

Introdução à Pesquisa em Informática

# Como ler textos científicos e registrar o que foi lido?

**Lesandro Ponciano**

2024

# Objetivos da Aula

- Contextualizar a leitura de texto científico
- Introduzir a leitura em três etapas
- Discutir duas formas de documentação
  - Fichamento
  - Resenha

# Por que **Ler** Textos Científicos?

## ■ Estudante

- Aprofundar o conhecimento sobre tópicos da área de estudo
- Obter fundamentação para trabalhos na área
- Observar exemplos de estrutura típica e forma de escrita dos trabalhos

## ■ Pesquisador

- Conhecer o que tem sido produzido na área
- Identificar os desafios que permanecem e os novos caminhos

## ■ Profissional

- Conhecer as inovações na área

## Assessing Green Strategies in Peer-to-Peer Opportunistic Grids

Lesandro Ponciano · Francisco Brasileiro

Received: 15 November 2011 / Accepted: 22 June 2012  
© Springer Science+Business Media B.V. 2012

**Abstract** Opportunistic peer-to-peer (P2P) Grids are distributed computing infrastructures that harvest the idle computing cycles of computing resources geographically distributed. In these Grids, the demand for resources is typically bursty. During bursts of resource demand, many Grid resources are required, but on other occasions they may remain idle for long periods of time. If the resources are kept powered on even when they are neither processing their owners' workload nor Grid jobs, their exploitation is not efficient in terms of energy consumption. One way to reduce the energy consumed in these idleness periods is to place the computers that form the Grid in a "sleeping" state which consumes less energy. In Grid computing, this strategy introduces a tradeoff between the benefit of energy saving and the associated costs in terms of increasing the job response

time, also known as makespan, and reducing the hard disks' lifetime. To mitigate these costs, it is usually introduced a timeout policy together with the sleeping state, which tries to avoid useless state transitions. In this work, we use simulations to analyze the potential of using sleeping states to save energy in each site of a P2P Grid. Our results show that sleeping states can save energy with low associated impact on jobs' makespan and hard disks' lifetime. Furthermore, the best sleeping strategy to be used depends on the characteristics of each individual site, thus, each site should be configured to use the sleeping strategy that best fits its characteristics. Finally, differently from other kinds of Grid infrastructures, P2P Grids can place a machine in sleeping mode as soon as it becomes idle, i.e. it is not necessary to use an aggressive timeout policy. This allows increases on the Grid's energy saving without impacting significantly the jobs' makespan and the disks' lifetime.

**Keywords** Energy saving strategies · Peer-to-peer Grids · Resource availability · Sleeping states

### 1 Introduction

Opportunistic Grids are distributed computing infrastructures that harvest the idle computing

### RESEARCH

### Open Access

## Considering human aspects on strategies for designing and managing distributed human computation

Lesandro Ponciano\*, Francisco Brasileiro, Nazareno Andrade and Livia Sampaio

### Abstract

A human computation system can be viewed as a distributed system in which the processors are humans, called workers. Such systems harness the cognitive power of a group of workers connected to the Internet to execute relatively simple tasks, whose solutions, once grouped, solve a problem that systems equipped with only machines could not solve satisfactorily. Examples of such systems are Amazon Mechanical Turk and the Zooniverse platform. A human computation application comprises a group of tasks, each of them can be performed by one worker. Tasks might have dependencies among each other. In this study, we propose a theoretical framework to analyze such type of application from a distributed systems point of view. Our framework is established on three dimensions that represent different perspectives in which human computation applications can be approached: quality-of-service requirements, design and management strategies, and human aspects. By using this framework, we review human computation in the perspective of programmers seeking to improve the design of human computation applications and managers seeking to increase the effectiveness of human computation infrastructures in running such applications. In doing so, besides integrating and organizing what has been done in this direction, we also put into perspective the fact that the human aspects of the workers in such systems introduce new challenges in terms of, for example, task assignment, dependency management, and fault prevention and tolerance. We discuss how they are related to distributed systems and other areas of knowledge.

**Keywords:** Human computation; Crowdsourcing; Distributed applications; Human factors

### 1 Introduction

Many studies have focused on increasing the performance of machine-based computational systems over the last decades. As a result, much progress has been made allowing increasingly complex problems to be efficiently solved. However, despite these advances, there are still tasks that cannot be accurately and efficiently performed even when the most sophisticated algorithms and computing architectures are used [1,2]. Examples of such tasks are those related to natural language processing, image understanding and creativity [3,4]. A common factor in these kinds of tasks is their suitability to human abilities; human beings can solve them with high efficiency and accuracy [1,2].

