernandes Formighieri
Marcelo Vicente de Paula
Marcelo Soares Souza
Andrei Stecca Steindorff

Introdução

BioTI ainda é um termo pouco utilizado no Brasil, embora "Bio-IT" já seja comum ao redor do planeta. Por exemplo, em maio de 2017 aconteceu a 15ª edição da *Bio-IT World Conference & Expo*[1], com foco em medicina de precisão. A BioTI é ainda mais ampla e pode ser entendida como a aplicação da TI na resolução de problemas da área biológica, incluindo saúde humana, desde a infraestrutura até visualização e análise de dados.

Uma das aplicações em que a BioTI pode apresentar características diferentes da maioria dos casos de TI é na infraestrutura de hardware e software para Bioinformática. Esta pode ser descrita como a aplicação da TI na resolução de problemas da biologia molecular, é um caso que apresenta desafios diferentes e alguns cuidados específicos.

[1] http://www.bio-itworldexpo.com

Desde os anos 1990, o campo da biologia molecular passou por grandes mudanças. O aumento da velocidade e a redução do custo de geração de dados biológicos (sequenciamento, genotipagem, etc.) chegaram ao ponto em que análises manuais ou com algoritmos utilizados até então eram incapazes de lidar com o montante de informação. A solução adotada incluiu desenvolvimento e implementação de algoritmos otimizados para grandes quantidades de dados, entre outras atividades. Com o avanço das tecnologias, novos softwares são demandados e desenvolvidos nessa área, e a demanda por infraestrutura continua crescendo.

Embora relativamente nova, a Bioinformática já passou de novidade para parte essencial no avanço da ciência em diversos campos, juntamente com o surgimento de diversas "Ômicas", como Genômica, Transcriptômica, Metabolômica, Metagenômica, Exômica, Fluxômica e tantas outras. Por exemplo, Teh et al. (2017) escreveram um *review* descrevendo diversas "Ômicas" utilizadas na pesquisa de culturas oleaginosas. Considerando, portanto, a amplitude de aplicações, e sem a pretensão de tratar a BioTI de forma completa, utilizaremos como exemplo a demanda recente do Laboratório de Bioinformática em Bioenergia (LBB/Embrapa Agroenergia), um caso de Bioinformática genômica, demanda comum em unidades de pesquisa agropecuária.

No LBB, as demandas envolvem microrganismos e plantas, incluindo análises como: montagem e anotação estrutural de genomas; comparação de genomas e transcriptomas; repetições e elementos transponíveis; filogenia molecular; variantes e marcadores moleculares; descoberta e caracterização de genes, promotores e reguladores; e anotação funcional. Mais detalhes podem ser vistos na página do LBB[2].

Os diferentes tipos de análises geram diferentes demandas computacionais, como processamento, memória, espaço em disco ou I/O. Quando a análise envolve busca de similaridade local entre sequências, normalmente com NCBI BLAST+ (ALTSCHUL et al., 1990), existe forte demanda por processamento. Quando se trata de

[2] https://lbb.cnpae.embrapa.br

montagem de genomas complexos, a utilização de memória é ainda mais crítica. Montagens que, com memória RAM suficiente, demoram semanas normalmente travam quando utilizam SWAP (discos SSD ainda não foram avaliados nesse contexto).

Além da demanda por memória RAM e processamento, que podem ser relativamente incomuns, muitos dos softwares utilizados na Bioinformática são desenvolvidos em instituições de pesquisa, por estudantes, que normalmente não são programadores profissionais e têm pouco tempo de dedicação exclusiva para a atividade. Dessa forma, além de o desenvolvimento normalmente não incluir boas práticas de programação, após a saída do estudante da instituição, a evolução, a manutenção e a correção de erros do software fica limitada (quando não é completamente interrompida), sendo comuns falhas, baixa performance e diversas incompatibilidades na aplicação do software em ambientes distintos do original.

No contexto de alta demanda de TI no País (e no mundo), a Bioinformática é uma área muito pequena e especializada, sendo raros os profissionais de TI com capacitação específica ou experiência para atender a essa demanda de suporte. Normalmente, são pessoas de formação próxima atendendo a essa demanda e que aprendem as particularidades com a própria experiência. Muitas vezes, são os próprios bioinformatas, muitos destes sem formação acadêmica em TI, que precisam estruturar e manter a infraestrutura de TI para Bioinformática.

O contexto de criação do LBB não fugiu à regra, já que foi iniciado por um engenheiro-agrônomo em uma unidade ainda nova da Embrapa, que não possuía nem prédio, nem profissional de TI e nem data center. Com o passar do tempo, o prédio da unidade foi construído e adaptado e a equipe de TI foi sendo montada. Em paralelo, o desenvolvimento de uma infraestrutura adequada para atender à Bioinformática demorou também alguns anos e, para atingir a maturidade atual, pelas razões já citadas, dependeu de tempo e expertise de profissional externo à Embrapa.

Considerando o contínuo desenvolvimento de tecnologias e a relativa redução de custos na geração de dados biológicos, é esperado que cada vez mais unidades de pesquisa, públicas ou privadas, criem ou ampliem laboratórios de Bioinformática.

Nesse contexto, o objetivo deste documento é trazer um guia básico para auxiliar profissionais de TI e usuários sem TI a começar a dimensionar e implementar uma infraestrutura de BioTI para Bioinformática, e ainda trazer algumas dicas para uma manutenção eficiente e segura. Para ampliar o alcance do trabalho, utilizamos referências em língua portuguesa sempre que possível.

Planejamento

Organizamos este tópico na forma de perguntas comentadas, tanto para dar uma dimensão inicial do tamanho, da complexidade e do nível de especialização do tema, quanto para auxiliar no dimensionamento de infraestrutura e na escolha de ferramentas de BioTI. Não existem fórmulas mágicas, mas aqui estão várias das principais perguntas que devem ser respondidas da melhor maneira possível antes da primeira compra ou da primeira contratação.

Quais as maiores demandas por processamento e memória?

É importante entender a demanda atual e prever a demanda futura da melhor maneira possível. Se o foco são organismos de genomas menores e mais simples, como vírus, bactérias, leveduras e microalgas, independentemente do tipo de análise, a demanda de processamento tende a ser relativamente baixa. Por outro lado, genomas de plantas são maiores e mais complexos, em vários casos demandando equipamentos de alta performance para algumas análises. Por exemplo, um genoma de planta com cerca de 2 GB e conteúdo repetitivo acima de 50% pode demorar mais de um mês para cada montagem *de novo* realizada (num servidor de alto desempenho, dados NGS tipo *Illumina HiSeq*) e pode demandar de 500 GB a mais de 1.000 GB de memória

RAM, dependendo de tipo e quantidade de dados biológicos (FERREIRA FILHO et al., 2017). Sem memória RAM suficiente, a montagem nesse caso não pode ser realizada.

Quando a quantidade de dados biológicos a ser analisada é grande, tarefas como alinhamentos de sequências (como com NCBI BLAST+) e análises *ab initio,* de modo geral, costumam gerar alta demanda por processamento, podendo durar semanas mesmo em bons servidores com dezenas de *threads*. De forma diferente da montagem, esse tipo de demanda pode ser executada com menor poder de processamento, e o que deve ser avaliado é a quantidade de dados a serem analisados e os prazos de entrega dos resultados. Muitos dados e prazos curtos dependem necessariamente de maior poder de processamento.

A quantidade de dados exige *storage*?

Ao realizar análises, são gerados arquivos intermediários e finais que multiplicam muito o espaço utilizado pelos arquivos originais. Não é raro que a demanda chegue a dezenas de vezes maior (em alguns casos, até centenas, variando com tipo de análise, software ou script utilizado, etc.). Um aspecto relevante é que a quantidade de dados brutos gerados com o passar do tempo costuma ser muito maior do que o previsto inicialmente, portanto é importante tratar da capacidade de armazenamento com boa margem de segurança e muita antecipação.

Práticas como a utilização de formatos menores para os resultados (como NCBI BLAST+ tabular), a compactação de arquivos, a não geração de arquivos de logs que sejam desnecessários, e a limpeza de arquivos intermediários no decorrer e depois das análises podem reduzir bastante a demanda por espaço (e são algumas boas práticas recomendadas). Políticas de segurança também afetam essa demanda, como tipo de RAID e política de backup. De qualquer maneira, quanto mais dados brutos, maior a capacidade de armazenamento necessária, sendo essencial fazer estimativas (otimista, pessimista e esperada) a respeito da demanda por espaço em disco nesse dimensionamento.

Que tipo de disco utilizar?

O objetivo deste tópico é determinar a maior ou menor importância da velocidade de leitura e escrita em discos. Análises que dependam de escrita prioritária de grande quantidade de informação com prazos curtos, pouca capacidade de memória RAM (em relação à demanda), pouco espaço em discos (demandando compactação/descompactação) e acesso de muitas pessoas ao mesmo tempo, são exemplos de situações que demandam maior capacidade de I/O, levando a uma demanda por discos mais rápidos (SSD, que também são mais caros). Não havendo tanta pressa ou tantos recursos, discos SATA são suficientes para a grande maioria das situações atuais (e são bem mais baratos). Uma solução intermediária em velocidade e preço são os discos SAS, os sucessores dos discos padrão SCSI, com vantagens de gerenciamento e aplicabilidade.

Havendo possibilidade de recursos, uma alternativa interessante é ter uma partição com discos SSD (para análises que demandam alta performance) e outras com SATA (para maior capacidade de armazenamento a baixo custo).

Como organizar o backup?

Backup bem estruturado é item obrigatório, então não se trata de precisar ou não, mas da definição da infraestrutura e da política mais adequadas. Dados brutos demandam backup (permanente) quando apenas são obtidos, mas para resultados intermediários e finais de análises, assim como para dados de usuários, é necessária uma política de backup frequente específica e acompanhamento constante.

Quais as frequências necessárias de backup total e incremental para cada conjunto de dados? Existe necessidade de restrição de quantidade de dados para grupos ou usuários? Qual a forma de armazenamento mais adequada (fita ou disco – recuperação mais rápida)? E qual é o tempo de retenção necessário a cada tipo de dado? Como será feita a duplicação do backup em outro local (prevenção de acidentes)? Em que horário a rede atual aguenta a demanda extra, se aguenta? Esses e outros fatores são determinantes na definição dessa política.

Existem diferentes modelos para essas políticas disponíveis na internet, como o do software Bacula[3], mas esse é um assunto crucial no qual recomendamos a contratação de um profissional qualificado, se já não for o seu caso.

É possível iniciar o assunto? Podemos trazer algumas considerações aplicáveis à maioria dos casos, como: a) atualmente é mais recomendável a utilização de discos (*storage*, com redundância) em vez de fitas; b) discos padrão SATA são suficientes para a maioria dos casos; c) softwares gratuitos, como Bacula, já possuem interfaces amigáveis para quem preferir e são mais do que suficientes para a função; d) dados de usuários são sensíveis, e é recomendável um tempo de retenção mais longo; e) análises que geram muitos dados intermediários e que não sejam excessivamente longas não precisam entrar no backup até que sejam finalizadas; f) política de organização de diretórios, de conteúdo, de utilização de espaço e de utilização de servidores está intimamente relacionada à política de backup.

Existe demanda por desenvolvimento de sistemas Web?

Se for necessário um portal apenas com informação estática (por exemplo, do laboratório e/ou de projetos), este pode até ser colocado em servidor externo. Mas quando existe demanda por disponibilização de informações dinâmicas e/ou controladas, seja através de softwares externos ou através de sistemas baseados em Web desenvolvidos localmente, o ideal é desenvolver localmente, o que envolve infraestrutura de hardware e software próprios, ferramentas computacionais específicas e outro tipo de conhecimento de programação (diferente de scripts e afins).

Exemplos de demandas que normalmente justificam a implantação de infraestrutura local para Web: a) disponibilização de conteúdo dinâmico de projetos, como buscas booleanas em conteúdo armazenado em SGBDs; b) visualização de estruturas anotadas em genomas (*genome browsers*, como o Jbrowse (SKINNER at al., 2009); c) BLAST (e

[3] http://www.bacula.com.br/modelo-de-politica-de-backup.

outras análises) contra bases de dados privadas; e d) sistemas Web desenvolvidos sob demanda (como organização e buscas em resultados de análises, organização de serviços, enfim, todo tipo de demanda específica e/ou que não deva ser compartilhada em ferramentas de fora da organização).

Essa infraestrutura envolve diferentes aspectos, e existem algumas diferentes formas de fazê-la, mas alguns dos principais pontos normalmente considerados são: boa conexão com a internet; data center adequado, com segurança elétrica, ar condicionado, monitoramento; sistema de backup adequado; responsável pela manutenção do sistema capacitado; DMZ externa com proxy reverso; DMZ Interna com demais servidores e afins; virtualização de servidores específicos com sistemas operacionais estáveis e "leves" (Web, arquivos, PostgreSQL, MySQL, e backup); ferramentas para desenvolvimento Web (como o *framework* Django, baseado em Python); boa velocidade de conexão entre os servidores, os *storages* e o backup (*switch*, cabeamento 1 GB ou 10 GB, fibras ópticas, etc.).

Finalmente, você tem autorização da empresa para ter domínio próprio? Ou será serviço dentro da intranet? A rede, interna ou externa, suportará a demanda dos seus serviços?

Que distribuição GNU/Linux utilizar?

