



**HAL**  
open science

## PROINFODATA: Monitoring a Large Park of Computational Laboratories

Cleide B. Possamai, Diego Pasqualin, Daniel Weingaertner, Eduardo Todt,  
Marcos A. Castilho, Luis De Bona, Eduardo De Almeida

► **To cite this version:**

Cleide B. Possamai, Diego Pasqualin, Daniel Weingaertner, Eduardo Todt, Marcos A. Castilho, et al.. PROINFODATA: Monitoring a Large Park of Computational Laboratories. 10th IFIP International Conference on Open Source Systems (OSS), May 2014, San José, Costa Rica. pp.226-229, 10.1007/978-3-642-55128-4\_34 . hal-01373110

**HAL Id: hal-01373110**

<https://inria.hal.science/hal-01373110v1>

Submitted on 28 Sep 2016

**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.



Distributed under a Creative Commons Attribution 4.0 International License

# PROINFODATA: Monitoring a Large Park of Computational Laboratories

Cleide L. B. Possamai, Diego Pasqualin, Daniel Weingaertner, Eduardo Todt, Marcos A. Castilho, Luis C. E. de Bona, and Eduardo C. de Almeida

Centro de Computação Científica e Software Livre, Departamento de Informática -  
Universidade Federal do Paraná

{cleide,dpasqualin,danielw,todt,marcos,bona,eduardo}@inf.ufpr.br

## Abstract

This paper briefly presents a model for monitoring a large, heterogeneous and geographically scattered computer park. The data collection is performed by a software agent. The collected data are sent to the central server over the Internet, and stored by the storage system. An on-line portal makes up the visualization system, featuring charts, reports, and other tools for assessing the state of the park. This system is currently monitoring circa 150,000 machines.

## 1 Introduction

Digital inclusion is a major challenge of modern society. The Brazilian government has made large investments in equipment and infrastructure to grant access to information and communication technologies. With the success of installing laboratories in a large territory, arises another challenge: to monitor such giant park, ensuring that all investment meets its social and economic objectives. Aiming to solve this problem, it is necessary to define a model and implementation of IT solutions that enable the monitoring, in an easy and transparent way.

## 2 System Description

The proposed system has three independent modules: *i*) Data Collection System; *ii*) Data Storage System; and *iii*) Visualization System.

### 2.1 Data Collection

The collection module gathers data from computers spread over the country. A client-server scheme is adopted. The agent implements the client Web Service, collecting the information from the source machine and storing it locally in XML format, connecting to the central server and sending the data. The central server

collects this data from all agents, storing them in a database and also providing authentication methods and secure connections.

The implementation of the Web Service client was done in C due to performance issues and minimization of the number of packages that must be sent, since in many regions of Brazil the quality of internet connection is poor. The Web Service server is an Apache Axis because it contains the base protocols and safety extensions implemented. Due to the large amount of machines, the server does not maintain the connection states of the clients.

The computers are distributed by the ministry of education with Linux Educational (<http://linuxeducacional.c3sl.ufpr.br/>), with the monitoring agent installed. Since schools may install another operating system or distribution, the agent has been designed to work across major operating systems and distributions.

In Linux-based systems, data collection was developed using Shell Script language, whereas on machines with Windows operating system, data collection was developed using the Python programming language. In order prevent the server from being overloaded, each agent sends the inventory at random times, distributing the total load along all the day. The data collected by the agent are the identifier code of schools, type of machine, network usage and inventory.

## 2.2 Storage System

This module uses a typical storage architecture for large-scale data (Data Warehouse - DW), oriented to read operations that favors the analysis of large volumes of data. A DW architecture consists of three stages: loading, storing and reading data. This division increases the overall performance of the system where different operations (i.e. writing, reading) are directed to different structures. This also adds versatility, so that it is possible to replace one technology by another without impact on the rest of the system, e.g., through the use of new technologies that may further improve data read performance, e.g., MapReduce [3], and C-Store [4].

The three stages are divided in two separate databases: a temporary data database (Staging Area) and a database of historical records (Data Warehouse).