In the last years, there has emerged a new computing approach that takes advantage of human abilities to execute these kinds of tasks. Such approach has been named *Human Computation* [1,5].

Applications designed to execute on human computation systems may encompass one or multiple tasks. They are called *distributed human computation applications* when they are composed of multiple tasks, and each individual task can be performed by a different human being, called *worker*. In the last years, distributed computing systems have been developed to support the execution of this type of application. They gather a crowd of workers connected to the Internet and manage them to execute application tasks. The precursor of such systems is reCAPTCHA [6]. Currently, there is a broad diversity of distributed human computation applications and distributed systems devoted to execute them, such as: games

\*Correspondence: [lesandrop@isd.ufcg.edu.br](mailto:lesandrop@isd.ufcg.edu.br)  
Federal University of Campina Grande, Department of Computing and Systems, Av. Aprígio Veloso, 882 – Bloco CO, 58429-900, Campina Grande – PB, Brazil

This work has been funded by CNPq Brazil (grants 560262/2010-8 and 305858/2010-6) and the European Commission through the DEGISCO project (grant agreement nr RI-261561).

L. Ponciano · F. Brasileiro (✉)  
Departamento de Sistemas e Computação,  
Universidade Federal de Campina Grande,  
Av. Aprígio Veloso, s/n - Bloco CO, 58429-900,  
Campina Grande, PB, Brazil  
e-mail: [lfbic@dscc.ufcg.edu.br](mailto:lfbic@dscc.ufcg.edu.br)

L. Ponciano  
e-mail: [lesandrop@isd.ufcg.edu.br](mailto:lesandrop@isd.ufcg.edu.br)

# Como Ler um Artigo?

## Primeira Leitura

*Título, resumo, introdução, título de seções e subseções, referências*

- Identificar
  - Que tipo de artigo é? Medição? Análise de um sistema existente? Descrição de um protótipo?
  - Está relacionado com quais outros artigos? Qual a base teórica usada para analisar o problema?
  - As premissas são válidas?
  - Qual é a principal conclusão?

# Como Ler um Artigo?

## Segunda Leitura

*Figuras, diagramas, tabelas e outras ilustrações, referências*

- Analisar
  - Os eixos dos gráficos estão claramente rotulados? O que eles indicam? As relações fazem sentido?
  - Os resultados mostram barras de erro ou teste de hipótese que mostrem a confiança estatística dos resultados?
  - Quais artigos são citados, que são importantes, e que eu ainda não li e que devo ler?

# Como Ler um Artigo?

## Terceira Leitura

*Todo o artigo integralmente*

- Recriar mentalmente o trabalho usando as mesmas premissas dos autores
  - Quais são as inovações? Quais são os pontos fortes? Quais são os pontos fracos?
  - Quais são as premissas e falhas ocultas?
  - Quais ideias interessantes eu tenho na medida em que faço essa recriação?

# Fichamento

Técnica de estudo pessoal em que o registro é feito em fichas com resumo de ideias principais do texto lido

## Fichamento de Citações

Frases mais importantes que são citadas ao longo do texto. Transcrição exata de como estão escritas.

## Fichamento de Conteúdo

Esquema e opinião próprias do leitor a partir do texto, é um texto corrido, um resumo ou resenha

## Fichamento Bibliográfico

Ideias e conceitos que aparecem no texto são organizadas e descritas com as palavras do leitor, e indicadas as páginas onde aparecem no texto



# Volunteers' Engagement in Human Computation for Astronomy Projects

Ponciano, Lesandro; Brasileiro, Francisco; Simpson, Robert; Smith, Arfon. "Volunteers' Engagement in Human Computation for Astronomy Projects," in *Computing in Science & Engineering*, vol. 16, no. 6, pp. 52-59, Nov.-Dec. 2014. doi: [10.1109/MCSE.2014.4](https://doi.org/10.1109/MCSE.2014.4)