A distribuição GNU/Linux a ser utilizada depende em parte da preferência de quem gerencia o sistema, mas existem alguns fatores que devem ser levados em consideração. O primeiro é a escolha de uma versão LTS (*Long Term Stable*), que é garantia de atualização por mais tempo, permitindo que a substituição de versão seja realizada quando possível e sem risco de segurança. Hoje, existem várias distribuições LTS gratuitas, como Debian, CentOS e Ubuntu, que são estáveis e atendem à grande maioria das demandas. Se é necessário suporte 24x7, e se existir demanda e orçamento para a aquisição, pode-se optar por uma distribuição paga, como a *Red Hat Enterprise Linux*.

Os serviços precisam ficar o tempo todo online?

Se seus sistemas Web não podem ficar fora do ar, pelas características da sua demanda, isso trará impactos em diversos aspectos da infraestrutura. A solução passa por dois pontos principais: a) contrate um profissional de TI qualificado para instalação, configuração e manutenção da infraestrutura; e b) considere seriamente as recomendações técnicas desse profissional. Contar com um profissional de TI é uma das recomendações mais importantes desse trabalho, não apenas para dimensionamento e implementação, mas principalmente para os constantes acompanhamentos e manutenção da infraestrutura.

Alguns dos principais aspectos, detalhados posteriormente, serão listados a seguir. É recomendável que os equipamentos tenham garantia enquanto em produção, com prazo curto para resolução de problemas, assim como a utilização de sistema operacional que inclua suporte pago. É necessário plano de manutenção, ampliação e substituição de equipamentos. O data center precisa ter refrigeração e energia corretamente dimensionadas e com a devida redundância, bem como sistemas de monitoramento e alarme, proteção contra incêndios e outros incidentes.

SGBDs de uso livre atendem à demanda?

Além da utilização no desenvolvimento de sistemas próprios, os SGBDs são necessários a várias ferramentas computacionais externas utilizadas para a Bioinformática. A maioria delas utiliza PostgreSQL e MySQL, opções gratuitas e que atendem plenamente às demandas de Bioinformática. Ambas possuem comunidades e documentações extensas. A escolha de versão paga só se justificaria por preferências pessoais (como know-how anterior e necessidade por suporte constante).

Serão necessários scripts e pequenos programas?

Existem vários programas prontos para as mais diversas análises de Bioinformática, afinal, boa parte dos problemas a serem resolvidos são comuns a todos os usuários. Entretanto, problemas específicos surgem a todo momento, como: processamento de grande quantidade de

dados; adaptação de programas aos seus dados; ou desenvolvimento de fases completas de análise.

A capacidade de resolver esses problemas de forma rápida, verificável, reproduzível e precisa, seja adaptando scripts ou desenvolvendo-os desde o início com diferentes linguagens de programação, é muito importante para o bioinformata. E, se o desenvolvimento for necessário, deverá existir infraestrutura de BioTI adequada.

A escolha das linguagens de programação depende de preferência pessoal, mas ainda assim existem as mais adaptadas para determinados contextos, e as mais utilizadas, como shell script e awk/sed para ações simples, Python e Perl (puros ou através de módulos de Biopython e Bioperl) para scripts mais complexos, R para estatísticas e gráficos, e até C ou C++ para análises mais complexas e pesadas, e que demorariam muito nas linguagens interpretadas citadas.

Considerando que precisará organizar e desenvolver scripts, é recomendável centralizar o que for sendo desenvolvido por meio da utilização de controlador de versão, como o Git (ex.: GitLab para arquivos locais, GitHub para versões públicas).

Existe demanda para virtualização?

Os representantes vendem a virtualização (deles) como uma solução moderna, urgente e imprescindível, mas será que você precisa disso para Bioinformática? Cada vez menos, e cada vez mais. Menos, porque uma das demandas para virtualização são os softwares abandonados, que exigem bibliotecas específicas e que não funcionam em versões mais recentes dos sistemas operacionais, mas dada a maior abundância de softwares e à maturidade dos grandes centros da área, essa dependência é cada vez menor. Por outro lado, precisamos mais, porque a grande quantidade de dados e a importância crescente da área exige uma infraestrutura profissional, mais estável, mais rapidamente recuperável, modular e eficiente, e a virtualização é aplicável em vários desses aspectos.

A Máquina Virtual (VM) apresenta diversas vantagens, como: ser facilmente desligada (como em caso de invasão); ser duplicada; permitir melhor utilização de máquinas reais; ter distribuição GNU/Linux mais adequada a cada tipo de utilização; a compartimentalização especializada facilita a utilização e a manutenção (apesar do maior número de SOs para manter); ser redimensionada sempre que necessário, etc. A partir de certo porte, a Bioinformática não pode mais prescindir de ambientes virtualizados.

Mas preciso comprar aquela solução comercial então? Até pode, mas não precisa, pois existem ótimas ferramentas de utilização livre que permitem a virtualização de praticamente tudo. Por que não tudo? Porque os servidores dedicados a análises de grande demanda computacional ainda funcionam melhor rodando diretamente no servidor real. Entretanto, essa tecnologia tem apresentado grande desenvolvimento e, dependendo de quando estiver lendo, talvez possa já usar VM em tudo.

Estão disponíveis data center e estrutura de rede adequados?

Considerando que se possa ter chegado à conclusão de que tem demanda para BioTI, recursos para contratar um profissional de TI e para comprar servidor e *storage*, a próxima e crucial pergunta é: realmente tenho onde colocar os equipamentos? Ou seja, já existe data center instalado, com espaço, refrigeração, segurança e energia adequados para atender à minha demanda? Há espaço em rack, ou espaço para um novo rack? A estrutura de rede (*switches*, cabos, pontos de lógica, etc.) entre esse data center e meus locais de acesso físico é adequada? O profissional que cuidará da infraestrutura terá acesso ao local sempre que necessário? Como será paga a conta de energia elétrica? A rede já possui Firewall? Suas regras são compatíveis à minha demanda? Ou tenho como obter local e recursos para providenciar essa instalação?

Existe central de alto desempenho que possa utilizar?

Se você tem acesso viável a uma central de Bioinformática com servidores de alto desempenho com perfil de multiusuários, mais duas perguntas

precisam ser feitas: a) realmente preciso de infraestrutura local?; b) qual o grau de independência que essa unidade local deve ter?

Como pode ser visto, em todos os pontos levantados até aqui, mesmo antes de detalhar algumas ferramentas (abaixo), está claro que são necessários muitos recursos (humanos e financeiros) para a instalação de uma infraestrutura local de BioTI, e ela só se justifica quando existe uma conjunção de fatores, como: demanda constante por análises de Bioinformática; disponibilidade de bioinformata e de responsável pela TI; e impossibilidade ou velocidade menor que a necessária na utilização de uma central de processamento.

Uma vez que seja definido que, mesmo havendo acesso a uma central de processamento, se faz necessária a unidade local, é preciso identificar o melhor custo/benefício, considerando custo de implantação/manutenção e independência de análises. Naturalmente, demandas pontuais que implicam altíssima performance costumam ser realizadas nessas centrais multiusuário, e a independência da unidade local deve levar em consideração a demanda mediana, ou seja, as análises mais pesadas demandadas com maior frequência. Deve ser considerada também a natural ampliação da demanda e as opções existentes para aquisição de equipamentos e contratação de pessoas. E uma última e importante consideração: é bom manter o plano de manutenção (reposições) e expansão sempre atualizado, pois as oportunidades de compras podem acontecer em janelas de tempo muito curtas, e oportunidades podem ser perdidas.

Ferramental de BioTI

Sistema operacional - distribuições GNU/Linux

O sistema operacional é o software que faz a interface entre o hardware e os aplicativos utilizados pelo usuário. É complexo e incorpora aspectos de baixo nível (como gerência de memória) e de alto nível (como a interface gráfica) (MAZIERO, 2017).

Segundo o *Web Technology Surveys* (W3Techs), dois tipos de SOs são responsáveis por mais de 99,90% dos servidores Web: Windows, com 33,50%; e Unix/Unix-like, com 66,60% (sendo 27,00% Unix e 39,60% GNU/Linux)[4]. Esse mesmo estudo aponta que, entre as distribuições GNU/Linux identificadas, as principais são: Ubuntu (37,80%), Debian (31,10%) e CentOS (20%).

Esse cenário acontece há anos, e alguns motivos são discutidos no portal iDGNow![5], como: maior estabilidade e segurança, não exigência do hardware mais recente e potente (é leve) e liberdade. O predomínio do GNU/Linux em servidores de Bioinformática é ainda maior, principalmente por apresentar muitas opções mais seguras e robustas que são *open source*. Existem ferramentas e linguagens de programação em quantidade, variabilidade e qualidade suficientes para atender a todas as principais demandas dos servidores, e a maioria das ferramentas de Bioinformática foi desenvolvida para SO GNU/Linux. Em outras palavras, se você precisa de um servidor médio a grande para as análises, a melhor opção será o GNU/Linux para a maioria absoluta dos casos. Para demandas pequenas, utilizando pacotes pagos e desktops/notebooks, o SO escolhido afeta menos a performance.

Nossa escolha: **Debian e Ubuntu (versões LTS)** - Como pode ser visto na Figura 1, um recorte da figura de *timeline* de distribuições GNU/Linux, existem dezenas de distribuições, mas algumas são mais antigas e foram a base de criação de diversas outras, possuindo comunidades de desenvolvimento grandes e bem organizadas. As duas grandes famílias de distribuição mais adotadas em BioTI hoje são derivadas do Debian e do RedHat. O RedHat é uma distribuição paga e possui suporte comercial. Como visto anteriormente, o Debian mais sua principal distribuição, o Ubuntu, atingem atualmente 68,90% dos servidores Web com GNU/Linux identificados, e o CentOS, derivado do RedHat, 20%. O RedHat, segundo a W3Techs, atinge 3,10% dos servidores Web[6].

---

[4] https://w3techs.com/technologies/overview/operating_system/all.
[5] http://idgnow.com.br/ti-corporativa/2010/08/31/cinco-motivos-que-colocam-o-linux-a-frente-do-windows-em-servidores/.
[6] https://w3techs.com/technologies/details/os-linux/all/all.

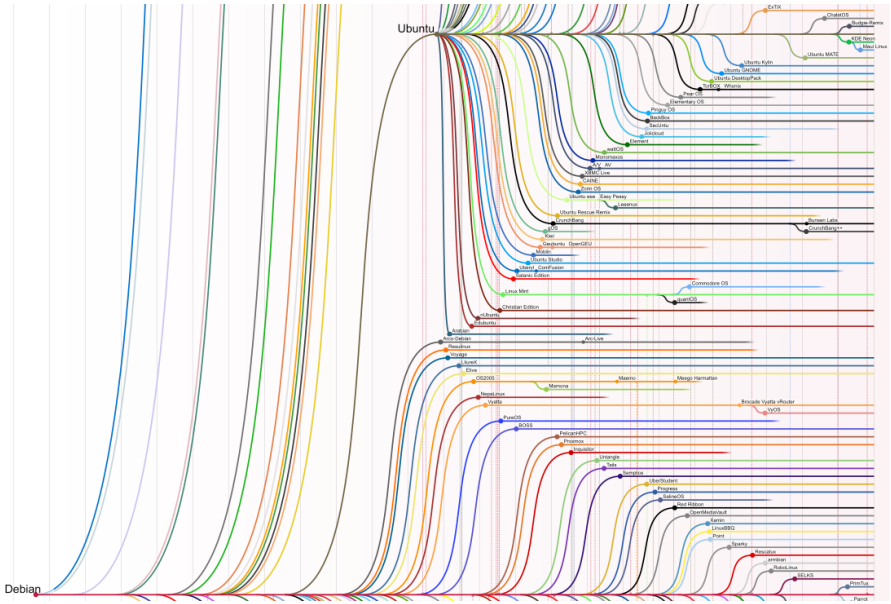


Figura 1. Pedaço da figura "Linux Distribution Timeline", da página da Wikipedia para "Distribuição Linux". Textos "Ubuntu" e "Debian" destacados[7].

Em qualquer das distribuições (link das mais utilizadas na Tabela 1), é recomendada a utilização de versões LTS (*Long-term Support*), que possuem garantia de atualizações de cerca de 5 anos, permitindo que as substituições de versões possam ser realizadas no momento propício para cada data center (ou até de cada servidor).

Tabela 1. Link das principais distribuições GNU/Linux.

Distribuição	Url
Debian	http://www.debian.org
Ubuntu	http://www.ubuntu.com
CentOS	http://www.centos.org
RedHat	http://www.redhat.com/pt-br

[7] https://pt.wikipedia.org/wiki/Distribui%C3%A7%C3%A3o_Linux.

O que diferencia as principais distribuições, e qual é a mais adequada? O CentOS é baseado no RedHat e possui grande parte de suas funcionalidades. A principal diferença é que o RedHat é pago e possui suporte, sendo escolhido preferencialmente em data centers que precisam, por diferentes razões, estar on-line 24x7 (o tempo todo). Debian e Ubuntu (baseado em Debian) são predominantes e gratuitos. As versões do Debian primam pela estabilidade, e as do Ubuntu permitem a inclusão de código menos testados, sendo essa a diferença mais significativa. Entretanto, como as versões LTS possuem datas diferentes de lançamento, às vezes a situação de atualização se inverte.

A adoção de uma em detrimento de outra (incluindo outras diferentes dessas quatro) deve ser baseada na *expertise* local, com a análise de disponibilidade de mão-de-obra qualificada, mais especializada em uma do que em outra. Apesar de todas terem como base ferramentas muito parecidas (em grande parte são as mesmas), existem pequenas diferenças, principalmente na forma de configurar o sistema, que podem dificultar a adoção de uma distribuição quando não se tem a devida base de conhecimento.

