

# Volunteer Computing

Anderson Andrei da Silva 8944025

Patrick Abrahão Menani 8941050

Vinícius Pessoa Duarte 8941043

*MAC 5742-0219 Introdução à Programação Concorrente, Paralela e Distribuída*

## Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Objetivo</b>	<b>2</b>
<b>3</b>	<b>História</b>	<b>3</b>
<b>4</b>	<b>Implementação</b>	<b>5</b>
4.1	Como funciona . . . . .	5
4.2	Servidores . . . . .	5
4.3	Clientes . . . . .	5
4.4	Tratamento de erros . . . . .	5
4.5	Um exemplo simples . . . . .	6
<b>5</b>	<b>Benefícios</b>	<b>7</b>
<b>6</b>	<b>Desvantagens</b>	<b>7</b>
<b>7</b>	<b>Exemplos</b>	<b>8</b>
7.1	Exemplos - Middleware . . . . .	8
7.2	Exemplos - Projetos . . . . .	8
<b>8</b>	<b>Nuvens Públicas - Uma alternativa</b>	<b>9</b>
<b>9</b>	<b>Conclusão</b>	<b>9</b>
<b>10</b>	<b>Referências</b>	<b>10</b>

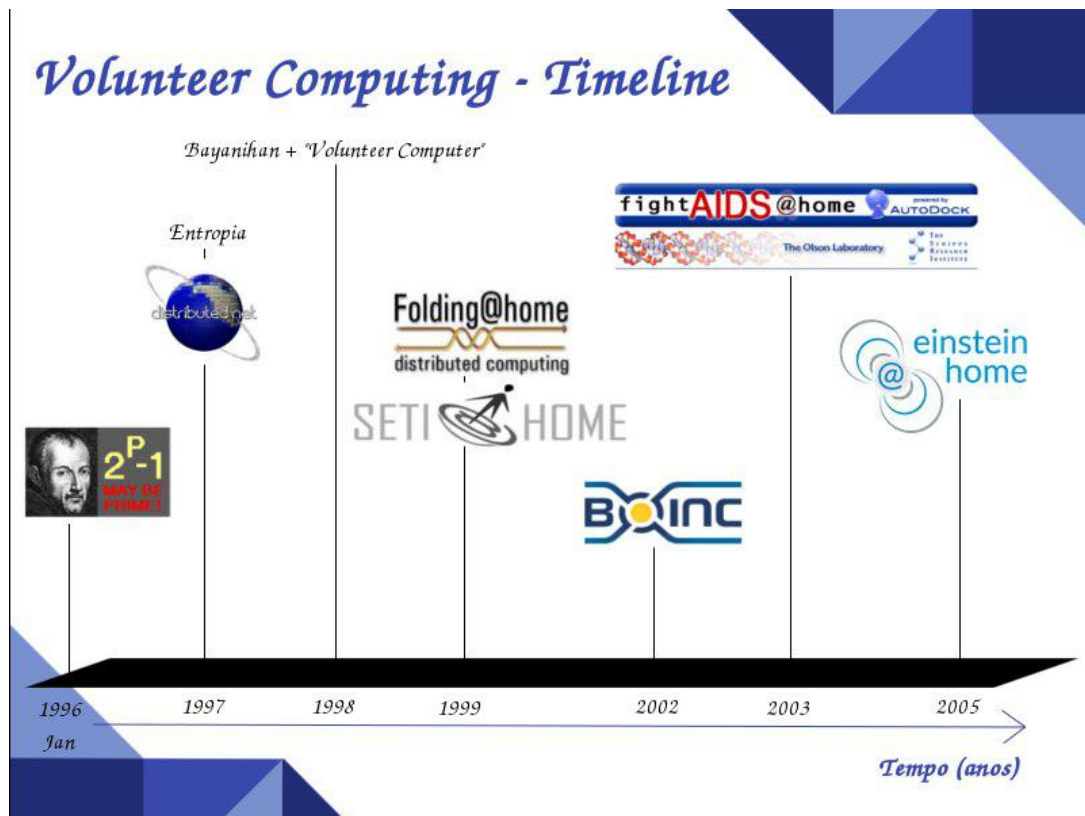
## **1. Introdução**

Projetos envolvendo HPC são em sua grande maioria bastante custosos tanto no âmbito financeiro quanto no âmbito computacional. No meio universitário principalmente, isso tem dificultado a criação e manutenção de projetos do tipo. A partir de tais necessidade pensou-se em compartilhamento de recursos. Aqueles que se dispõem a fazer são chamados nesse contexto de voluntários, adeptos desses projetos e sem vínculo direto com eles na maioria das vezes. São gerenciados por meio de softwares intermediadores conectados via internet e assim doam seus recursos ociosos, processamento e armazenamento, contribuindo para a sustentação de tais projetos. Este conceito então passou a ser chamado, no decorrer do anos, de Volunteer Computing.

## **2. Objetivo**

De forma que fique menos custoso o desenvolvimento de projetos maiores, o objetivo principal de Volunteer Computing é distribuir tarefas adequadas para cada voluntário, utilizando da forma mais consciente possível os recursos que o mesmo dispõe, sem interferir no desempenho de sua máquina e no seu uso pessoal. Dessa forma qualquer usuário pode ajudar projetos do tipo, e o mesmo pode alcançar enorme poder computacional, maior até do que se tivesse estruturas específicas para tal. Para fazer tal distribuição são utilizados middlewares. Software que distribuem tarefas para as máquinas voluntárias e que, quando são finalizadas, checam e os submetem ao servidor principal.

### 3. História



A figura ilustra uma linha do tempo simplificada pelo grupo, pontuados os tópicos principais abordados nesse trabalho. Descrevendo-a em detalhes e em ordem cronológica teremos:

- **1996 - GIMPS**

Criado por George Woltman, assim como os programas Prime95 e MPrime, o projeto busca descobrir novos números de Mersenne. A ideia, já presente nos softwares citados, agora é inserida no âmbito de Volunteer Computing. Ao descobrir um novo número de Mersenne, os créditos pela descoberta são dados ao dono da máquina que realizou a tarefa. O mais novo número descoberto, em janeiro de 2016, é o  $2^{74.207.281}$ , descoberto por Curtis Cooper.

- **1997 - Entropia**

A Entropia,Inc. foi fundada em 1997 e desenvolveu um dos primeiros middlewares utilizados, inclusive pelo GIMPS, distributed.net e Folding@home (que serão abordados a seguir).

- **1997 - distributed.net**

O projeto de autoria da Distributed Computing Technologies,Inc. foi um dos primeiros a aplicar o conceito de Volunteer Computing à GPUs.

- **1998 - Bayanihan**

Projeto do MIT Lab for Computer Science, este também é um middleware, feito principalmente para projetos em Java, e aqui tivemos a criação do termo "Volunteer Computing" por Luis F. G. Sarmenta

- **1999 - *Folding@HOME***

Sob atual responsabilidade do Pand Lab da Universidade de Stanford, este é um projeto distribuído com o objetivo de simular e estudar dobra de proteínas.

- **1999 - *SETI@HOME***

Criado na Universidade da Califórnia, Berkeley, o projeto SETI@HOME busca por vida extraterrestre analisando sinais de rádio. E aqui é interessante pontuar também, que as descobertas em máquinas voluntárias são registradas nos nomes de seus donos.

- **2002 - *BOINC***

Também criado na Universidade da Califórnia, Berkeley, este é o middleware mais utilizado recentemente, tendo substituído o Entropia em vários projetos como o GIMPS, distributed.net, Folding@home, e sendo utilizados por todos os outros citados neste trabalho.

- **2003 - *FightAIDS@HOME***

Desenvolvido no Laboratório de Olson no Instituto de Pesquisa Scripps, que também pode ser executado em sistemas Android, busca através de simulações a cura para o vírus HIV.

- **2005 - *Einsteins@HOME***

Foi criado em parceria entre os grupos americanos LSC (Instituto de Tecnologia da Califórnia e de Massachusetts) e grupo não governamental alemão MPG. Esse projeto procura por fortes ondas gravitacionais emitidas por buracos negros, estrelas de nêutrons, estrelas de quarks e entre outros objetos que possam emití-las.

