# BOINC

Um sistema para Computação Voluntária

**Jeferson Prevedello**
**Rafael Antonioli**

## Sumário

- Origem do BOINC
- Fórmula de Mersenne
- Computação voluntária
- Boinc x Grid
- Computação redundante
- Segurança
- Projetos

## Origem do BOINC

- Surgiu a partir do projeto GIMPS
  - *Great Internet Mersenne Prime Search*
- Objetiva buscar o maior número primo utilizando a fórmula de Mersenne
- Trabalha com computação em cluster para dividir o processamento
- GIMPS surgiu em 1996

## Fórmula de Mersenne

- $M = 2^n-1$ onde n é um número natural
- Até hoje foram encontrados 44 números primos de Mersenne
- $3+1 = 4 = 2^2$
- $7+1 = 8 = 2^3$
- 13 é primo mas não é Mersenne pois

13+1= 14 e 14 não pode ser potência de 2

## Fórmula de Mersenne

- Quatro primeiros primos de Mersenne: 3, 7, 31 e 127
- 44º primo de Mersenne possui 9.808.358 dígitos
- Descoberto em 4 set. 2006 utilizando um cluster com 700 computadores
- Utilizado para pesquisas em criptografia e teoria dos números

## Computação voluntária

- Qualquer dispositivo que possua um processador e esteja interconectado por uma rede, pode realizar computação voluntária
- Exemplos: Telefones celulares, videogames, laptops e computadores, futuramente existirão geladeiras e televisores interconectados entre si

## Computação voluntária

- Funciona bem quando não há grandes transferências de dados entre computadores
- Essa limitação tende a ser superada de acordo com o aumento das velocidades de transmissões da Internet

## Requisitos de funcionamento

- Um servidor que distribui tarefas, mantém uma lista dos participantes do projetos e armazena os resultados recebidos dos clientes
- Informações são trocadas pela porta 80 (http) para evitar problemas com *firewalls*

## Projetos científicos

- Cientistas estão tirando da gaveta projetos que antigamente jamais poderiam ser solucionados por falta de poder computacional ou altos custos de processamento
- Qualquer pessoa pode criar um projeto BOINC e utilizar computação voluntária

## Boinc

**B**erkeley
**O**pen
**I**nfrastructure for
**N**etwork
**C**omputing

- Disponível para Windows, Linux x86, Mac OS X e Solaris / Sparc

## Boinc

- Em produção desde 1999
- Finalizado em 2002
- Trabalha com CPU ociosa para processar dados científicos
- Utilizado em áreas diversas: Biologia Molecular, Climatologia e Astrofísica
- Projetos sem fins lucrativos

## Motivação

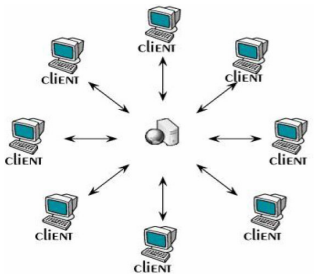
- Sistema de créditos / ranking
  - Usuário da semana
- Proteção de tela diversificada
- Importância de contribuir com a ciência
- Possibilidade de criar equipes de colaboradores

## Motivação

- Acompanhar a evolução das pesquisas
- Possibilidade de criação de um ambiente controlado para projetos de interesse privado

## Modelo cliente/servidor

Computação distribuída baseada no modelo cliente/servidor



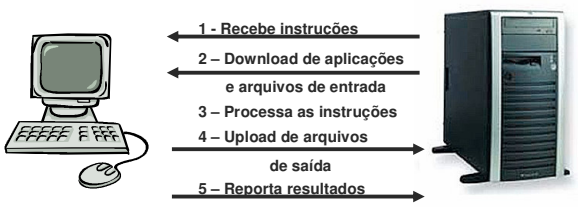

## Arquitetura

- Arquitetura mínima
  - Pentium 166 – 32 Mb RAM
- Arquitetura recomendada
  - Acima de 1 GHz – 64 Mb RAM
- Requisitos
  - Acesso periódico à Internet
- Cliente BOINC consome 16 Mb RAM

## Boinc x Grid

- Boinc requer pouca entrada/saída e alto poder de CPU
- Grid não possui limitações de E/S e CPU e envolve compartilhamento de recursos
- Boinc em muitos casos são computadores voluntários

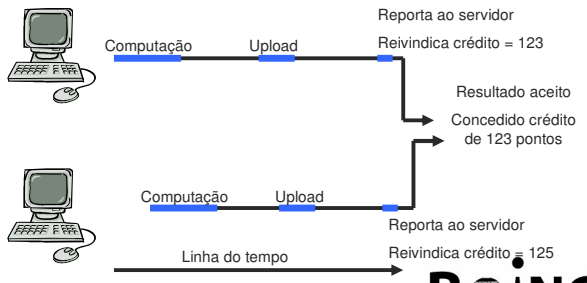
## Funcionamento do BOINC




## Estrutura básica

- Cada unidade de trabalho (Work Unit) é enviada para pelo menos 3 PCs
- Toda unidade de trabalho possui *timeout*
- Se não chegar resultado nesse tempo, é reenviado para outro cliente processar

## Computação Redundante




## Sistema de créditos

- 1º voluntário: crédito 100 pontos
- 2º voluntário: compara com o primeiro, se correto recebe 80 pontos
- 3º voluntário: pode ser descartado, mesmo assim recebe 70 pontos

## Tratamento de resultados

- Pode ser customizado:
  - O número de máquinas que processam o mesmo trabalho
  - Qual a quantidade de resultados iguais é necessária para considerar um resultado aceitável
  - Tempo de *timeout* de uma unidade de trabalho