Atualmente, em Bioinformática existem grupos adotando a distribuição Ubuntu LTS, com a criação de pacotes específicos e formação de grupos de discussões em torno dela. Muitos outros utilizam Debian. No LBB utilizamos, nos últimos anos, Ubuntu nos servidores para processamento e Debian nos demais (VMs), mas depois de uma análise das versões atuais das diferentes distribuições, está em andamento uma migração/atualização de todos os servidores para a versão Debian LTS mais recente, por trazer uma base mais estável, versões mais atualizadas de pacotes de Bioinformática e uma forte comunidade técnica.

Monitoramento de infraestrutura

Muitas análises de Bioinformática demoram dias ou semanas para serem realizadas, aumentando a importância do monitoramento do ambiente (infraestrutura de hardware e software). As vantagens do monitoramento incluem ganho de tempo, melhor utilização dos recursos, redução de erros e de retrabalho, e ainda obtenção de

métricas que permitem acompanhar a performance e auxiliam no dimensionamento de melhorias na infraestrutura.

Esse monitoramento consiste na utilização de programas especialistas, normalmente com módulo *servidor* e módulo *clientes*, responsáveis por coletar informações em tempo real do ambiente computacional, inclusive da rede de comunicação, armazenar essas informações, tratá-las e utilizá-las para identificar situações de risco para o ambiente, ou de interrupção de serviços. Alarmes podem ser configurados para envio de avisos por e-mail, SMS ou outro meio. Por exemplo, caso ocorra um problema em um disco de armazenamento, ou em um servidor, ou em uma página específica no servidor de aplicações, ou ainda um aumento significativo de temperatura, entre outros eventos monitoráveis, os responsáveis são imediatamente alertados para que possam tomar medidas adequadas.

Nossa escolha: **Zabbix** – é uma ferramenta de Monitoramento largamente utilizada em data centers ao redor do mundo, com mais de 2 milhões de downloads[8]. Possui arquitetura servidor/agente na qual o servidor coleta dados dos Agentes sobre estado de "saúde" dos equipamentos/serviços. Possui interface Web para monitoramento visual e permite configurações diversas, incluindo de envio de notificações (E-Mail e SMS) alertando sobre estados não desejados (sobrecarga, desligamento, sobreaquecimento, etc.). É o sistema de monitoramento utilizado pelo CERN LHC (*Large Hadron Collider*) para coleta de dados experimentais (TELESCA et al., 2014).

Suporta de pequenas a grandes redes de forma gratuita. É uma ferramenta de código aberto, disponível por meio de pacotes em praticamente todas as principais distribuições GNU/Linux. O Zabbix também pode compor estrutura de gerenciamento com SNMP (Protocolo Simples de Gestão da Rede) (SILVA, 2015a, SILVA et al., 2015). Mais informações podem ser obtidas na página do Zabbix[9]. Tutoriais, artigos,

[8] https://www.zabbix.com/files/Brochures/General_Brochure_3.2.pdf.
[9] http://www.zabbix.org.

vídeos e outros tipos de arquivos podem ser encontrados na página da Comunidade Brasileira de Usuários do Zabbix[10]

Existem várias alternativas ao Zabbix, como pode ser visto na Figura 2. Dentre os softwares que apresentam recursos semelhantes, alguns detalhes como experiência da equipe, Sistemas Operacionais suportados, plataforma de desenvolvimento e quantidade de pontos suportados auxiliam na escolha da opção ideal a cada caso. Borges e colegas fizeram uma comparação detalhada de Zabbix e Nagios (BORGES et al., 2015). Apesar de utilizarem versões anteriores às atuais, é uma boa referência em português para aprofundamento no tema.

Name	Yes count	Yes + Plugin (50%)	IP SLA Reports	Logical Grouping	Trending	Trend Prediction	Auto Discovery	Agentless	SNMP	Syslog	Plugins	Triggers / Alerts	WebApp	Distributed Monitoring	Inventory	Platform	License	Maps	Access Control	IPv6	Latest release date (2017)
Hinemos	16	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Full Control	Yes	Yes	Java	GPL	Yes	Yes	Yes	09/06/2017
Zabbix	16	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Full Control	Yes	Yes	C, PHP	GPL	Yes	Yes	Yes	20/04/2017
OpenNMS	16	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Full Control	Yes	Yes	Java	AGPLv3	Yes	Yes	Yes	08/06/2017
Cacti	15	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Full Control	Yes	Yes	PHP	GPL	plugin	Yes	Yes	22/10/2017
Nagios	10	15	plugin	Yes	Yes	No	plugin	Yes	plugin	plugin	Yes	Yes	OK	Yes	plugin	C	GPL	Yes	Yes	Yes	23/02/2017
NetXMS	14	14	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Full Control	Yes	Yes	C++, Java	GPL	Yes	Yes	No	03/04/2017
Icinga	9	14	plugin	Yes	Yes	No	plugin	Yes	plugin	plugin	Yes	Yes	Full Control	Yes	plugin	C[1]	GPL		Yes	Yes	29/03/2017
Netdisco	12	12	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	OK	Yes	Yes	Perl	BSD	Yes	Yes	Yes	2017-10

Figura 2. Adaptação da tabela do item Features, originalmente com 47 softwares[11]. Filtragem realizada para deixar apenas softwares de utilização livre, que tiveram release lançado em 2017, ordenando pela contagem de recursos (colunas "Yes count" e "Yes count + Plugin" adicionadas, com pontuação de "plugin" 50% menor que "Yes"). Nomenclatura normalizada.

SGBD (Sistema Gerenciador de Banco de Dados)

São ferramentas responsáveis pelo gerenciamento de um ou mais bancos de dados para armazenamento e acesso controlado de informações diversas. Os SGBDs de arquitetura relacional são os mais comuns no mercado. Atualmente, os produtos da Oracle Corp. (*Oracle*

[10] http://zabbixbrasil.org/?page_id=7
[11] https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems.

Dabatase) e Microsoft (*SQL Server*) são os líderes do mercado, entre os produtos pagos. Já entre os de código aberto e licença livre, os mais utilizados são o PostgreSQL e MySQL. A revisão "*Bioinformatics Tools in Agriculture: An Update*" (ZAYNAB et al., 2017) descreve o que são os tipos de dados biológicos e como os bancos de dados ajudam a lidar e organizar essa grande quantidade de informação.

PostgreSQL

PostgreSQL é um sistema gerenciador de banco de dados objeto relacional livre, de código aberto, em desenvolvimento há mais de 30 anos, que contém recursos avançados que permitem o armazenamento de grandes quantidades de dados de forma eficiente e segura.

Em termos de recursos e eficácia, está entre um dos melhores bancos de dados disponíveis, estando no mesmo patamar de grandes opções comerciais como IBM DB2 e Oracle SQL. Tem como vantagem sobre outras soluções livres sua ampla gama de recursos que garantem uma escalabilidade em nível corporativo.

Todas as principais distribuições GNU/Linux possuem pacotes prontos para instalar. Mais informações podem ser obtidas na página do PostgreSQL[12]. Tutoriais podem ser encontrados na página PostgreSQL Tutorial[13]

MySQL

É um dos SGBDs mais populares do mundo, sendo facilmente escalável desde pequenas aplicações até grandes e complexos banco de dados. Possui uma vasta gama de ferramentas e interfaces para administração e hoje ocupa uma posição de destaque entre pequenas e médias aplicações. Apesar de ser uma ferramenta de código aberto, é mantida pela empresa Oracle, que direciona o desenvolvimento do MySQL focado em pequenas e médias aplicações deixando o foco das grandes aplicações *enterprise* para sua solução comercial.

---

[12] http://www.postgresql.org.
[13] http://www.postgresqltutorial.com.

Existem casos de uso de MySQL em soluções de grande porte, geralmente com implementações proprietárias de melhorias ou com melhorias específicas[14]. Possui pacote nas principais distribuições GNU/Linux. Mais informações podem ser obtidas na página do MySQL[15]. Tutoriais da versão 5.7 estão disponíveis na mesma página[16].

Nossa Escolha: **PostgreSQL** + **MySQL** – sempre recomendamos que sejam avaliados a sua demanda, a expertise da equipe e os recursos disponíveis na escolha de qualquer solução, mas essa combinação de SGBDs vai resolver praticamente todas as demandas atuais de Bioinformática para SGBD. De modo geral, recomendamos o PostgreSQL, mas existem algumas exceções, como utilização de software que foi desenvolvido com suporte apenas ao MySQL; ou equipe com bom know-how em MySQL somada à demanda que o MySQL suporta. Com os dois, atenderá a demandas de diferentes portes e a maioria das específicas.

Backup

Chamado de cópia de segurança ou redundância de dados, o backup é parte essencial de qualquer data center (ou até estação de trabalho), e é considerado de criticidade alta em termos de Segurança da Informação (SI). Dicas de boas práticas de SI podem ser consultadas no portal do Tribunal de Contas da União[17]. Visa basicamente à recuperação de dados, que devem estar seguros, tanto em termos de integridade quanto em termos de acesso controlado. Os dados devem ser recuperáveis sempre que necessário e, em alguns casos, muito rapidamente. Detalharemos o tema no item "Manutenção".

Nossa escolha: **Bacula** – é uma das ferramentas abertas mais eficientes de gerenciamento para backup e restauração de dados e oferece ampla quantidade de funcionalidades avançadas de gerenciamento de armazenamento, as quais facilitam a recuperação de arquivos

---

[14] https://code.facebook.com/posts/1474977139392436/webscalesql-a-collaboration-to-build-upon-the-mysql-upstream/.
[15] http://mysql.org.
[16] https://dev.mysql.com/doc/refman/5.7/en/tutorial.html.
[17] http://portal.tcu.gov.br/biblioteca-digital/cartilha-de-boas-praticas-em-seguranca-da-informacao-4-edicao.htm.

perdidos ou corrompidos. É uma ferramenta de média complexidade na sua configuração e operação, possuindo poucas interfaces gráficas, mas que disponibiliza uma poderosa interface no modo texto (shell). A versão comercial[18] traz muitas funcionalidades estendidas, mas a versão gratuita é suficiente para a maioria das demandas existentes até em ambientes mais complexos. Possui pacotes nas principais distribuições GNU/Linux. Mais informações podem ser obtidas no blog da ferramenta[19] e na página Bacula do Brasil[20], que contém diversos tutoriais.

Existem diversas alternativas, sendo algumas das mais atualizadas mostradas na Figura 3. Boas alternativas para avaliação no contexto de BioTI são: Amanda[21], BackupPC[22] e Rsync[23]. Rosa e colegas (2015) apresentam uma extensa análise comparativa entre Bacula e Amanda, boa referência em português para mais detalhes sobre o tema, apesar da utilização de versões anteriores na comparação.

Servidor Web

Software servidor Web é o responsável por receber e responder requisições via HTTP na WWW (*World Wide Web*). Normalmente, esse software está instalado num equipamento dedicado (hardware servidor Web), que pode ser um servidor físico ou uma máquina virtual. Segundo a análise da W3techs de outubro, o software para servidor Web mais popular é o Apache (48,30%), seguido por NGINX (34,70%) e Microsoft-IIS (10,70%)[24].

Nossa escolha: **NGINX** – um servidor Web e de proxy reverso (HTTP) de código aberto cada vez mais utilizado, e que, a julgar pela tendência de preferência do mercado (Figura 4), se tornará o mais utilizado dentro dos próximos dois anos.

---

[18] http://www.bacula.com.br/enterprise
[19] http://blog.bacula.org.
[20] http://www.bacula.com.br.
[21] http://www.amanda.org
[22] http://backuppc.sourceforge.net
[23] https://en.wikipedia.org/wiki/Rsync#Variations
[24] https://w3techs.com/technologies/overview/web_server/all.

Package	Yes count	License	Language	Version for Windows	Version for Mac OS	Version for Linux	Graphical user interface	Command line interface	Last updated (stable)
Syncthing	5	MPL	Go	Yes	Yes	Yes	Yes	Yes	setembro, 2017
DAR	4	GPLv2	C++	Yes	Yes	Yes	No	Yes	setembro, 2017
Bacula	5	AGPLv3.0	C, C++	Yes	Yes	Yes	Yes	Yes	julho, 2017
Box Backup	5	BSD/GPLv2.0	C++	Yes	Yes	Yes	Yes	Yes	julho, 2017
AMANDA	4	BSD	C, Perl	Yes	Yes	Yes	No	Yes	junho, 2017
BackupPC	5	GPLv2.0	Perl	Yes	Yes	Yes	Yes	Yes	junho, 2017
duplicity	4	GPL	Python	Yes	Yes	Yes	No	Yes	junho, 2017
Bup	4	LGPLv2.0	Python, Bash, C	Yes	Yes	Yes	No	Yes	junho, 2017
obnam	2	GPLv3	Python	No	No	Yes	No	Yes	junho, 2017
Back In Time	3	GPL	Python	No	No	Yes	Yes	Yes	março, 2017
BorgBackup	3	BSD-3	Python, Cython, C	No	Yes	Yes	No	Yes	março, 2017

Figura 3. Adaptação da tabela "Free software"[25], originalmente com 20 softwares. Filtragem realizada para deixar apenas softwares que tiveram release estável lançado em 2017, ordenada pela coluna "Last update". Nomenclatura normalizada. Adicionada coluna "Yes count".

