

browserCloud.js - A federated community cloud served by a P2P overlay network on top of the web platform

David Dias

Thesis to obtain the Master of Science Degree in P2P Networks, Cloud computing and Mobile Applications
BSc in Communication Networks

Examination Committee

Chairperson: Prof. Doutor.

Supervisor: Prof. Doutor. Luís Manuel Antunes Veiga

Member of the Committee: Prof. Doutor. João Dias Pereira

March 2015

Acknowledgements

THANK YOU EVERYBODY :D

20th of September, Lisbon

David Dias

**–To all of the first followers, you
undoubtly changed my life.**

Abstract

Grid computing has been around since the 90's, its fundamental basis is to use idle resources in geographically distributed systems in order to maximize its efficiency, giving researchers access to computational resources to perform their jobs (e.g. studies, simulations, rendering, data processing, etc). This approach quickly grew into non grid environments, causing the appearance of projects such as SETI@Home or Folding@Home, that use volunteered shared resources and not only institution-wide data centers as before, creating the concept of Public Computing. Today, after having volunteering computing as a proven concept, we face the challenge of how to create a simple, effective, way for people to participate in this community efforts and even more importantly, how to reduce the friction of adoption by the developers and researchers to use this resources for their applications. This work explores current ways of making an interoperable way of end user machines to communicate, using new Web technologies, creating a simple API that is familiar to those used to develop applications for the Cloud, but with resources provided by a community and not by a company or institution.

Resumo

IGUAL AO ABSTRACT MAS EM PORTUGUÊS

Palavras Chave

Computação na Nuvem, Redes entre pares, Computação voluntária, Partilha de ciclos, Computação distribuída e descentralizada, Plataforma Web, Tolerância à falhas, Mecanismo de reputação, Nuvem comunitária

Keywords

Cloud Computing, Peer-to-peer, Voluntary Computing, Cycle Sharing, Decentralized Distributed Systems, Web Platform, Javascript, Fault Tolerance, Reputation Mechanism, Community Cloud

Index

1	Introduction	1
1.1	Overview	3
1.2	Problem Statement	3
1.3	Extended motivation and Roadmap	3
1.4	Research Proposal	4
1.5	Contributions	4
1.6	Publications	4
1.7	Structure of the thesis	4

List of Figures

List of Tables

1 Introduction

“Your system can fail no matter how well you thought you tested it... what users will not tolerate is losing their data”. – ¹

Today, in the information communications technology landscape, with the introduction of social networks, search engines, Internet of things, which eventually will drive home and vehicle automation, user data has been growing at a large pace. The store, transfer, processing and analysis of all this data brings considerable new knowledge breakthroughs, enabling us to optimize systems towards a better and enhanced experience. However, how to use the information available to achieve these breakthroughs has been one of the main challenges since then.

Another challenges, is the fact that typically today, user generated data is controlled by some entity, company or organization, which holds the right to keep this information private, exploiting user data for their own goals and business. In order to enable more people to use Big Data analysis, we need to reduce the cost that is inherent to process all this user information, which typically need big amounts of CPU cycles for processing, analysis and inference.

Currently addressing this issues in part, Cloud computing has revolutionized the computing landscape mainly due to key advantages to developers/users over pre-existing computing paradigms, the main reasons are:

- Virtually unlimited scalability of resources, avoiding disruptive infrastructure replacements.
- Utility-inspired pay-as-you-go and self-service purchasing model, minimizing capital expenditure.
- Virtualization-enabled seamless usage and easier programming interfaces.

¹Lehene C. HStack, <http://hstack.org/why-were-using-hbase-part-2>

- Simple, portable internet service based interfaces, straightforward for non expert users, enabling adoption and use of cloud services without any prior training.

Grid computing had offered before a solution for high CPU bound computations, however it has high entry barriers, being necessary to have a large infrastructure, even if just to execute small or medium size computing jobs. Cloud computing solves this by offering a solution “pay-as-you-go”, which transformed computing into an utility.

Still, even though we are able to integrate several Cloud providers into an open software stack, Cloud computing relies nowadays on centralized architectures, resorting to data centers, using mainly the Client-Server model. In this work, we pursue a shift in this paradigm.