## Segurança

- Utiliza assinatura codificada para prevenir resultados falsos
- Programas e dados codificados com *hash* de chave privada
- Chave pública para os clientes
- Possibilidade de múltiplos servidores redundantes a falhas

## Tipos de ataques

- Falsificação de resultados
  - Utilizando computação redundante
  - Verificação do resultado
  - Se a maioria dos resultados é aceita, então o processamento está correto

## Tipos de ataques

- Falsificação de créditos
  - Utilizando computação redundante
  - Verificação de créditos
  - Cada participante recebe um crédito mínimo para resultados corretos

## Tipos de ataques

- Execução maliciosa distribuída é combatida com:
  - Assinatura de código, servidor isolado gera chaves privadas
  - Clientes recebem chave pública
  - Mesmo que ataquem o servidor BOINC não é possível fazer clientes aceitar arquivos falsos
  - Chave pode ser alterada periodicamente

## Tipos de ataques

- Ataques de DOS (Denial of Service) são evitados utilizando:
  - Arquivos com tamanho máximo associado
  - Uploads com autenticação de chaves
  - Assinatura digital
  - Verificações na descrição dos arquivos antes de aceitá-los

## Tipos de ataques

- Roubo de informações de contas de clientes no servidor
  - Responsabilidade do administrador do projeto
  - Utilizando servidores com *firewall* e desativando serviços de rede não usados
  - Acessar as máquinas com protocolos do tipo SSH
  - Realizar auditorias de segurança

## Tipos de ataques

- Roubo de informações através de ataque na rede dos clientes
  - BOINC não previne que atacantes usem *sniffers* para pegar contas de usuários, acessar seu cadastro e mudar preferências
  - Acesso a outras informações não é possível com este ataque

## Tipos de ataques

- Roubos de arquivos dos projetos
  - Arquivos de entrada e saída dos projetos não são criptografados
  - Aplicações poderiam fazer isso, mas como os dados residentes na memória são *cleartext*, podem ser acessados com um depurador

## Tipos de ataques

- Abuso intencional dos participantes através dos projetos
  - Não há *sandboxing* (ambiente controlado) de aplicações
  - Participantes quando se unem a projetos BOINC confiam a segurança dos seus sistemas aos projetos

## Tipos de ataques

- Abuso acidental de participantes através de projetos
  - BOINC detecta quando uma aplicação utiliza muito espaço em disco, memória ou tempo de CPU e os aborta
  - Projetos podem minimizar esse problema através de testes prévios dos seus problemas matemáticos em plataformas diferentes

## Diversidade de projetos

- Uma única instância do Boinc pode trabalhar com diversos projetos simultâneos
- O usuário define o processamento que vai destinar para cada projeto

## Criando um projeto

- Requer uma máquina com gcc, curl, automake, autoconf, Python, Apache, MySQL e OpenSSL
- Proprietário do projeto é responsável pela segurança e manutenção dos seus servidores

## Seti@home

- Projeto que procura por sinais de extraterrestres enviados através de ondas de rádio
- Processa dados provenientes de ondas de rádio captadas no universo pela NASA, radiotelescópios varrem distâncias até 200 anos luz da Terra

## Seti@home

- WU de 320 Kbytes
- Download 3 min
- Processamento 4 horas (AMD 2800+)
- Resultado tem entre 7 e 30 Kbytes
- Prazo limite de 15 dias

## Climateprediction.net

- Calcula modificações climáticas no planeta, utilizando parâmetros alternados (atividades vulcânicas, solares, interferências humanas) e executando um modelo de milhares de anos

## Climateprediction.net

- WU de até 1 Mbyte
- Download 20 min
- Processamento 600 horas (AMD 2800+)
- Resultado tem entre 1 e 8 Mbytes
- Prazo limite de 1 ano

## Einstein@home

- Procura por pulsares e ondas gravitacionais emitidas por Pulsares, Buracos Negros, Estrelas de Nêutrons, Estrelas de Quarks e outros objetos bastante densos, que teoricamente, podem emitir fortes Ondas Gravitacionais.

## Einstein@home

- WU de até 7 a 20 Mbytes
- Download de 1 a 3 horas
- Processamento 6 horas (AMD 2800+)
- Resultado tem 330 Kbytes
- Prazo limite de 15 dias

## Rosetta@home

- Procura estruturas de proteínas, aminoácidos e suas interações para construir proteínas complexas e possibilitar cura ou tratamento de doenças humanas (Câncer, HIV, Alzheimer e Malária)

## Rosetta@home

- WU de até 18 Mbytes
- Download de 1 a 3 horas
- Processamento 3h horas (AMD 2800+)
- Prazo limite de 29 dias

## Possíveis problemas

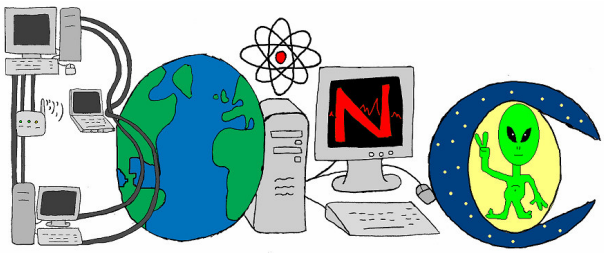
- Degradação do hardware
  - Superaquecimento
  - Travamento
  - Reinicializações
- Incompatibilidade com alguns softwares

## Referências


- Anderson, D. P., Korpela, E., and Walton, R. 2005. High-Performance Task Distribution for Volunteer Computing.
- Anderson, D. P. 2004. BOINC: A System for Public-Resource Computing and Storage.


BOINC

## Obrigado



Berkeley Open Infrastructure for Network Computing