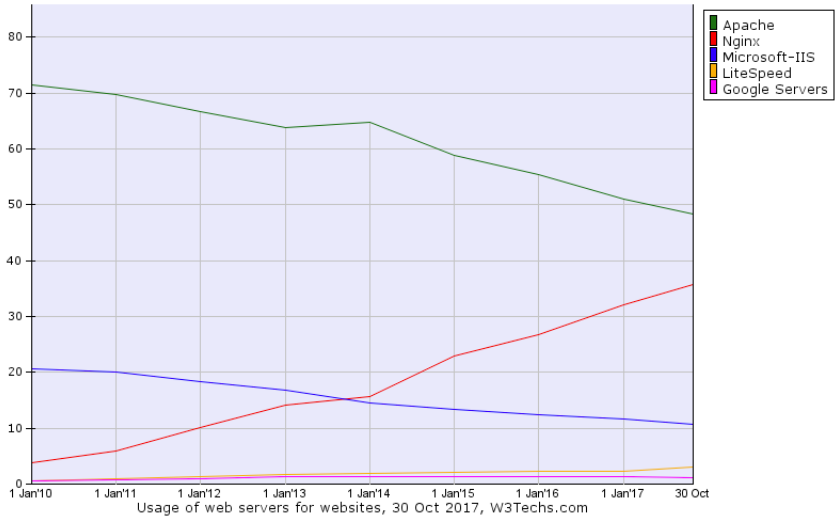


Figura 4. Preferência histórica por servidores Web, anual.
Fonte: World Wide Web Technology Surveys (2017)

[25] https://en.wikipedia.org/wiki/List_of_backup_software.

Essa substituição pelo NGINX se deve à sua simplicidade, robustez e performance, com baixa utilização de recursos computacionais. Exemplo de teste simples e bem descrito pode ser verificado na página do DreamHost[26]. O NGINX ainda permite o *deploy* (disponibilização para cliente final), de forma simplificada, de aplicações Web desenvolvidas em Python (Django), Ruby (Rails), PHP e outras.

A funcionalidade de proxy reverso permite que aplicações e sistema Web estejam espalhados em servidores distintos e tenham um único ponto de entrada e redistribuição concentrando a administração da distribuição de recursos. A principal alternativa ao NGINX é o servidor Web Apache2, ferramenta de código aberto que possui vasta base instalada e muitas pessoas com expertise, mas o Apache é mais pesado, consumindo mais recursos de memória e processamento. O NGINX possui pacotes nas principais distribuições GNU/Linux. Mais informações podem ser obtidas na página Nginx[27]. Um manual para iniciantes está disponível na mesma página[28]

Controle de versões

Um sistema de controle de versões, ou versionamento, permite gestão das diferentes versões de um documento, que no nosso caso são principalmente scripts e códigos de programação. A adoção dessas práticas e ferramentas permite a possibilidade de análise e recuperação de versões anteriores, bem como facilidade e organização de trabalho de diferentes pessoas no mesmo sistema (ou até script), diretamente ou por meio de ramificações (LEPREVOST et al., 2014).

Nossa escolha: **Git (GitLab)** – Atualmente, a ferramenta mais utilizada e versátil para Controle de Versão é o Git, uma plataforma baseada em Web e focada em velocidade. Utilizamos localmente o GitLab. Sua interface permite o registro de solicitações de mudanças, compartilhamento fácil de alterações no código, integração com outras plataformas de controle de versão (GitHub, por exemplo) e um acompanhamento preciso das alterações realizadas em Código-Fonte

[26] https://help.dreamhost.com/hc/en-us/articles/215945987-Web-server-performance-comparison.
[27] https://nginx.org.
[28] https://nginx.org/en/docs/beginners_guide.html.

e/ou nos Documentos. O Git foi criado por Linus Tarvalds em 2005, para o desenvolvimento do kernel do GNU/Linux.

França e Medeiros (2016) apresentam uma detalhada análise comparativa de sistemas de controle de versões, na qual foram comparados Git, Subversion[29] e Mercurial[30]. Concluem que Git é mais utilizado principalmente por ser mais completo e mais rápido, mesmo demandando mais tempo de aprendizado. Se ainda não se convenceu, encontrará uma lista com os principais sistemas na página da Wikipedia[31]. O GitLab possui pacotes para as distribuições Debian e Debian-like, como Ubuntu. Mais informações podem ser obtidas na página do GitLab[32]. O principal tutorial inicial para o Git está disponível em várias línguas na página do Git Project[33]. A Wikipedia traz informação detalhada sobre suas características e links para softwares e informações adicionais[34].

Virtualização

"Virtualizar" é criar uma versão virtual de algo real. Trataremos aqui apenas da virtualização de servidores, na lógica de Consolidação de servidores – várias Máquinas Virtuais (VMs) sendo executadas em uma máquina real. O assunto foi tratado de maneira objetiva por Caciato (2012) e com detalhes técnicos em Maziero (2017).

Algumas das vantagens são: a) redução de espaço físico utilizado e de custo de manutenção do data center (energia, refrigeração, etc.); b) maximização da utilização dos servidores reais; c) facilitação de monitoramento, manutenção e recuperação de desastres; d) expansão e até retração de recursos de servidores de acordo com a demanda; e) migração rápida de VMs para outros servidores reais quando necessário.

---

[29] https://subversion.apache.org
[30] https://www.mercurial-scm.org
[31] https://pt.wikipedia.org/wiki/Sistema_de_controle_de_vers%C3%B5es#Lista_de_sistemas_de_controle_de_vers.C3.A3o.
[32] https://gitlab.com/gitlab-org/gitlab-ce.
[33] https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git.
[34] https://pt.wikipedia.org/wiki/Git.

Nossa escolha: **LIBVIRTD**/**QEMU**/**KVM** – O **KVM** é o módulo nativo do Kernel do GNU/Linux para virtualização, utilizando recursos diretos do processador do servidor. O conjunto de ferramentas **QEMU** possibilita a criação de ambientes virtualizados. O **LIBVIRTD** é um serviço que permite a fácil gestão dessas ferramentas, com interface amigável para administração de Máquinas Virtuais. Esse conjunto de ferramentas está disponível em quase todas as distribuições GNU/Linux e tem-se provado eficiente e eficaz na virtualização. Links na Tabela 2.

Tabela 2. Links das ferramentas de virtualização recomendadas.

KVM	http://www.linux-kvm.org/page/Main_Page
QEMU	https://www.qemu.org
Libvirt	https://libvirt.org

No LBB, hospedamos em dois servidores reais várias VMs específicas para diferentes funções, quais sejam: Arquivos, Web, Backup, MySQL, PostgreSQL, Proxy Reverso, Zabbix e GitLab.

O site de referência para a implementação e uso do LIBVIRTD possui excelente documentação[35].

Existem várias alternativas, e você poderá analisar diferentes tabelas comparativas[36].

Desenvolvimento Web

O desenvolvimento de páginas e aplicativos Web não é o foco, mas é uma ferramenta importante para a Bioinformática, para a disponibilização de resultados de pesquisa ou para sistemas interativos que se tornam parte do resultado. Existem muitas plataformas e abordagens de desenvolvimento, e o know-how de cada equipe tem um peso muito grande nessa decisão. Portanto, trataremos aqui do nosso contexto, de algumas tendências e da nossa escolha.

[35] http://wiki.libvirt.org/page/Main_Page
[36] https://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software.

O contexto de um laboratório de pesquisa em Bioinformática normalmente inclui: novas tecnologias e análises que surgem frequentemente; demanda por resultados maior que a oferta de tempo de bioinformatas; boa parte das equipes não tem graduação em TI; parte da equipe formada por bolsistas fica meses ou poucos anos no grupo, e normalmente são pouco experientes. Em resumo: pouco tempo, alto *turnover*, demanda constante por treinamento e bolsistas com experiências em diferentes linguagens. A partir desse contexto, sugerimos que as ferramentas de desenvolvimento adotadas possuam características como as descritas na Tabela 3.

Tabela 3. Características desejáveis a ferramentas de desenvolvimento Web voltadas a laboratórios de pesquisa em Bioinformática.

• Rápida para aprender e desenvolver.
• Linguagem, abordagens e melhores práticas devem gerar códigos limpos e de fácil entendimento e manutenção, mesmo quando não dá tempo de realizar documentação detalhada (altamente recomendada, mas não prioritária em algumas abordagens).
• Comunidade grande e ativa, com perspectiva de continuidade e crescimento.
• Flexível e abrangente – ser utilizada em diferentes escalas e/ou contextos.
• *Open source*, e ter ampla documentação gratuita.

Sobre tendências, cada vez mais se buscam abordagens e ferramentas que permitam desenvolvimento rápido, flexibilidade e agilidade para modificar o sistema, facilidade de manutenção, etc. Abordagens mais recentes, como desenvolvimento ágil, tem ganhado espaço. Apesar do legado e do lobby das grandes empresas, convencendo o alto escalão das empresas a comprar (e as universidades a ensinar) Java, opções Microsoft, etc., métodos preditivos e robustez têm perdido espaço para métodos adaptativos e ferramentas ágeis. Embora ainda menor no Brasil, essa mudança segue forte lá fora. Por exemplo, a Tabela 4 traz alguns casos de sites e aplicativos famosos desenvolvidos em Python e Django. Uma lista

maior pode ser verificada na página do Python[37]. Outra realmente grande de sites em Django pode ser acessada no DjangoSites[38].

Tabela 4. Empresas e aplicativos que têm site (ou em que o próprio aplicativo é) desenvolvido total ou parcialmente com Python ou Django[39].

Python	Youtube, Dropbox, Google, Yahoo Maps.
Django	Pinterest, Instagram, Spotify, Mozilla, Nasa, Prezi.

Nossa escolha: ***framework* Django/Python** – é uma ferramenta com todas as características descritas na Tabela 3 e com relevância e potencial que atendem com sobra às demandas de desenvolvimento por bioinformatas, como pode ser visto na Tabela 4.

O Django utiliza arquitetura MVT (*Model-View-Template*), adaptada do MVC (*Model-View-Controller*). Entre suas vantagens, podemos citar que é tecnologia aberta com amplo suporte de comunidade de desenvolvedores (incluindo fóruns e códigos-fonte), tem sintaxe do código limpa e pragmática, sendo possível rápido desenvolvimento e fácil manutenção. Outra vantagem é o fato do Python estar sendo cada vez mais utilizado em Bioinformática, substituindo o Perl, inclusive com o aumento das funcionalidades do Biopython, reduzindo a curva de aprendizado e facilitando o intercâmbio de soluções entre ferramentas, scripts e sistemas Web que utilizam Django.

Referências para Django podem ser encontradas no blog de Eric Hideki[40]. Um ótimo livro on-line com informações complementares à documentação oficial pode ser acessado na página do Guia para Python[41].

[37] https://wiki.python.org/moin/OrganizationsUsingPython.
[38] https://djangosites.org.
[39] https://www.shuup.com/en/blog/25-of-the-most-popular-python-and-django-websites, http://djangostars.com/blog/10-popular-sites-made-on-django.
[40] https://ericstk.wordpress.com/2013/05/08/django-fontes-de-estudo-e-referencias.
[41] http://python-guide-pt-br.readthedocs.io/pt_BR/latest.

O Django leva o desenvolvedor a utilizar de boas práticas e define um conjunto de regras para estruturação e codificação de projetos, como é descrito na página de melhores práticas para Django[42]. O site do Django também provê um excelente tutorial de início[43].

Linguagens de Programação (scripts e programas)

Para instruir o computador a fazer o que queremos, utilizamos diversas linguagens de programação. Visando a aumento de produtividade para os casos mais comuns, os bioinformatas trabalham com linguagens de programação de alto nível, mais distantes do código de máquina, como Perl e Python. Por outro lado, cada vez mais, existem demandas pesadas que exigem linguagens compiladas, como o C/C++, para maior eficiência. Encontrará detalhes e classificações sobre o tema na página da Wipedia[44].

Nossas escolhas: **Python, Perl, Shell Script, R e C++.** Não custa repetir que o know-how da equipe e as especificidades da demanda local são essenciais na escolha, mas essas são as sugestões baseadas na nossa experiência e nas nossas demandas. Python, shell script e R para o dia-a-dia, Perl e C++ quando necessário.

Shell script

"O shell é o 'prompt' da linha de comando do Unix e GNU/Linux, é o servo que recebe os comandos digitados pelo usuário e os executa". E shell script é "um arquivo de texto que contém comandos do sistema e pode ser executado pelo usuário" (JARGAS, 2017). Scripts podem também ser definidos como arquivos executáveis, com instruções definidas que são executadas por um interpretador (podendo ser o sh, zsh, ksh ou o mais comum em distribuições GNU/Linux: Bash).

O shell script facilita consideravelmente a vida e o trabalho do usuário. Automatização de tarefas é refletida em aumento de velocidade e facilidade. Em análises de Bioinformática, essa automatização é

---

[42] https://django-best-practices.readthedocs.io/en/latest/applications.html
[43] https://docs.djangoproject.com/en/1.11/intro/tutorial01
[44] https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o.

mandatória para o bom rendimento dos trabalhos. Existem diversos tutoriais disponíveis para tarefas simples e complexas em shell script[45]. Aurélio Jargas mantém um blog com documentação extensa sobre shell script, desde os primeiros passos até a utilização profissional do mesmo[46]. Destacamos o "Canivete Suíço do Shell (Bash)"[47], com tabelas de referência rápida a operadores, variáveis especiais, formatadores, etc.