The Staging Area is a database that receives connections from the web server to write data collected during the day. This database has its data deleted after the consolidation of daily data held by DW. The Data Warehouse is a database responsible for storing all historical data. DW reads and consolidates the data from the staging area, once a day, starting at predefined times.

The Data Mart is a subset of DW, summarized and optimized for reading operations [1]. These subsets are determined according to the interest of the consultations. The goal of DMs is to increase the overall performance by creating specific structures for each universe of reports, such as inventory, availability, audit, and network usage. The database was implemented using technology based on open source, in our case, with the management database system (RDBMS) PostgreSQL ([postgresql.org](http://postgresql.org)). What makes Data Warehouse

a high performance system is that the information is joined in a database in a dimensional way [2], reducing the use of join operations in queries.

### 2.3 Visualization System

The visualization system is composed of a web portal, which is in turn subdivided into a server and a client. The web server is responsible for receiving and processing requests, fetching the requested data from the database (Data Marts), and returning the report or graph to the client. The client application is the web portal itself, viewed through a browser and responsible for presenting an interface that facilitates the search and visualization of monitored data.

The proposed model is based on a multilevel grouping of monitored machines, for example, arranged by geographic location and institution. The visualization system has in the first level an overview of the machine park, showing the total number of monitored machines, or any other aggregation meaningful to the administrator. A second level provides information classified by federation States. Selecting in a third level, the user can view data from a specific city and at the last level the machines are displayed individually.

The first version was developed with the Pentaho ([www.pentaho.com](http://www.pentaho.com)) tool, but due to the lack of some desirable native features, like caching, resulting in a noticeable slowness in requisitions answers, it was replaced by the Play Framework ([www.playframework.com](http://www.playframework.com)) with the Highcharts ([www.highcharts.com](http://www.highcharts.com)).

## 3 Experimental Results

All the tools used in the project are open source software, as well as the final product delivered (<http://git.c3sl.ufpr.br>). The result of the implementation can be seen in the portal <http://proinfodata.c3sl.ufpr.br>.

Figure 1 presents the visualization of availability information of the computer park for a region (a) and for a federation State (b). The pie chart shows global information, whilst the bar chart groups the data by region, and the line chart shows the history of the last six months. When selecting a city, a report containing the data from all machines of monitored schools within that city is generated.

Currently, the number of monitored computers with Linux Educacional is 139,045, with Windows is 246, and with other systems is 4,323, summing up to 143,614 machines. The staging/dw model was tested with approximately 200,000 connections per day, writing data individually and in parallel.

## 4 Discussion and Conclusions

The results show that the proposed model and the subsequent implementation are effective and allow the administrator to manage the set of laboratories distributed over the country, based on a system that is optimized to support the



Fig. 1. Data about availability grouped by Region (a) and State (b)

giant number of clients that should send status information to a central server. The hierarchical approach makes easy to the user to locate the information needed. The color codes used to identify the groups of machines according to the time elapsed since the last contact make easy to recognize problems such failures and machines that were not yet put into operation. According to the Data Mart model, the information of individual machines is consolidated calculating *a priori*, once a day, subtotals for cities, States and regions, accelerating the responsiveness of the system to user queries. One fundamental characteristic is that the system is open source, allowing code inspection and assurance of privacy about personal data. As future work, it could be added artificial intelligence modules able to find anomalous behaviours and to generate alarms to the administrator. Also, since the system is modular and designed to be platform independent there are few obstacles to continuous improvements.

## References

1. Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.*, 10(4):452–483, October 2001.
2. Ladjel Bellatreche and Komla Yamavo Woameno. Dimension table driven approach to referential partition relational data warehouses. In *Proceedings of the ACM Twelfth International Workshop on Data Warehousing and OLAP, DOLAP '09*, pages 9–16, New York, NY, USA, 2009. ACM.
3. Ralf Lämmel. Google’s mapreduce programming model – revisited. *Science of Computer Programming*, 70(1):1 – 30, 2008.
4. Mike Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O’Neil, Pat O’Neil, Alex Rasin, Nga Tran, and Stan Zdonik. C-store: A column-oriented dbms. In *Proceedings of the 31st International Conference on Very Large Data Bases, VLDB '05*, pages 553–564. VLDB Endowment, 2005.