Pando: An Easy-to-Deploy P2P Volunteer Computing Platform for the Web
Erick Lavoie, Laurie Hendren, Frédéric Desprez

To cite this version:
Erick Lavoie, Laurie Hendren, Frédéric Desprez. Pando: An Easy-to-Deploy P2P Volunteer Computing Platform for the Web. 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W), IEEE, Sep 2017, Tucson, AZ, United States. hal-01652993

HAL Id: hal-01652993
https://inria.hal.science/hal-01652993
Submitted on 30 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Leverage all your (and your friend’s!) devices to accelerate processing workflows with **Pando: An Easy-to-Deploy P2P Volunteer Computing Platform for the Web**

**Motivation**

Current volunteer platforms in use today (ex: BOINC platform) require a significant amount of effort and money to deploy which limits the volunteer computing approach to a few high-profile projects. Moreover, the effort in installing a client limits the number of participants.

The web platform now offers a combination of excellent execution performance, security sandboxing, and standard communication protocol including WebRTC (peer-to-peer browser communication), and portability across many devices and operating systems, and requires no installation of client software. Moreover, social media enables the quick mobilization of millions of users.

Those capabilities now enable the construction of newer and simpler volunteer computing platforms that may tap into more devices than ever, reach significantly more participants, and make the deployment easy for smaller and even one-shot computation projects.

**Contributions**

A web-based volunteer computing approach that is simple to deploy, does not require a dedicated server, and can leverage a wide variety of existing end-user devices because code is executed in browsers;

Open source implementations in JavaScript of stream abstractions, each individually available through the Node Package Manager (NPM) for reuse in other projects and that are easily composable with a growing list of community-developed pull-stream modules;

An open-source command-line tool (pando), compatible with Unix pipelines that use the stream abstractions as well as all scripts and procedures to replicate our performance experiments on the Grid5000 testbed.

**Design Goals**

**Clear Programming Model**: Base the system on a programming model that is clear and guides a correct implementation;

**Easy Deployment**: Make the system easy to deploy, both for the project owner and the volunteers;

**Usage of Existing Commodity Hardware**: Use existing hardware including workstations, laptops, tablets and mobile phones that individuals, businesses, and universities already possess;

**Scalability**: Allow connecting at least a thousand volunteers;

**Elasticity**: Make the volunteer nodes available quickly;

**Performance**: Show a linear improvement in throughput compared to a single processor;

**Fault-tolerance**: Tolerate individual nodes that suddenly disconnect or stop answering.

**Usage of Existing Hardware**

**Clear and Simple Programming Model**

**Computation Model**

**Design**

**Related Work**

Three generations of projects, as surveyed by Fabiaski and Danielek [2] over more than 40 articles on browser-based voluntary computing published in over 20 years, to which we add a 4th generation:

1st Generation (90s and 2000s): Based on Java and Web Applet (requires installation of plugins)

2nd Generation (2007-2010): Based on JavaScript but really slow

3rd Generation (2010-2015): Based on JavaScript but now fast! (still require server)

4th Generation (2015–…): JavaScript + Peer-to-Peer communication (no server!), browserCloud.js [2], and Pando

**Conclusion and Future Work**

Presented Pando, a new volunteer computing platform designed to be easy to deploy and which does not require a dedicated server and can be scaled to a thousand browser tabs for computation. This makes it useful to leverage hardware investments already made in small- and medium-businesses and university departments.

We will perform an in-depth evaluation of the performance of Pando, support more applications, optimize the performance of single computation nodes, and explore various overlay topologies for applications with more complex communication patterns during computation.

**Links and References**

**Source Code Repository**: [https://github.com/elavoie/pando-computing](https://github.com/elavoie/pando-computing)

**Demo/Teaser**: [https://www.youtube.com/watch?v=29ABvs3wNNI](https://www.youtube.com/watch?v=29ABvs3wNNI)


**Reproduction Steps for Performance Experiments**: [https://github.com/elavoie/pando-handbook#publication-specific-instructions-for-reproducing-experiments](https://github.com/elavoie/pando-handbook#publication-specific-instructions-for-reproducing-experiments)