Perl

Primeira das linguagens interpretadas a ser largamente utilizada na Bioinformática[48]. Possui uma vasta variedade de bibliotecas (módulos) desenvolvidas para o processamento de texto, tornando-a bastante flexível no tratamento de dados comuns na Bioinformática. Tem uma grande comunidade de apoio ao desenvolvimento e uso[49].

Existe um conjunto de módulos em Perl chamado Bioperl[50] (STAJICH et al., 2002), que traz uma série de funcionalidades e tipos de dados especificamente para lidar com arquivos e formatos comuns na Bioinformática.

Existem muitos scripts prontos e muitos bioinformatas com know-how nessa linguagem, o que torna interessante ter algum conhecimento ao menos para entender e fazer pequenas modificações em scripts de terceiros. Tutoriais disponíveis na página Tutorials Point[51].

Python

Linguagem de Programação Orientada a Objetos que está se tornando padrão no desenvolvimento das principais ferramentas e soluções para Bioinformática utilizando-se do *framework* Biopython. É a linguagem a ser aprendida prioritariamente.

---

[45] https://www.shellscript.sh.
[46] http://aurelio.net/shell/
[47] http://aurelio.net/shell/canivete/
[48] https://web.stanford.edu/class/gene211/handouts/How_Perl_HGP.html.
[49] https://learn.perl.org/tutorials/
[50] http://bioperl.org/
[51] https://www.tutorialspoint.com/perl/.

Python é hoje uma das principais linguagens de programação[52] e possui uma grande comunidade de desenvolvedores, facilitando o aprendizado por meio de fóruns, sites e livros[53].

Assim como Perl, possui uma ampla e variada gama de funcionalidades para processamento de texto, o que a torna uma linguagem bastante versátil no tratamento de dados de Bioinformática.

Biopython (COCK et al., 2009) é um conjunto de funcionalidades (*framework*) criado para facilitar o desenvolvimento de ferramentas de Bioinformática. Vem sendo adotado como *framework* padrão para o desenvolvimento de soluções em Bioinformática e possui uma excelente documentação[54].

R

O R foi criado como uma alternativa de código aberto à linguagem S nos anos 1990 (IHAKA; GENTLEMAN, 1996), sendo cada vez mais utilizado por cientistas e estatísticos para a análise de dados.

Um dos pontos fortes do R é a facilidade na geração de imagens (vetoriais ou rasta) de alta qualidade, inclusão de símbolos matemáticos e de fórmulas quando necessárias, já que o R é um conjunto integrado de facilidades de software voltado para manipulação de dados e exibição gráfica.

A linguagem R pode ser utilizada por meio de uma interface multiplataforma mais amigável chamada RStudio[55], entre tantas outras interfaces. O RStudio torna o trabalho com R ainda mais fácil, de forma a nos oferecer teclas de atalho, conclusões de código, gerenciamento de janelas e interações gráficas que podem ser utilizadas em vez de chamar funções complexas. Uma lista de pacotes e ferramentas para R, incluindo interfaces no item *Integrated Development Environments*[56].

---

[52] http://www.tiobe.com/tiobe_index
[53] http://greenteapress.com/wp/think-python/)
[54] http://biopython.org/DIST/docs/tutorial/Tutorial.html
[55] https://www.rstudio.com/
[56] https://awesome-r.com

A documentação é vasta e existem pacotes para as mais diversas análises de dados, incluindo muitas opções para Bioinformática. Um bom ponto de partida para entender melhor a linguagem e suas capacidades é pela própria página da iniciativa[57] ou por livros gratuitos como o "R for Data Science"[58].

C++

Diferente de Python e Perl, linguagens interpretadas, a linguagem de programação C++ é compilada nativamente, trazendo inúmeros ganhos no quesito performance e baixo uso de recursos.

Como desvantagem, C++ traz uma sintaxe bastante complexa para novos programadores. Não existe também um conjunto de funções (*framework*) voltado especificamente para atividades de Bioinformática. Porém, muitas funções comuns ao processamento de texto são encontradas na linguagem e na biblioteca padrão.

Material sobre desenvolvimento em C++ pode ser encontrado facilmente na internet, possuindo milhares de sites, livros e tutoriais espalhados. Recomendamos iniciar estudos no site Cplusplus[59].

Infraestrutura física

Por infraestrutura física, considera-se todo o conjunto de equipamentos, a infraestrutura para a comunicação de dados – cabeada ou *wireless*, a infraestrutura para o provimento de energia elétrica, refrigeração, segurança e o ambiente onde o conjunto principal de racks, servidores, unidades de armazenamento e ativos de rede está instalado.

[57] https://www.r-project.org/.
[58] http://r4ds.had.co.nz.
[59] http://www.cplusplus.com/doc/tutorial.

Equipamentos

Servidores

Servidores são computadores especialistas utilizados para a disponibilização de serviços, como: a) processamento de rotinas em lote; b) controle do armazenamento e disponibilização de arquivos; c) execução de aplicações em um modelo cliente-servidor; d) execução de aplicações Web; e) execução de softwares de alta demanda computacional (principal utilização de servidores em Bioinformática).

Qualquer computador, seja real ou virtual, pode ser utilizado como um *servidor*, mas é essencial que o equipamento seja compatível com a demanda e os prazos de entrega. Em Bioinformática, diversas análises exigem um grande volume de memória, processadores de alta velocidade e com vários núcleos, e um sistema de I/O de bom desempenho, ou seja, servidores de médio a grande porte, idealmente em rack.

Para garantir performance e alta disponibilidade, é desejável que os servidores contem com memória e processadores de excelente qualidade, fontes redundantes, discos *hot swapping* (substituíveis com sistema funcionando) e garantia com prazos de solução compatíveis com sua necessidade. Também é importante considerar a velocidade de conexão com a rede e a oferta de mais de uma porta de conexão com a rede cabeada.

Na aquisição de servidores, recomenda-se preferir boas marcas estabelecidas no mercado. São equipamentos que ficarão ligados 24x7 (vinte quatro horas por dia, sete dias por semana), e que exigem bom desempenho e confiabilidade. Entre os principais fornecedores, temos experiências prévias e recomendamos, quando for possível a escolha (links na tabela 5):

- Dell Computadores
- HP - *Hewlett Packard Enterprise*

- IBM: a IBM vendeu uma parte de seu negócio de servidores para a Lenovo. Entretanto, continua comercializando alguns modelos, inclusive voltados para o ambiente GNU/Linux.

Tabela 5. Link dos principais fornecedores de servidores.

Dell	http://www.dell.com/br/empresa/p/enterprise-products?~ck=anav
HP	https://www.hpe.com/br/pt/servers.html
IBM	https://www.ibm.com/linuxone

É possível encontrar ajuda para o dimensionamento da solução para cada caso também nos sites de fabricantes e fornecedores, como o "Guia para comprar servidor"[60].

Nossas considerações: A aquisição de servidores sempre levou em consideração a relação custo/benefício, ou seja, um equilíbrio entre modelos que não são necessariamente top de linha (normalmente são desproporcionalmente caros), mas que ao mesmo tempo permitam tanto expansibilidade quanto a execução dos serviços em tempo hábil por vários anos. São modelos para instalação em rack, o que otimiza a utilização do espaço no data center. O esforço foi para contratar com maior tempo de garantia possível, o que varia entre 3 e 5 anos.

Storages

Storages são unidades especialistas de armazenamento de dados. Por serem dedicadas a esse fim, contam com vários recursos para prover armazenamento eficiente e rápido de grandes volumes de dados.
São compostos basicamente de unidades controladoras e gavetas de discos, que podem ser de vários tamanhos, capacidades e tecnologias. Também é importante considerar qual tipo de ligação oferece para a conexão com os servidores da rede.

[60] http://www1.la.dell.com/content/topics/segtopic.aspx/pt/dell-server-basics-buy-guide?c=br&l=pt&cs=brbsdt1.

Normalmente oferecem grande capacidade de expansão, por meio da inclusão de gavetas onde podem ser acrescentados novos discos. Esse é um detalhe importante no momento da especificação: verificar qual a capacidade para receber novas gavetas por controladora. Outra maneira de expandir a sua capacidade é trocar os discos por discos de maior armazenamento. Também é importante verificar quais os tipos de discos e qual a capacidade total — em se utilizando os discos de maior espaço possível.

Também são oferecidos em formato de torre ou adequados para rack. Esta última opção é recomendada, por facilitar a montagem do ambiente e ocupar mais racionalmente o espaço no data center.

Em relação aos discos utilizados, estes podem ser os listados a seguir. Mais informações sobre SSD podem ser obtidas na página Info Wester[61]. Informações sobre outros tipos de discos foram descritas por Faria (2017).

- SATA: *Serial* ATA, SATA ou S-ATA. São discos que utilizam um cabo serial de 7 pinos, relativamente estreito, o que facilita a instalação e a refrigeração do equipamento. Existem modelos que possuem a capacidade *hot swapping.* Têm preço mais acessível. Evoluíram do padrão IDE.
- SAS: *Serial Attached* SCSI. Oferece compatibilidade com o padrão SATA versão 2 ou superior. É uma evolução do padrão SCSI. É normalmente utilizada em *storages* e servidores mais sofisticados e de preço mais elevado, uma vez que fornece uma série de serviços não oferecidos no padrão SATA.
- SSD: *Solid-State Drive.* Difere totalmente dos discos tradicionais, pois são unidades de memória flash de grande capacidade, sem partes móveis. Por isso, a velocidade de acesso e gravação é muito superior. O preço elevado ainda é um limitador à sua maior utilização, mas tende a diminuir. Atualmente, são oferecidos em capacidades que vão de 64 Gb a 2 Tb.

[61] https://www.infowester.com/ssd.php.

Quanto à arquitetura, *storages* podem ser:

- DAS: *Direct-attached Storage*. Conecta-se diretamente ao servidor, não utilizando uma rede compartilhada para isso. Por ser uma ligação direta, pode utilizar mecanismos de comunicação muito velozes. Como desvantagem, não permite a ligação com vários servidores.
- NAS: *Network-Attached Storage*. Conecta-se com os servidores através de uma rede cabeada. Este tipo de implementação permite a centralização de dados e o seu compartilhamento entre vários servidores, melhorando a organização e a gestão do ambiente.
- SAN: *Storage Area Network*. Também se conecta com os servidores através de uma rede, mas é uma estrutura mais robusta e de melhor desempenho. É utilizado em grandes data centers como uma opção sofisticada de armazenamento de dados e seu compartilhamento. Une a alta velocidade de transmissão que pode ser obtida por tecnologia DAS com a flexibilidade e o compartilhamento característicos da tecnologia NAS.

Links para mais informações sobre a arquitetura de *storages* na Tabela 6.

Tabela 6. Links para posts na Wikipedia sobre arquitetura de storages.

DAS	https://pt.wikipedia.org/wiki/Direct_Attached_Storage
NAS	https://pt.wikipedia.org/wiki/Network-Attached_Storage
SAN	https://pt.wikipedia.org/wiki/Rede_de_área_de_armazenamento

Para oferecer alta disponibilidade e performance, *storages* devem contar com fontes redundantes, várias portas para comunicação e tecnologia RAID, que, combinada com discos *hot swapping*, permite a substituição de discos defeituosos sem que se desligue o equipamento. A tecnologia RAID também oferece suporte para várias configurações de montagem de discos, o que pode ser utilizado para obter desempenhos

superiores e alta disponibilidade do ambiente de armazenamento. Mais informações sobre RAID podem ser obtidas no site GlobalMind[62].

Outra tecnologia que pode estar presente nos *storages* é a deduplicação. Esta é uma técnica que utiliza camadas de software para organizar os blocos físicos de maneira a evitar redundâncias, o que pode aumentar de maneira drástica a capacidade total de armazenamento, além de oferecer diversas ferramentas para a gestão dos dados armazenados. Entretanto, essa tecnologia ainda não está presente nos *storages* de preço mais acessível, limitando-se aos equipamentos de custo superior. Informações complementares no site da Dell EMC[63].

Outro aspecto importante a ser considerado é o consumo energético, e Silva (2015b) faz uma análise detalhada da eficiência energética relacionada ao armazenamento de dados, discutindo diferentes aspectos da TI Verde. Por exemplo, mostra que discos com maior capacidade gastam menor energia para o mesmo total de dados (um disco de 1 TB gasta menos energia que dois discos de 0,5 TB, por exemplo). Trata também como bons redutores de consumo energético a virtualização, a deduplicação e as melhores práticas de gestão.

Nossas considerações: Utilizamos *storages* com discos SAS, *hot swapping*, com boa capacidade de expansão e com fontes redundantes. Inicialmente compramos *storage* com conexão a servidores através de fibra ótica, mas atualmente optamos por iSCSI, de velocidade satisfatória e melhor relação custo/benefício. Pretendemos comprar alguns discos SSD para análises específicas com alta demanda de I/O. Além da quantidade de dados e da demanda por I/O, vários aspectos precisam ser considerados, como: políticas de backup, melhores práticas de gestão, otimização de consumo energético, rede e virtualização.

---

[62] http://www.globalmind.com.br/entenda-o-que-e-raid-e-sua-importancia-para-performance-eou-seguranca-dos-dados/.

[63] https://brazil.emc.com/corporate/glossary/data-deduplication.htm.

Redes de computadores

Uma rede é um conjunto de equipamentos de computação interligados de modo a poder compartilhar informação e recursos, seja a conexão entre o celular e a TV, seja toda a intranet de uma grande Empresa. Na Wikipedia[64] estão descritos sistemas de classificação, hardwares, modelagem, topologia, etc. Trataremos diretamente de poucos aspectos neste documento.