## 1. Fichamento de Conteúdo

O artigo caracteriza dois projetos de ciência cidadã na área de astronomia, são eles: GalaxyZoo e The Milky Way Project. Nesses projetos, os voluntários executam tarefas de computação humana, como classificação ou marcação de galáxias em imagens captadas por telescópios. As tarefas são executadas por meio de um sistema web chamado Zooniverse. O objetivo do estudo é caracterizar o engajamento de voluntários ao executarem tarefas dos projetos nesse sistema. É usado um conjunto de dados coletados do sistema contendo 1.031.0994 mil execuções de tarefas por 110.302 voluntários ao longo de mais de dois anos. A caracterização é realizada por meio de quatro métricas (*frequency*, *daily productivity*, *typical session duration* e *devoted time*), análise de correlação entre as métricas e modelagem de distribuição. Os resultados mostram que o conjunto de voluntários que atua nos projetos se subdivide em "voluntários transientes", aqueles que executam tarefas no projeto uma vez e não voltam mais, e "regulares", aqueles que executam tarefas no projeto por pelo menos dois dias. Os transientes são a maioria (67%). Apesar de serem a minoria, os regulares são extremamente importantes, pois eles executam a maior parte das tarefas disponíveis no sistema (78%). Nos dois projetos, a métrica *frequency* pode ser modelada por uma distribuição Zipf, as métricas *daily product activity*, *typical session duration* e *devoted time* podem ser modeladas por uma distribuição Log Normal. A execução de tarefas no sistema e a chegada de novos voluntários no sistema ocorrem em dias de rajadas (*busy days*), dias nos quais há cobertura na mídia (ex. site da BBC) que leva muitas pessoas a acessarem a página Web do projeto.

## 2. Fichamento Bibliográfico

- *Frequency* (frequência) é o número de dias em que o voluntário esteve ativo executando tarefas no sistema/projeto (página 54).
- *Daily productivity* (produtividade diária) é a quantidade média de tarefas que o voluntário executa por dia em que está ativo (página 54).
- *Typical session duration* (duração média da sessão) é a quantidade de tempo seguido que o usuário permanece executando tarefas no projeto/sistema (página 54).
- *Devoted time* é o total de tempo que o voluntário permaneceu executando tarefas no sistema contando todas as suas participações, ou seja, somando as horas em todos os dias em que esteve ativo (página 54).

## 3. Fichamento de Citações

- "To take advantage of these human abilities, a new computing approach—called human computation—has emerged to let humans perform the tasks for which there's still no satisfactory solution via today's silicon-based computers"
- "Volunteer thinking projects are analogous to volunteer computing projects, such as SETI@home"
- "Our results suggest that we can broadly divide participants into transient volunteers (those who execute tasks only one day and don't return) and regular volunteers (those who return at least one more day to execute more tasks after executing the first task in the project)."
- "Given these intervals, we use the threshold-based methodology [11] to determine the sessions."
- "There's a strong correlation between volunteers' frequency and their devoted time to the project"

<https://lesandrop.github.io/site/teaching/IP1/Fichamentos/ExemploDeFichamento1.html>

# Search Algorithms for Regression Test Case Prioritization

Li, Zheng; Harman, Mark; Hierons, Robert M. (2007) "Search Algorithms for Regression Test Case Prioritization", *IEEE Transactions on Software Engineering*, vol. 33, no. 4, pp. 225-237, doi: [10.1109/TSE.2007.38](https://doi.org/10.1109/TSE.2007.38)

## 1. Fichamento de Conteúdo

Testes de regressão são executados em software para verificar se uma mudança introduziu algum erro no software. É um processo demorado. A priorização dos casos de teste visa permitir que falhas (testes que não passam) sejam descobertos o quanto antes. Li, Harman e Hierons (2007) defendem que não há solução ótima para o problema, podendo ele ser mapeado no problema da mochila, que é provadamente NP-difícil. Assim, eles investigam o desempenho de heurísticas na solução desse problema. Em particular, heurísticas baseadas em algoritmos de busca (Hill Climbing e Genetic Algorithms) e algoritmos gulosos (Greedy, Additional Greedy, e 2-Optimal Greedy). Experimentos são realizados com seis programas que variam de 374 a 11.148 linhas de código (LOC, lines of codes) e possuem suites de teste de tamanho médio que variam de 230 a 4350 testes. Os resultados mostram que as heurísticas Additional Greedy, 2-Optimal e Genetic Algorithm são melhores que Greedy Algorithm puro. A explicação é que nesse tipo de problema há muitos ótimos locais, principalmente em grandes programas, de modo que, algoritmos baseados em busca global tendem a ser melhores do que algoritmos de busca local.

## 2. Fichamento Bibliográfico

- O algoritmo de Busca Gulosa (Greedy Algorithm) trabalha com o princípio do "próximo melhor", calcula um peso de cada teste em termos de maximizar a métrica de interesse (como a cobertura do código), e escolhe como próximo teste a ser executado, aquele que possui maior peso.
- O algoritmo de Busca Gulosa Adicional (Additional Greedy Algorithm) calcula o peso de cada teste buscando definir um próximo teste de modo a maximizar a parte dos testes que não foi atingida pelas decisões anteriores. Ou seja, está sempre buscando a parte adicional, não é métrica "geral".
- O 2-Optimal Algorithm seleciona os próximos 2 testes que juntos maximizam a métrica de interesse.
- Escalada ou subida da colina (Hill Climbing) é um algoritmo subótimo ou de ótimo local, seleciona o próximo teste baseado no "vizinho" do teste atual que gera melhor resultado na métrica de interesse.
- Algoritmo Genético (Genetic Algorithms) define um cromossomo composto por todos os testes. Cada posição do cromossomo indica ordem em que respectivo teste será executado. A função de aptidão do indivíduo é calculada baseada no valor obtido para o indivíduo na ordem do indivíduo na população. São realizadas mutações e cruzamentos. (No texto não está muito clara a condição de término)