É interessante pontuar que existem vários outros projetos no intervalo de tempo ilustrado, porém não caberia aqui pontuar todos. Mas a partir de 2005 não vimos muita coisa nova ser feita, apenas, em sua grande maioria, a continuação de projetos já criados. Então apesar da ideia de Volunteer Computing ser bem interessante, fica o questionamento do porque de não se ter tantos novos projetos sendo criados.

## 4. Implementação

### 4.1. Como funciona

Para implementar um sistema de computação voluntária, são necessárias a abstração do problema a ser resolvido, servidores, clientes e, opcionalmente, middleware. É necessário que o problema a ser resolvido possa ser dividido em problemas menores, chamados aqui de tarefas. Essas tarefas são então distribuídas para os doadores de modo a serem executadas. Ao fim da execução, o doador entrega os dados computados para alguém que tem a capacidade de montar a resposta para o problema original.

### 4.2. Servidores

Os servidores têm como função criar as tarefas a partir do problema original para, então, distribuí-las aos clientes, ou doadores, que irão então executá-las.

Também é prerrogativa dos servidores o recebimento dos dados computados por cada cliente, implementação de políticas de tratamento de erros e montagem da resposta para o problema original.

Ferramentas como o BOINC, framework que funciona como middleware, calculam estimativas de recursos necessários para a realização bem-sucedida de uma tarefa, como por exemplo:

- Número de cálculos de ponto flutuante;
- Espaço de armazenamento necessário;
- Conexão de internet para transmitir os dados.

O cálculo dessas estimativas permite que o servidor distribua as tarefas para clientes que tenham capacidade de executá-las. Com isso, é esperado que mais tarefas sejam completadas, o que aumenta a performance do sistema, além de diminuir o desperdício dos recursos computacionais oferecidos pelo cliente.

### 4.3. Clientes

Os clientes são basicamente as máquinas que doam seus recursos para a computação de tarefas relacionadas a algum projeto. Esses clientes recebem dos servidores algum conjunto de tarefas, computando-as então. Após a computação ser realizada, retorna os dados obtidos para os servidores e então pode receber novas tarefas a serem executadas. Em geral, têm instalado um software que realiza o intermédio com o cliente, além de definir os momentos em que uma tarefa pode ser executada.

Normalmente, uma tarefa é executada quando a máquina está em estado ocioso, i.e., com pouco uso de seus recursos, por exemplo quando a proteção de tela está ativada. Além disso, a tarefa tem uma prioridade de execução baixa no sistema operacional. Com isso, tenta-se evitar que o usuário perceba perda de performance em sua máquina devido à execução de uma tarefa, de modo que o usuário não interrompa as tarefas.

### 4.4. Tratamento de erros

Durante o processo de computação de uma tarefa, diversos problemas podem ocorrer, por exemplo:

- Clientes podem não retornar os dados computados;

- Clientes com mal-funcionamento podem retornar respostas erradas;
- Clientes tentam fraudar as respostas por quaisquer motivos.

Para garantir a integridade dos dados recebidos, é comum que os sistemas de computação voluntária utilizem o paradigma de computação replicada.

Com esse paradigma, os servidores distribuem as tarefas para mais de um cliente, que executaram essas mesmas tarefas e rotornaram os dados para o servidor.

Ao receber esses dados replicados, o servidor utiliza alguma métrica relacionada ao problema em questão para verificar a semelhança entre as respostas obtidas pelos clientes. Caso os dados tenham semelhança dentro de algum intervalo definido, a resposta é aceita, caso contrário, a tarefa deverá ser executada novamente.

#### 4.5. *Um exemplo simples*

Nessa seção é apresentado um exemplo simples de computação voluntária, sem a abstração de máquina.

Imagine que o professor Alfredo decidiu organizar, com os professores DCC, um almoço para os alunos de escola pública que participam de um projeto de computação no IME.

O professor Alfredo define que 3 coisas são necessárias:

- Comida;
- Sobremesa;
- Entretenimento.