Os equipamentos de BioTI normalmente estão dentro de uma LAN ou de uma VLAN. A maior parte do tráfego de dados ocorrerá entre os servidores e os *storages*. Entretanto, as ligações entre o data center e as estações de trabalho precisam ter qualidade suficiente para não impactar no desempenho de todo o processo.

A preferência deve ser por redes cabeadas, instaladas no padrão de cabeamento estruturado, categoria 6. As normas NBR 14565[65] e EIA/TIA-568-B[66] disciplinam a instalação de redes de dados. Nesse padrão, são utilizados conectores RJ45, cabos UTP, *patch panels*, *switches* de borda e *switches* core para montar toda a rede de dados.

Os *switches, ou comutadores*, aparelhos responsáveis por realizar a interligação dos diferentes equipamentos de uma rede, organizar o tráfego e definir diferentes camadas e níveis de acesso são a parte central dessas redes.

Os mais simples contam somente com uma interface de controle, normalmente Web, e permite somente a definição de uma rede. Os *switches* gerenciáveis já permitem a implementação de VLANs (*Virtual Local Area Network* - rede local virtual), que, se bem projetadas, podem aumentar consideravelmente o desempenho e a segurança do ambiente de redes. Os mais completos são os *switches core*, equipamentos de maior porte que normalmente fazem a ligação da rede local, ou das redes locais, com a WAN (*Wide Area Network* - rede de longa

[64] https://pt.wikipedia.org/wiki/Rede_de_computadores.
[65] http://www.abntcatalogo.com.br/norma.aspx?ID=307178.
[66] https://pt.wikipedia.org/wiki/EIA/TIA-568.

distância). Oferecem uma maior variedade de portas, de velocidades de conexão e de gerenciamento. Vários aspectos devem ser considerados na escolha do switch mais adequado, como os apresentados na página da DL Tec do Brasil[67].

Outros aspectos devem ser considerados na estruturação da sua rede, como a inclusão de proxy reverso em uma DMZ (Zona desmilitarizada), a inclusão e a configuração de Firewall, o equilíbrio entre segurança e produtividade, os testes de vulnerabilidade, estabilidade, as auditorias, e um bom material para começar a aprofundar a leitura é o trabalho de Carlos Marocco (2015), utilizando softwares livres.

Nossas considerações: Durante a elaboração do projeto do ambiente computacional, se já não existir, deve ser considerada pelo menos a implementação de uma rede de comunicação mínima para suportar a interligação entre os servidores, *storages* e as estações de trabalho. *Switches* de borda gerenciáveis normalmente são suficientes para a instalação inicial, por exemplo, com 24 portas para conectores RJ45 e quatro portas SFP (fibra), todas com velocidade 1Gbps. Cabos categoria 6, se possível de marca reconhecida, são recomendados. O LBB está na mesma rede da unidade, mas com seus equipamentos do rack e suas estações de trabalho em uma VLAN separada. O proxy reverso está alocado em uma DMZ externa.

Unidades de fitas automatizadas

Unidades de fitas são equipamentos para armazenamento de dados digitais em fitas magnéticas, podendo ser automatizados (como a troca automática de fitas). Em uma análise rápida, podemos afirmar que no passado recente foi a opção com melhor custo/benefício para backup, mas, com a redução do preço dos HDs, muito mais rápidos, deixou de ser viável para a maioria das utilizações, principalmente pela baixa velocidade de leitura de escrita de dados específicos (leitura sequencial) e pelos cuidados especiais para armazenamento das fitas, apesar da maior durabilidade média. Na página da Dell, foi descrita uma análise

[67] http://www.dltec.com.br/blog/redes/melhor-switch-para-sua-lan

comparativa sobre as diferentes mídias para backup, com prós e contras de cada tipo[68].

Assim, as soluções modernas para backup oferecem armazenamento em disco. Está disponível uma série de equipamentos especializados em backup, e o custo/benefício, junto com o orçamento disponível, é que vai determinar o porte dessa solução.

Nossas considerações*:* A infraestrutura do LBB inclui uma unidade de fita automatizada, comprada quando discos ainda eram muito caros, na qual estão sendo armazenados nossos backups. Entretanto, muitas fitas já estão indicando erro, e solicitamos compra de *storage* para realizar backup em disco. Como já temos a unidade de fitas, continuaremos com o backup em fita para termos duplicação do backup, mas, para o contexto médio de BioTI, não recomendamos a aquisição de unidades de fita para esse fim.

Racks

Um rack é um tipo de armário padronizado para organização de servidores, *storages*, *switches* e outros equipamentos de TI. Os mais comuns têm largura de 19' (482,6 mm) e profundidade de 17,7' (449,58 mm), e um número variável de Us (Unidades de rack), sendo que cada U corresponde a 1,75' (44,45 mm). Esse padrão de tamanho é utilizado para os equipamentos de rack. Por exemplo, um servidor de 19' e 3Us tem 5,25' (133,35 mm) de altura.

Os *racks* mais sofisticados têm características como: espaço específico para a fiação de rede, a fiação elétrica, portas reversíveis e facilmente removíveis e possibilidade de colocação de sensores de monitoramento.

Nossas considerações: a utilização de rack é muito importante, não apenas para economia de espaço, mas também como auxiliar para controle de acesso físico, para facilitar a refrigeração pelo posicionamento de cabos e pelo fluxo de ar, pela redução de riscos de acidentes e quebra de fios e conectores, etc.

---

[68] http://www.dell.com/learn/br/pt/brbsdt1/sb360/article-disk-vs-tape.

Data center

Um data center é um local projetado para armazenar os principais ativos de *hardware* do ambiente computacional da organização, como: servidores, unidades de armazenamento - *storages*, e ativos de rede - *switches core, switches* de borda, roteadores. Os data centers devem permitir o funcionamento dos equipamentos de modo seguro e estável, 24 horas por dia, sete dias por semana.

Quando se fala em BioTI, existe tanto a necessidade de funcionamento *online* 24x7 quanto do processamento de longas rotinas, que podem durar semanas. Portanto, os data centers para BioTI devem contar com uma rede elétrica adequada para o fornecimento de energia para todo o equipamento, estabilizada e redundante. Essa rede elétrica deve fornecer proteção a falhas, o que pode ser obtido com a utilização de *nobreaks* e geradores secundários de energia.

Deve contar também com um sistema de refrigeração adequado, redundante e com monitoramento externo, pois um superaquecimento pode causar danos a dados armazenados e aos equipamentos, muitas vezes com perda total.

É importante também utilizar sistemas de prevenção de eventos, como incêndios, níveis de umidade inadequados e inundações. Estão disponíveis no mercado diversas soluções que integram sensores, sistemas de monitoramento e alarmes. Também é possível o desenvolvimento de soluções locais, utilizando técnicas de automação predial, sensores de custo mais acessível e plataformas como o Arduíno e o Raspberry.

Os data centers podem ser classificados em relação ao seu nível de segurança e disponibilidade, através da certificação TIER e da norma TIA 942[69].

[69] https://pt.wikipedia.org/wiki/Padr%C3%A3o_TIER.

Nossas considerações: embora inicialmente instalado separadamente, o rack do LBB foi realocado no data center da unidade, visando economia de recursos e ganho de segurança. Considerando que, normalmente, não é possível instalar de uma vez um data center com o máximo de qualidade e segurança, a opção recomendada é que as melhorias sejam feitas sempre que possível, podendo seguir como modelo a norma TIA 942.

Manutenção

Data center parado significa prejuízo, maior ou menor, de acordo com cada caso, mas o ideal é que ele só pare de maneira planejada, e o mínimo de tempo possível. Para tanto, ele precisa ser confiável, disponível e ter redundância. Isso envolve a refrigeração, a energia, a proteção contra desastres e a mitigação de desastres, a boa organização de cabos e afins, e tudo isso sem esquecer a importância da qualificação da equipe e da eficiência energética. Existem inúmeros trabalhos sobre o tema de organização e manutenção de infraestrutura, como: Frigo (2015), Santos (2014) e Driemeyer (2016), que tratam inclusive de refrigeração e energia, e muitas soluções comerciais, normalmente caras e só viáveis se adotadas institucionalmente.

Detalharemos a seguir alguns aspectos importantes, que podem estar sob a responsabilidade da equipe de BioTI.

Auditorias - esteja preparado

Auditorias são avaliações sistemáticas para verificar a conformidade de uma série de normas específicas a cada caso, realizadas tanto por órgãos internos quanto por externos, normalmente seguindo roteiros pré-estabelecidos. Nas instituições públicas, utilizam normas do Governo Federal[70]. Se sua infraestrutura for passível de auditoria, é necessário buscar informações acerca das respectivas normativas.

[70] https://www.governoeletronico.gov.br/eixos-de-atuacao/governo/sistema-de-administracao-dos-recursos-de-tecnologia-da-informacao-sisp/legislacao.

Gestão e Monitoramento – constantes e completos

O gerenciamento da infraestrutura e dos recursos de TI consiste em uma série de procedimentos para administrar de maneira efetiva e eficaz a utilização dos bens e serviços relacionados à TI. Tem por objetivo maximizar o desempenho, garantir o pleno funcionamento de toda a infraestrutura e permitir a melhor utilização possível do parque computacional.

A questão da disponibilidade financeira é uma das mais importantes. A instalação de uma infraestrutura de TI exige um considerável aporte de recursos, mas não basta pensar em aquisição e instalação, já que gerenciamento e manutenção do ambiente envolvem também recursos humanos e financeiros. A falta de gerenciamento adequado coloca em risco a continuidade do serviço e a integridade física do ambiente, com a possibilidade de descontinuidade de serviços, atrasos ou impossibilidade de entregas e, ainda, perda de equipamentos e infraestrutura.

Portanto, se não houve previsão anterior de recursos para o gerenciamento, é importante buscar alternativas para viabilizá-lo, ou, na pior das hipóteses, garantir recursos para reparos e correções emergenciais. Vale lembrar-se nesse momento do custo de uma equipe parada por falta de recursos, do custo de atrasos e da perda de dados, que pode comprometer partes ou mesmo projetos inteiros.

Considerando que existam recursos, a próxima preocupação é com o monitoramento constante e completo dos ativos, visando principalmente o funcionamento contínuo e em alta performance do ambiente computacional. O monitoramento envolve a verificação de disponibilidade do recurso e de capacidade de atender as demandas existentes.

Conforme descrito em tópico anterior, a ferramenta *open source* de referência para esse monitoramento é o Zabbix. As suas telas e relatórios podem ser customizadas de maneira a oferecer apresentações mais adequada a cada equipe, ou para situações específicas. Por

exemplo, se em um determinado momento os processos executados sabidamente consomem muito espaço em disco, uma tela do Zabbix pode ser configurada para apresentar as estatísticas de I/O, em forma de gráficos diversos ou de tabelas. Pode também apresentar gráficos que expressem o crescimento do espaço ocupado, e ainda gráficos mostrando o percentual de espaço livre. A mesma lógica é aplicável para processamento, memória RAM e SWAP, uso de rede, etc. Além da documentação do aplicativo, existem diversos trabalhos em português que tratam da utilização da ferramenta, como: Silva (2015a), Silva et al. (2015) e Soares e Costa (2014).

É recomendável que se faça um projeto com todos os itens que se pretende monitorar e os equipamentos que serão considerados, bem como com os tipos de saída que se pretende implementar. São muitas as possibilidades de relatórios e telas de consulta.

Isso feito, o próximo passo é instalar a ferramenta em um servidor, preferencialmente uma máquina exclusiva para esse fim (normalmente uma VM). Considerando que seu objetivo é monitorar todo o ambiente, não é prudente deixá-lo junto com qualquer outro serviço. Depois, os agentes precisam ser instalados nos equipamentos que serão monitorados. Isso feito, o próximo passo é a implementação dos *triggers*. Como uma tradução livre já diz, são "gatilhos" que podem disparar alertas, ações ou mesmo outros *triggers*, quando uma condição ou evento previstos ocorrem no ambiente computacional[71].

O próprio ambiente de monitoramento é dinâmico. Dificilmente a primeira implementação cobrirá todo o espectro possível. A própria experiência da equipe no uso da ferramenta e o acompanhamento dos eventos provocará alterações no seu modelo, que deverá ser sempre aperfeiçoado buscando apresentar as informações necessárias para permitir uma gestão o mais eficiente possível da infraestrutura computacional.

[71] https://m2networks.com.br/trabalhando-com-triggers-no-zabbix/.

É recomendável que todo o ambiente seja monitorado. Desde a disponibilidade e o tráfego na rede, a saúde dos equipamentos, o percentual de sua utilização, a disponibilidade dos sistemas em produção até informações de ambiente, como temperatura do data center, nível de umidade, presença de fumaça, risco de incêndio. A distância entre o que é recomendável e o que é possível passa pelo aporte de recursos, notadamente financeiros e de pessoal especializado. No nosso caso, temos o Zabbix monitorando o ambiente computacional, e um equipamento adquirido para monitorar aspectos físicos do data center, composto por um conjunto de sensores que registram temperatura, umidade, risco de incêndio e câmeras de segurança com CFTV. Apesar de essa estrutura registrar o risco de incêndios, estamos buscando recursos para a obtenção de equipamentos para a extinção automática de fogo.