## 3. Fichamento de Citações

- "Test case prioritization (...) orders test cases so that the test cases with highest priority, according to some criterion (a "fitness metric"), are executed first."
- "Given a function  $f$  that assesses the rate of achievement of code coverage, an efficient solution to the test case prioritization problem would provide an efficient solution to the knapsack problem, which is known to be NP-hard [7]. Thus, prioritization techniques for code coverage are necessarily heuristic [18]."
- "The following two research questions motivated this study: Q1. Which algorithm is most effective in solving the test case prioritization problem for regression testing? Q2. What factors could affect the efficiency of algorithms for the test case prioritization problem for regression testing?"
- "The results of the empirical study show that the Additional Greedy, 2-Optimal, and Genetic Algorithms always outperform the Greedy Algorithm. These results for the programs studied are shown to be statistically significant for all programs studied."

<https://lesandrop.github.io/site/teaching/IP1/Fichamentos/ExemploDeFichamento2.html>

# Importância e Uso do Fichamento

- O fichamento deve permitir ao leitor obter as principais informações do artigo **sem precisar consultá-lo novamente**
- Por ser muito trabalhoso fazer o fichamento, é importante **realizar o fichamento dos artigos que sejam muito relevantes** ao estudo do leitor
- Textos produzidos no fichamento podem ser a base para contextualizar **os trabalhos relacionados** em futuros trabalhos científicos a serem escritos pelo leitor

# Resumos pela ABNT NBR 6028

## ■ Resumo crítico

- Redigido por especialistas como uma análise crítica
- Também chamado de resenha

## ■ Resumo indicativo

- Indicação de pontos principais do documento
- Não dispensa a leitura do documento, pois não é auto-contido

## ■ Resumo informativo

- Informa ao leitor finalidades, metodologia, resultados e conclusões do documento
- Pode dispensar a leitura do documento, pois é auto-contido

# Resenha

- Texto contendo
  - síntese dos principais aspectos de um documento
  - informações e argumentos detalhados sobre o documento
  - juízo de valor sobre o conteúdo do documento
- Tem o papel importante de indicar se o leitor da resenha deve ou não ler o trabalho
- A resenha científica deve ser uma síntese e crítica fundamentada do trabalho a que se refere
  - Expor contradições, os furos e as inconsistências

# Estrutura da Resenha

- 1) Título da resenha
- 2) Referência completa do documento a que a resenha se refere
- 3) Introdução
- 4) Síntese do documento
- 5) Crítica ao documento
- 6) Conclusão

# Fichamento *versus* Resenha

- São formas de síntese diferentes e atendem a propósitos diferentes
  - Faz-se o fichamento quando se está lendo o texto
  - Faz-se o resumo quando já se compreendeu o texto
  - O fichamento é algo pessoal e a resenha é algo formal
- Ambos podem ser usados
  - Pode-se primeiro fazer um fichamento e depois fazer uma resenha a partir dele

# Atividade de Fixação

Sobre fichamento é **incorreto** afirmar:

- a) É uma técnica de escolha de artigos relevantes para uma pesquisa científica.
- b) É algo de uso pessoal de cada pesquisador na pesquisa científica.
- c) Permite o delineamento dos trabalhos relacionados à pesquisa sendo feita.
- d) Tem relação direta com a análise da literatura científica da área do pesquisador.

# Referências

Keshav, S. “How to Read a Paper”. Disponível em:

<https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf> Acesso em: 07 jan. 2020.

ACEVEDO, Claudia Rosa; NOHARA, Jouliana Jordan. Como fazer monografias : TCC, dissertações e teses. 4. ed. rev. e atual. São Paulo: Atlas, 2013. 1 recurso online ISBN 9788522476831.

CASTRO, Cláudio de Moura. Como redigir e apresentar um trabalho científico. São Paulo, SP: Pearson, 2010. 1 recurso online ISBN 9788576058793.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6028: Informação e documentação - Resumo - Apresentação. Rio de Janeiro. 2003. Disponível em: <http://www.abnt.org.br/> Acesso em: 06 jan. 2021.



# Introdução à Pesquisa em Informática

**Prof. Dr. Lesandro Ponciano**

<https://orcid.org/0000-0002-5724-0094>