Com as tarefas definidas é preciso pedir que os professores ajudem com as tarefas. Os professores Gubi, Leliane, Carlinhos e Daniel se voluntariam para ajudar, então o Alfredo distribui, de acordo com as habilidades dos professores, as tarefas definidas anteriormente:

- Leliane e Daniel - Ficam responsáveis pela comida;
- Carlinhos - Será o responsável pela sobremesa;
- Gubi - Cuidará do entretenimento (Mágica).

Na medida em que um professor completa sua tarefa, leva a comida ou apresentação para o saguão do bloco B, onde o almoço será montado. O Alfredo então verifica se cada professor levou o combinado, senão, pede para que o professor vá fazer a tarefa correta.

Tendo todas as tarefas corretamente executadas, o professor Alfredo pode então montar a mesa e chamar as crianças.

## 5. Benefícios

A técnica de volunteer computing pode oferecer alguns benefícios tanto para a comunidade científica como para a sociedade de uma forma geral alguns destes benefícios são:

- Conscientizar a população sobre projetos de pesquisa devido a divulgação da possibilidade de uma pessoa engajada, porém leiga em certo assunto, poder colaborar em um projeto doando sua máquina para uso;
- Devido a possibilidade de ajudar sem precisar de conhecimentos específicos em uma área, os projetos de volunteer computer tem uma possibilidade de aumentar o engajamento do público em relação a colaboração em pesquisas científicas;
- Como o processamento doado é voluntário, não se espera nada em troca dele, portanto, esta técnica pode permitir métodos de pesquisa mais baratos para processos que anteriormente precisariam de um investimento grande por parte dos pesquisadores, ou da instituição que os financia;
- O processamento é dividido e executado de forma distribuída entre vários voluntários possibilitando assim gerar resultados mais rapidamente;
- Os computadores ociosos em diversas situações possuem um processamento útil que está parado, sem nenhuma utilidade, portanto, a volunteer computing se propõe a utilizar os recursos computacionais de forma mais eficiente, dando uma utilidade para os computadores ociosos.

## 6. Desvantagens

Como toda técnica, a de volunteer computing possui algumas desvantagens, como por exemplo:

- Ela é uma tecnologia ainda em desenvolvimento, podendo apresentar alguns problemas em relação aos sistemas utilizados, tanto para o lado do cliente quanto para o lado dos servidores;
- Como ela utiliza o processamento de um computador voluntário ela acaba ocasionando um aumento do consumo de energia do doador, pelo fato de utilizar o computador enquanto ele estaria ocioso. Um período em que a máquina deveria gastar menos energia acaba se tornando em um período comum de utilização da máquina;
- Durante o período que o computador do voluntário é utilizado pode ocorrer um decréscimo na performance da máquina do doador (voluntário).

## 7. Exemplos

### 7.1. Exemplos - Middleware

Esta seção aborda alguns exemplos dos Middlewares disponíveis hoje em dia.

- Berkeley Open Infrastructure for Network Computing (BOINC) é o sistema de middleware mais utilizado. Ele oferece um sistema de software para os clientes em diversas plataformas, como Windows, Mac OS X, Linux, Android, e outras variantes de Unix;
- XtremWeb é usado principalmente como uma ferramenta de pesquisa, ele é desenvolvido por um grupo baseado na University of Paris-South;
- Xgrid é desenvolvido pela Apple. Ele fornece um sistema de cliente e servidor que são exclusivos do sistema Mac OS X;
- Grid MP é uma plataforma de middleware desenvolvida pela United Devices, foi utilizada em alguns projetos de volunteer computing como o grid.org, World Community Grid, Cell Computing e Hikari Grid

### 7.2. Exemplos - Projetos

Esta seção aborda alguns projetos ativos hoje em dia que fazem uso da técnica de volunteer computing.