Também para o futuro, pretendemos utilizar mais recursos do Zabbix na direção de uma gestão mais proativa. Com a utilização de agentes SNMP, é possível programar a ferramenta para disparar ações de correção e alteração do ambiente, além de gerar alertas. Por exemplo, é possível disparar um evento de remoção de arquivos antigos caso o espaço ocupado em disco atinja um determinado patamar (SILVA et al., 2015).

Atualização de SOs – segurança e compatibilidade

As atualizações de sistema operacional são um conjunto de operações realizadas tanto para corrigir problemas, notadamente de segurança e de desempenho, como para instalar versões atualizadas de bibliotecas ou software básico. Os principais parâmetros (de comandos como *apt-get* ou *aptitude*) envolvidos são *update*, *upgrade* e *dist-upgrade* ou *full-upgrade.* O *update* atualiza a lista de repositórios (onde estão os pacotes), o *upgrade* atualiza para a versão mais recente encontrada na lista para a sua distribuição, e o *dist-upgrade* atualiza para a próxima versão da distribuição. Por exemplo: instruções do manual do Debian sobre atualizações de segurança[72].

[72] https://www.debian.org/doc/manuals/securing-debian-howto/ch4.pt-br.html#s-security-update.

Por um lado, a ausência de atualizações periódicas pode deixar o ambiente vulnerável a erros ou aumentar o risco de invasão por meio de programas maliciosos, que exploram falhas existentes em versões antigas. Por outro lado, a instalação de novas versões ou a aplicação de pacotes de atualização pode afetar o funcionamento de programas restritos às versões antigas.

O caminho mais seguro é manter uma política de avaliação e testes das novas versões e dos pacotes de atualização de maneira sistemática, e não totalmente automática para tudo. Mas essa tarefa consome muito tempo e recursos da equipe de manutenção, e é muitas vezes deixada de lado. Se for possível, o ideal é manter um ambiente de homologação completo o suficiente para que, o mais brevemente possível, as novas versões e pacotes de atualização sejam testados, e, em caso de sucesso, aplicados em produção.

Recomendamos a utilização de versões LTS (*Long Term Support*), que são estáveis e têm garantia de atualização por um período mais longo.

Se não há equipe para realizar também as atualizações controladas, as atualizações podem ser realizadas de forma automática. Utilizamos a ferramenta "*Unattended Upgrades*"[73] como forma de manter as ferramentas base atualizadas nos repositórios padrão. Basta instalar essa ferramenta que automaticamente será verificada diariamente a existência de novas atualizações.

Atualização de ferramentas computacionais - performance e compatibilidade

O controle da versão utilizada em qualquer software é sempre importante, mas especialmente crucial em Bioinformática. Um exemplo clássico em que existe uma "quebra" entre duas versões do mesmo software é o das versões NCBI BLAST e NCBI BLAST+ (ALTSCHUL et al., 1990; CAMACHO et al., 2009). Esse caso é emblemático pela relevância do programa e porque houve mudanças desde os programas e seus parâmetros até o resultado gerado, com grande

[73] https://wiki.debian.org/UnattendedUpgrades

aumento na velocidade das análises. Entender a diferença entre as versões é necessário, pois vários pipelines (principalmente mais antigos) utilizam o BLAST como motor principal na busca de similaridade de sequências. Isso acontece frequentemente em programas para análise de Bioinformática, como é o caso do programa HMMER (comparação de sequências por perfis de cadeias de Markov), onde as versões 2 e 3 não são intercompatíveis (EDDY, 2011). Manter catalogadas as versões utilizadas nas análises, scripts e pipelines é mandatório para mantê-los exequíveis e repetíveis ao longo do tempo.

Esse controle de versão não fica só no software em si, mas em compiladores (g++ por exemplo) e linguagens de programação utilizadas nos scripts (Perl, Python, R, etc.). Um exemplo relevante em Bioinformática é a mudança entre as versões Python 2 e Python 3. Geralmente scripts escritos em Python 3 não funcionam corretamente na versão 2 e vice-versa. Como existe uma vasta gama de scripts já escritos em Python 2, há certa resistência nessa migração, pois seria necessária a reescrita de vários desses programas, o que raramente acontece na prática. O usuário deve ficar atento a essas diferenças sempre que for utilizar ou escrever um novo programa em Python, priorizando a utilização da versão mais atual.

Organização de informação e espaço em disco - definição de políticas

A organização adequada da informação é essencial por diversas razões, a começar pelos princípios do método científico, já que é necessário conhecer detalhes da rastreabilidade de cada informação, para precisão dos resultados a serem discutidos e para permitir a reprodutibilidade das análises, sem inconsistência. Mas não basta saber a origem de cada dado e os parâmetros de cada análise, é preciso que a informação esteja segura, em termos de acesso somente por quem tem autorização, e de redundância/backup, e isso tem relação com a estrutura dos dados. A participação dos clientes (donos da informação) é importante na definição dessa organização, mas algumas dicas podem ser aplicáveis à maioria dos casos e podem ser um ponto de partida para a conversa com os clientes.

A respeito do controle de acesso, uma estratégia normalmente utilizada na organização de dados é a compartimentalização da informação em diretórios por projeto e, quando necessário, por subdiretórios para os diferentes grupos de acesso. Para facilitar também o backup, os dados podem ser separados por classe: "dados brutos" separados de "análises". Links simbólicos ("*ln -s"*, "ln --help") facilitam o trabalho com dados separados em diferentes locais, e as permissões podem ser editadas com os comandos Linux *chown* e *chmod* ("chown --help", "chmod --help"). Observação: uma revisão rápida de vários comandos básicos pode ser vista no blog de Elias Praciano[74], que contém muitos outros tutoriais úteis. Você também pode consultar a documentação específica da distribuição que estiver utilizando.

Para o backup, a questão é facilitar e otimizar a execução do mesmo. Isso se dá levando em consideração também a política de backup adotada na organização dos dados, evitando que dados redundantes ou sem importância (como temporários) sejam incluídos no backup.

Nesse quesito, além da organização dos diretórios, restringindo ao backup o que precisa de fato de redundância, a limpeza de dados que já cumpriram sua função (resultados intermediários de análises, logs temporários, etc.) e a compactação do que deve ser mantido ('.gz', '.bz', '.dsrc', etc.) é muito importante também para a boa utilização do espaço em disco existente. A diferença de espaço utilizado em disco ao se compactar o arquivo é de cerca de 10 vezes menos. A política de organização da informação auxilia no acompanhamento e no planejamento mais correto da ampliação do parque computacional.

Outra possibilidade a ser considerada é a organização de informação em bancos de dados (BDs), e essa solução é viável quando existe informação estruturada e com demanda por acesso rápido e/ou facilitado, como para os resultados de montagem e anotação estrutural/funcional de um genoma. A partir desse ponto, faz-se backup dos BDs, e os dados em arquivos que estão nos BDs podem ser apagados ou compactados, dependendo do caso.

---

[74] https://elias.praciano.com/2013/12/linux-comandos-basicos.

Backup - segurança e recuperação

A definição da política de backup deve incluir o levantamento das necessidades dos envolvidos na recuperação de dados, tanto de partições ou discos inteiros, quanto de arquivos específicos, e considerar a estrutura presente na instituição, como limites da rede (como de velocidade e horários de menor tráfego), locais de armazenamento (*storage*, fitas, nuvem), I/O (tipos de discos/fitas), etc. O período de armazenamento de cada nível de informação deve ser decidido junto ao cliente e de acordo com a política da empresa.

Tem relação direta com a organização da informação, tratada em tópico anterior, já que é possível que diferentes níveis de informação recebam tratamento diferenciado no backup, evitando ou reduzindo sobrecarga do sistema (demora, uso excessivo de rede e de espaço em disco/fita), aumentando segurança, velocidade de backup e de restauração, e resultando utilização mais adequada dos recursos. Por exemplo, análises que não sejam muito demoradas e que geram muitos dados intermediários, ou análises não moduladas (impossíveis de recomeçar depois de interrompidas), são bons candidatos a ficar fora do backup antes do término e da limpeza dos dados intermediários desnecessários. Pela mesma lógica, análises demoradas e modulares são boas candidatas a backup frequente.

De modo geral, é recomendável que o procedimento de backup seja realizado diariamente na forma incremental (em que apenas os dados novos ou modificados recentemente são adicionados), com base em um primeiro backup completo realizada no início do processo.

Recomenda-se o uso de programa próprio e especializado de backup que garanta a integridade das informações guardadas por meio de algoritmos de verificação como MD5. O Bacula contempla esses mecanismos de verificação.

Idealmente, uma cópia do backup (fitas transportadas fisicamente, ou réplica de *storage*, por exemplo) deve ser armazenada em prédio

distinto do prédio onde se encontra seu data center, evitando que desastres como incêndios eliminem todas as cópias.

Periódica e sistematicamente, é necessário verificar se o processo de recuperação está funcionando corretamente, com a recuperação de alguns dados por amostragem. Normalmente, as auditorias solicitam esse procedimento. É importante que tanto as políticas de organização da informação e de backup quanto o processo de backup em si sejam adequadamente documentados e monitorados, de modo a permitir que qualquer empregado do setor possa tanto acompanhar o bom funcionamento quanto recuperar o sistema ou arquivos específicos em caso de falha, mesmo que o responsável pelo backup esteja ausente por qualquer motivo. Utilizamos a interface gráfica bat (*Bacula Administration Tool*). Screenshots podem ser verificados na página Bacula DocuWiki[75]. Para habilitar o bat, siga as instruções do Guia do Usuário Bacula[76]. Existem diversas outras interfaces gráficas para facilitar a utilização do Bacula, como pode-se verificar no mesmo guia e também na página Bacula DocuWiki[77].

Manutenção, reposição e ampliação de hardware — proatividade

Depois de muito trabalho, a infraestrutura foi adequadamente dimensionada, adquirida, instalada e configurada. Clientes satisfeitos, backups funcionais, monitoramento ajustado, e as auditorias indicam que está tudo bem. Finalmente poderemos esquecer o assunto? Ainda não.

A **manutenção** é facilitada pelo monitoramento bem ajustado e com envio de avisos para todas as questões mais importantes, mas não se pode esquecer de verificar se os avisos estão funcionando corretamente. Visitas periódicas ao data center e verificações completas são recomendáveis. É preciso também definir que peças e em que quantidade devem estar em estoque próprio.

---

[75] http://wiki.bacula.org/doku.php?id=bat.
[76] http://www.bacula.org/2.4.x-manuals/en/main/Installing_Bacula.html#enablebat.
[77] http://www.bacula.org/2.4.x-manuals/en/main/GUI_Programs.html e http://wiki.bacula.org/doku.php?id=3rd_party_addons).

A **reposição** de peças não é simples. As altas performance e confiabilidade cobram seu preço em termos de especificidade/compatibilidade de peças homologadas (para evitar perda de garantia), muitas vezes importadas (o que envolve tempo e fontes específicas de fundos), e além de as mesmas serem mais caras que peças comuns, modelos mais antigos se tornam muito caros e normalmente impraticáveis.

Por exemplo, seu "velho" servidor tem discos *hotplug* de 300GB, em RAID (que exige discos de mesma capacidade), e, como alguns estragaram, você precisará de novos discos para recompor o RAID. Ao cotar discos de diferentes capacidades, você descobriu que um disco de 2TB é quase do mesmo preço que o de 300GB, e que, além do menor custo/benefício do disco menor, o mesmo demorará mais para ser entregue, pois não existe em estoque. A primeira observação é que é necessário ter alguns discos reserva de cada tamanho utilizado para troca, e a segunda é que, de tempos em tempos, é preciso substituir discos antigos e de difícil manutenção por discos mais novos e maiores, para atender ao aumento da demanda e para manter a manutenção viável, já que preços são influenciados pela relação entre demanda e oferta. Lógica similar de preço, estoque e compatibilidade é aplicada a outros tipos de peças, o que indica que é necessária atenção à vida útil de cada equipamento do parque, em termos de viabilidade da manutenção e em termos de performance demandada.

A **ampliação** dos recursos do parque, por sua vez, tem a ver com a combinação de diversos fatores, tais quais: vida útil dos equipamentos; previsão e efetivação de demandas; tempo de garantia; quebras de equipamentos; e disponibilidade de recursos financeiros (época de editais, projetos aprovados ou período de construção do orçamento do ano seguinte, nas privadas).

Por meio do acompanhamento de todas essas variáveis, é possível estabelecer e manter atualizado um cronograma de reposição e ampliação do parque instalado. Para instituições de pesquisa públicas, que normalmente não têm tanta previsibilidade no aporte de recursos de infraestrutura, esse cronograma se torna ainda mais importante,

pois o crescimento das demandas é difícil de prever e as janelas de oportunidade para compras são raras e curtas.

Considerações finais

- Manter uma infraestrutura de BioTI confiável e segura não é tarefa fácil, mas, se existe demanda, é uma tarefa possível e vale o esforço.
- É recomendável ter na sua equipe um profissional de TI qualificado e continuamente atualizado em infraestrutura de hardware e software para instalação, manutenção e atualização da infraestrutura. Esse profissional potencializará diretamente a produtividade da equipe de Bioinformática, e indiretamente as pesquisas relacionadas.
- Uma boa alternativa para manter os serviços sempre funcionais é manter cópias de VMs importantes, sincronizadas com a maior frequência possível, em máquinas reais diferentes. Idealmente, configuradas para assumir a função das VMs de produção automaticamente em caso de falha.
- A escolha das linguagens de programação não deve ser restrita à expertise da equipe atual. Facilidade de entendimento do código por outros programadores, velocidade de desenvolvimento, adequação às demandas, suporte, comunidade, dentre outros fatores, devem ser considerados.
- É possível montar a infraestrutura gastando menos do que com as soluções comerciais completas (soluções *enterprise*, normalmente possíveis apenas em nível corporativo). O hardware não precisa ser o topo de linha, que normalmente é desproporcionalmente caro, mas também não pode ser de modelos muito velhos, pois estes podem complicar muito reposição e ampliação.
- Existem softwares *open source* robustos e bem suportados para tudo que precisará em BioTI, desde os Sistemas Operacionais até o monitoramento fino da infraestrutura, incluindo SGDBs, virtualização, backup, linguagens de programação e seus *frameworks*.
- A existência de uma central de alto desempenho disponível e adequada, e as demandas atual e futura devem ser cuidadosamente consideradas na avaliação da necessidade e no dimensionamento do seu data center local.