Unlike the conventional approach to make Cloud Computing ‘green’ (i.e. Green Computing) by improving datacenter’s efficiency through expensive and strictly centralized control, our vision entails a shift in perspective, by enabling each user to contribute to this effort, leveraging his/her idle computing resources (sometimes up to 70% of power wasted), and thus reducing overall environmental footprint. Thus browserCloud.js resources are provided in a voluntary manner by common Internet users that want to share their idle computer cycles and storage available, while browsing the web, without having the concern to setup any application or system to do so.

Community Clouds are not a novelty in the Distributed Systems research area. However, existing models have been developed to follow the client-server model, transporting the data to the place where the computation will take place, which causes big bottlenecks in network traffic, limiting the amount of computed units done in a delimited window of time. One of browserCloud.js goals is exactly to mitigate this bottleneck by taking the computation (the algorithms that will perform operations over the data) to the machines where the data is stored.

To accomplish this, we propose a new approach to abandon the classic centralized Cloud Computing paradigm, towards a common, dynamic, and privacy-aware cloud infrastructure. This, by means of a fully decentralized architecture, federating freely ad-hoc distributed and heterogeneous resources, with instant effective resource usage and progress. Additional goals may include: arbitration, service-level agreements, resource handover, compatibility and maximization of host’s and user’s criteria, and cost- and carbon-efficiency models.

This work will address extending the Web Platform with technologies such as: WebRTC, Emscripten, Javascript and IndexedDB to create a structured peer-to-peer overlay network, federating ad-hoc personal resources into a geo-distributed cloud infrastructure, representing the definition made by C.Shirky of what an peer-to-peer means:

“An application is peer-to-peer if it aggregates resources at the network’s edge, and those resources can be anything. It can be content, it can be cycles, it can be storage space, it can be human presence.”, C.Shirky ([Shirky](#).)

Finally, browserCloud.js has the possibility to grow organically with the number of users. The management of these resources is done by an RESTful API, enabling desktop and mobile apps to use the resources available in a way that’s familiar to developers.

1.1 Overview

1.2 Problem Statement

1.3 Extended motivation and Roadmap

Our main goal with this work is to design and implement a system that is able to take advantage of volunteered computer cycles through the most ubiquitous growing platform, the browser. In order to create this system, several components will be developed:

- An efficient local storage module that offers persistence and availability, using browser storage for fast indexing.
- A distributed job scheduler able to receive jobs and coordinate with the nodes inside the network, without having to recur to a centralized control system.
- A job executioner able to receive different assets to perform the jobs (image/video manipulation, calculation, etc), taking advantage of the dynamic runtime available by the predominant language in the browser, javascript.
- A server to work as the entry point for browser to download the code necessary to run browserCloud.js logic. This is the only point that is considered to be centralized in the

network, due to the limitation of browsers being typically behind NAT and not having static IPs

- Structured peer-to-peer overlay network for browsers to communicate directly among themselves, without being necessary to take the data or the computation to a centralized system.
- A client API, RESTful, so it is easy to develop applications for Desktops and mobile platforms without having to change the codebase or building a new SDK
- A command line interface for access like ‘mountable’ partition to the storage in browser-Cloud.js, able to dispatch jobs in a very Unix way, by piping the results from one task to another task.

These components are fully described in section 4. After its development, a proposed evaluation is going to be executed, according to a set of assessment metrics, enabling us to compare the viability of browserCloud.js as a Cloud provider, comparing to existing centralized Cloud systems.

1.4 Research Proposal

1.5 Contributions

1.6 Publications

1.7 Structure of the thesis

Document roadmap: We start by describing the objectives of our solution in Section 2, and then, in Section 3 we present the state of the art for the technologies and areas of study relevant for the proposed work, which are: Cloud computing and Open Source Cloud Platforms (at 3.1), Volunteered resource sharing (at 3.2) and Resource sharing using the Web platform (at 3.3). In Section 4, we present the proposed architecture and respective software stack, moving to the system evaluation present on Section 5.

Bibliography

Shirky., C. Clay shirky's writings about the internet. In <http://www.shirky.com/>.