- Folding@HOME  
Criado em Stanford com objetivo de entender como funcionam interações entre proteínas e suas relações com doenças.  
O Alzheimer, por exemplo, é causado pela agregação de algumas pequenas proteínas chamadas Abeta peptídeos (agregados).  
Tóxicos para os neurônios, causando a morte de várias células neuronais. Criar drogas para combater a doença, além de ser possível ter ideia de como esses agregados se formam, talvez sendo possível impedir a formação deles.
- einstein@HOME  
Procura encontrar estrelas de nêutrons a partir da análise de dados gerados por detectores de ondas gravitacionais, rádio-telescópios e satélites de raios-gama  
Os voluntários do projetos já encontraram cerca de 50 estrelas de nêutrons; Tem como meta fazer a primeira detecção direta de ondas gravitacionais emitidas por estrelas de nêutrons em rotação (pulsares).
- FightAIDS@HOME  
Gerenciado pelo laboratório Olson no instituto de pesquisa Scripps, localizado nos Estados Unidos.  
Modela a evolução do vírus HIV à drogas e avalia possíveis substâncias candidatas a drogas contra o HIV. Recentemente se juntou a World Community grid, a qual agora tem mais de um milhão de voluntário registrados.



## **8. Nuvens Públicas - Uma alternativa**

Nos últimos dez anos, a Computação Voluntária vem perdendo força devido ao maior uso da tecnologia de nuvens públicas. Apesar de muitas dessas nuvens serem pagas para utilização, elas possuem uma empresa por trás que garante o funcionamento, diferentemente das "nuvens" da computação voluntária que normalmente são instituições de pesquisa que recrutam voluntários e dependem do engajamento desses para que os projetos tenham uma base sólida de poder de computação.

Também existem nuvens públicas nas quais um certo poder de processamento é disponibilizado gratuitamente, o que, aliado à maior certeza de quantidade de recursos disponíveis, pode tornar essas nuvens mais interessantes aos olhos de alguns pesquisadores.

Mesmo assim, mesmo com essa nova tecnologia, projetos de computação voluntária continuam recebendo apoio de empresas grandes, como a IBM. Assim como a computação voluntária não substitui completamente o uso de clusters, a nuvem não substitui a computação voluntária, a final, a nuvem precisa da construção de alguma infraestrutura, que deve ser paga se não com dinheiro, com publicidade e dados coletados pelos donos dessas nuvens.

## **9. Conclusão**

O uso de computação voluntária pode ser de extrema utilidade quando tratamos de processo computacionais extremamente custosos que podem ser distribuídos entre diversos dispositivos.

O campo científico tem muito a ganhar com a computação voluntária, o custo de se obter os resultados, que seriam necessários clusters extremamente caros, se torna praticamente nenhum para os pesquisadores, possibilitando uma maior evolução de diversas pesquisas.

## 10. Referências

- <http://www.volunteer-computing.org/>
- <http://setiathome.berkeley.edu/>
- <https://pt.wikipedia.org/wiki/B0INC/>
- <http://folding.stanford.edu/diseases/>
- <https://einsteinathome.org/pt-br/about>
- <http://www.wseas.us/e-library/conferences/2007corfu/papers/540-118.pdf>
- <http://www.volunteer-computing.org/EN/why-use-volunteer-computing.html>
- <http://www.volunteer-computing.org/EN/volunteer-powered-projects.html>
- [https://pdfs.semanticscholar.org/4fbd/68ba641f16d1566d3f1dc7704f2f63b938a3.pdf?\\_ga=2.980344.89964398.1497913612-119129416.1497913612](https://pdfs.semanticscholar.org/4fbd/68ba641f16d1566d3f1dc7704f2f63b938a3.pdf?_ga=2.980344.89964398.1497913612-119129416.1497913612)
- <http://www.sciencedirect.com/science/article/pii/S0167739X99000187>
- <http://groups.csail.mit.edu/cag/bayanihan/papers/eurotools98/>
- <http://groups.csail.mit.edu/cag/bayanihan/papers/fgcs/html/>
- <https://www.qinetwork.com.br/publica-privada-comunidade-ou-hibrida-conheca-os-modelos-de-cloud-computing>
- [https://en.wikipedia.org/wiki/Volunteer\\_computing](https://en.wikipedia.org/wiki/Volunteer_computing)