- O monitoramento da infraestrutura de hardware e software é muito importante e, como o backup, deve ser feito periodicamente.
- A segurança da informação é um assunto crucial e deve receber a devida atenção, com participação dos clientes pelo menos na organização da informação para controle de acesso e backup.
- Priorizamos referências em língua portuguesa neste tipo de documento, mas cabe destacar que a maioria das informações mais recentes e mais completas está disponível em inglês, língua essencial para pesquisa e TI.
- Este documento retrata a nossa experiência atual e está limitado a ela. Foi desenvolvido para ser um guia inicial no tema e não como única fonte de referência no assunto, que, por si só, é vasto e constantemente tem novidades - vide a substituição em andamento do Apache, que "reinava absoluto", pelo NGINX.

Referências

ALTSCHUL, S. F.; GISH, W.; MILLER, W.; MYERS, E. W.; LIPMAN, D. J. Basic local alignment search tool. **Journal of Molecular Biology**, London, v. 215, n. 3, p. 403-410, 1990. Disponível em: <https://doi.org/10.1016/S0022-2836(05)80360-2>. Acesso em: 04 dez. 2017.

BORGES, A. M.; VALENTIM, T. S.; LIMAS, J. M.; ANTUNES, A. C. Estudo comparativo entre Nagios e Zabbix In: FREITAS JUNIOR, V.; COSTA, G. C. (Org.). **Tecnologia e redes de computadores**. Sombrio, SC: Instituto Federal Catarinense, 2015. p. 65-92. Disponível em: <http://redes.sombrio.ifc.edu.br/wp-content/uploads/sites/7/2015/12/Livro-Tecnologia-e-Redes-de-Computadores-2015.pdf#page=65>. Acesso em 04 dez. 2017.

CACIATO, L. E. **Virtualização e Consolidação dos Servidores do Datacenter.** 2012. Centro de Computação, UNICAMP. Campinas, São Paulo. Disponível em: <http://www.ccuec.unicamp.br/biti/download/Artigo_Virtualizacao_Datacenter.pdf>. Acesso em: 04 Dez. 2017.

CAMACHO C.; COULOURIS G.; AVAGYAN V.; MA N.; PAPADOPOULOS J.; BEALER K.; MADDEN T. L. BLAST+: architecture and applications. **BMC Bioinformatics**, London, v. 10, artigo 421, 2009. Disponível em: <https://doi.org/10.1186/1471-2105-10-421>. Acesso em: 04 dez. 2017.

COCK, P. J. A.; ANTAO, T.; CHANG, J. T.; CHAPMAN, B. A.; COX, C. J.; DALKE, A.; FRIEDBERG, I.; HAMELRYCK, T.; KAUFF, F.; WILCZYNSKI, B.; DE HOON, M. J. L. Biopython: freely available Python tools for computational molecular biology and bioinformatics. **Bioinformatics**, Oxford, v. 25, n. 11, p. 1422-1423, 2009. Disponível em: <https://doi.org/10.1093/bioinformatics/btp163>. Acesso em: 04 dez. 2017.

DRIEMEYER, R. **Projeto de melhoria de Data Center com ênfase em infraestrutura e eficiência energética**. 2016. 131 f. TCC (Graduação em Engenharia da Computação) - UNIVATES, Lajeado, RS. Orientador: Edson Moacir Ahlert. Disponível em: <https://www.univates.br/bdu/bitstream/10737/1382/1/2016RodolfoDriemeyer.pdf>. Acesso em: 04 dez. 2017.

EDDY, S. R. Accelerated profile HMM searches. **PLoS Computational Biology**, San Francisco, v. 7, n. 10, 2011. Disponível em: <https://doi.org/10.1371/journal.pcbi.1002195>. Acesso em: 04 dez. 2017.

FARIA, F. A. **Tecnologia de discos rígidos**: IDE, SATA, SCSI e SAS. Disponível em: <http://www.ic.unicamp.br/~ducatte/mo401/1s2008/T2/079734-t2.pdf>. Acesso em: 04 dez. 2017.

FERREIRA FILHO, J. A.; BRITO, L. S. ; LEÃO, A. P.; ALVES, A. A.; FORMIGHIERI, E. F.; SOUZA JUNIOR, M. T. In silico approach for characterization and comparison of repeats in the genomes of oil and date palms. **Bioinformatics and Biology Insights**, London, v. 11, p. 1-12, abr. 2017. Disponível em: <https://doi.org/10.1177/1177932217702388>. Acesso em: 04 dez. 2017.

FRANÇA, F. A.; MEDEIROS, G. C. **Gerência e configuração de software**: análise prática de sistemas de controle de versão. 2016. 199 f. TCC (Graduação em Ciência da Computação) - Universidade do Sul de Santa Catarina. Palhoça, SC. Orientador: Saulo Popov Zambiasi. Disponível em: <https://www.riuni.unisul.br/handle/12345/2007>. Acesso em: 04 dez. 2017.

FRIGO, A. B. G. **Infraestrutura de data center e suas tendências com foco em eficiência energética.** 2015. 52 f. TCC (Graduação em Engenharia Elétrica) - UNESP, Guaratinguetá, SP. Orientador: Leonardo Mesquita. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/139220/000864819.pdf?sequence=1&isAllowed=y>. Acesso em: 04 dez. 2017.

IHAKA, R.; GENTLEMAN, R. R: A language for data analysis and graphics. **Journal of Computational and Graphical Statistics**, v. 5, n. 3, p. 299-314, 1996. Disponível em: <https://doi.org/10.1080/10618600.1996.10474713>. Acesso em: 04 dez. 2017.

JARGAS, A. M. **Introdução ao Shell Script.** Disponível em: <http://aurelio.net/shell/apostila-introducao-shell.pdf>. Acesso em: 04 dez. 2017.

LEPREVOST, F. V.; BARBOSA, V. C.; FRANCISCO, E. L.; PEREZ-RIVEROL, Y.; CARVALHO, P. C. On best practices in the development of bioinformatics software. **Frontiers in Genetics**, v. 5, artigo 199, 2014. Disponível em: <https://doi.org/10.3389/fgene.2014.00199>. Acesso em: 04 dez. 2017.

MAROCCO, C. A. D. **Proposta de topologia de rede de dados com segurança e foco na produtividade, utilizando ferramentas de software livre.** 2015. 65 f. TCC (Pós-graduação Lato Sensu em Rede de Computadores com Ênfase em Segurança) - UniCEUB, Brasília, DF. Orientador: Marco Antônio de Oliveira Araújo. Disponível em: <http://repositorio.uniceub.br/bitstream/235/8157/1/51307936.pdf>. Acesso em: 04 dez. 2017.

MAZIERO, C. A. **Sistemas operacionais:** conceitos e mecanismos. DINF/UFPR. Curitiba, PR: DINF/UFPR, 2017. 372 p. Disponível em: <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf>. Acesso em: 04 dez. 2017.

ROSA, M. G.; BORGES, R. L.; SOUZA, M. A. S.; MALLMANN, J. Comparativo entre Softwares de Backup em Ambiente Organizacional. In: FREITAS JUNIOR, V; SILVA, T. N.; NUNES, L. L. S. T.; LUZ, G. L. **Tecnologia e redes de computadores.** Sombrio, SC: Instituto Federal Catarinense, 2015. p. 93-122. Disponível em: <http://redes.sombrio.ifc.edu.br/wp-content/uploads/sites/7/2015/12/Livro-Tecnologia-e-Redes-de-Computadores-2015.pdf#page=93>. Acesso em: 04 dez. 2017.

SANTOS, R. H. S. **Modelo de Gestão de Predição de Falhas no Gerenciamento da Infraestrutura de *Datacenter*.** 2014. 118 f. Tese (Doutorado em Engenharia Elétrica) - UNESP, Ilha Solteira, SP. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/111128/000797388.pdf?sequence=1&isAllowed=y>. Acesso em: 04 dez. 2017.

SILVA, E. T. **Software livre no monitoramento de serviços e backup de dados por meio de redes de computadores.** 2015a. 63 f. TCC (Especialização em Redes de Computadores) - Universidade Tecnológica Federal do Paraná, Pato Branco, PR, Orientador: Christian Carlos Souza Mendes. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/5903/1/PB_ESPRC_II_2015_04.pdf>. Acesso em: 04 dez. 2017.

SILVA, N. R. **TI verde – o armazenamento de dados e a eficiência energética no data center de um banco brasileiro.** 2015b. 141 f. Dissertação (Mestrado) - Universidade Nove de Julho, São Paulo, SP. Orientador: Flavio Hourneaux Junior. Disponível em: <http://bibliotecatede.uninove.br/handle/tede/1155>. Acesso em: 04 dez. 2017.

SILVA, W. M.; MEDEIROS, R. M.; MARTINS, R. S. Análise e gerenciamento de redes usando uma metodologia proativa com Zabbix. **Holos**, Natal, n. 31, v. 8, p. 277-289, 2015. Disponível em: <https://doi.org/10.15628/holos.2015.2441>. Acesso em: 04 dez. 2017.

SKINNER, M. E.; UZILOV, A. V.; STEIN, L. D.; MUNGALL, C. J.; HOLMES, I. H. JBrowse: a next-generation genome browser. **Genome Research**, Cold Spring Harbor, v. 19, n. 9, p. 1630-1638, 2009. Disponível em: <https://doi.org/10.1101/gr.094607.109>. Acesso em: 04 dez. 2017.

SOARES, L. D.; COSTA, R. M. R. Gerência de redes: utilizando o Zabbix para monitorar a disponibilidade e transferência de imagens. **Caderno de Estudos em Sistemas de Informação**, Juiz de Fora, v. 1, n. 1, 2014. Disponível em: <https://seer.cesjf.br/index.php/cesi/article/view/122/42>. Acesso em: 04 dez. 2017.

STAJICH, J. E.; BLOCK, D.; BOULEZ, K.; BRENNER, S. E.; CHERVITZ, S. A.; DAGDIGIAN, C.; FUELLEN, G; GILBERT, J. G. R.; KORF, I.; LAPP, H,; LEHVÄSLAIHO, H.; MATSALLA, C.; MUNGALL, C. J.; OSBORNE, B. I.; POCOCK, M. R.; SCHATTNER, P.; SENGER, M.; STEIN, L. D.; STUPKA, E.; WILKINSON, M. D.; BIRNEY, E. The Bioperl Toolkit: perl modules for the life sciences. **Genome Research**, Plainview, v. 12, n. 10, p. 1611–1618, 2002. Disponível em: <https://doi.org/10.1101/gr.361602>. Acesso em: 04 dez. 2017.

TEH, H. F.; NEOH, B. K.; ITHNIN, N.; DAIM, L. D. J.; OOI, T. E. K.; APPLETON, D. R. Review: omics and strategic yield improvement in oil crops. **Journal of the American Oil Chemists Society**, New York, v. 94, n. 10, p. 1225-1244, 2017. Disponível em: <https://doi.org/10.1007/s11746-017-3033-8>. Acesso em: 04 dez. 2017.

TELESCA, A.; CARENA, F.; CARENA, W.; CHAPELAND, S.; BARROSO V. C.; COSTA, F.; DÉNES F.; DIVIÀ, F.; FUCHS, F.; GRIGORE, A.; IONITA, C.; DELORT, C.; SIMONETTI, G.; SOÓS, C.; VYVRE, P. V.; HALLER, B. System performance monitoring of the ALICE Data Acquisition System with Zabbix**. Journal of physics: Conference Series,** Bristol, v. 513, artigo 062046, 2014. Disponível em: <https://doi.org/10.1088/1742-6596/513/6/062046>. Acesso em: 04 dez. 2017.

WORLD WIDE WEB TECHNOLOGY SURVEYS. **Historical yearly trends in the usage of web servers for websites**: this report shows the historical trends in the usage of web servers since January 2010. Disponível em: <https://w3techs.com/technologies/history_overview/web_server/ms/y>. Acesso em: 04 dez. 2017.

ZAYNAB, M.; KANWAL, S.; ABBAS, S.; FIDA, F.; ISLAM, W.; QASIM, M.; REHMAN, N.; FURQAN, M.; RIZWAN, M.; ANWAR, M.; HUSSAIN, A.; TAYAB, M. Bioinformatics tools in agriculture: an update. **PSM Biological Research**, Punjab, v. 2, n. 3, p. 111-116, 2017. Disponível em: <https://journals.psmpublishers.org/index.php/biolres/article/view/70/42>. Acesso em: 04 dez. 2017.

Embrapa
Agroenergia

MINISTÉRIO DA
AGRICULTURA, PECUÁRIA
E ABASTECIMENTO

BRASIL
GOVERNO FEDERAL

CGPE 14188