ETHERNET-BASED REMOTE MONITORING IN WIDELY DISTRIBUTED PLANTS

Sorin Ignat* *sorin_i@email.ro*
Dorina Căpățînă* *dorinac@email.ro*
Ovidiu Ghiran* *oghiran@email.ro*

** SC IPA SA Cluj-Napoca Subsidiary*
109 Republicii Str., 400489 Cluj-Napoca, Romania
phone: +40264 596155, fax: +40264 590558, Email: ipacluj@automation.ro

Abstract:

Lately the PC world migrates to Linux-like operating systems and to larger-scale computers. For keeping their customers, the developers of those PC applications will adopt X toolkits as the GUI for the Un*x versions of their software. As a result, the next few years will see the number of X users and developers grow, as X becomes the essential point for software and hardware companies alike.

The degree of configurability is one of X's system power. Colors, fonts, and a wide variety of other resources can be set deeply by the end user.

PostgreSQL is the most advanced open-source database server available; it offers multiversion concurrency control, supporting almost all SQL constructs (including subselects, transactions, and user-defined types and functions), and have a wide range of language bindings available (including C, C++, Perl, Tcl, and Python).

Keywords: enterprise data-flow integration, IEEE 802.3 Ethernet, ORDBMS, C/C++ graphical user interface toolkit.

1. INTRODUCTION

In essence Linux is a free implementation of a Un*x kernel; it is known as a portable operating system taking advantages of a proven technology, and is based on open standards.

The X Window System is an advanced, graphical computing and network environment. It has been projected from the ground up as a multi-user system. X is a client-server, multi-user system. It is not integrated into the operating system, and rides on top of it, like other servers. X is an open standard, and runs on many platforms.

X is on the whole a networking protocol with graphical displaying capabilities. There is a trade-off in using desktop environments (they add some interoperability and ease-of-use features; their functionality overlaps the window manager's). These take memory and system resources to be launched. This aspect is important in implementing a distributed system for enterprise data flow integration activities (fig.1).

Desktop environment should be used without affecting user productivity. The target is keeping most system resources for the applications, and not to consume all memory and CPU usage for the desktop environment necessity.

XFce is a lighter weight, less featureful desktop environment that does not get as much consideration as the others (XFce is based on the GTK+ toolkit). The XFce environment assures a simple and efficient environment for a Linux system.

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES. POSTGRES pioneered many concepts that only became available in a lot of marketable database systems much later.

PostgreSQL uses a client/server model. A session in PostgreSQL acceptance consists of the following cooperating processes:

1. one server process, which supervises the database files, receives connections to the database from client applications, and performs actions on the database on behalf of the clients. The database server program is called postmaster.
2. the user's client (front-end) application that wants to perform database operations. Client applications can be very diverse in nature: a client could be a text-oriented tool, a graphical application, a web server that accesses the database to display web pages, or a specialized database maintenance tool.

PostgreSQL, provides more features than MySQL. These include more SQL functions, server-side procedural languages, and sophisticated methods for date manipulation. PostgreSQL also offers object-relational capabilities and geometric data types.

PostgreSQL is ACID compliant (atomicity, consistency, isolation, durability). The standard table handler for MySQL is not ACID compliant because it doesn't support consistency, isolation, or durability.

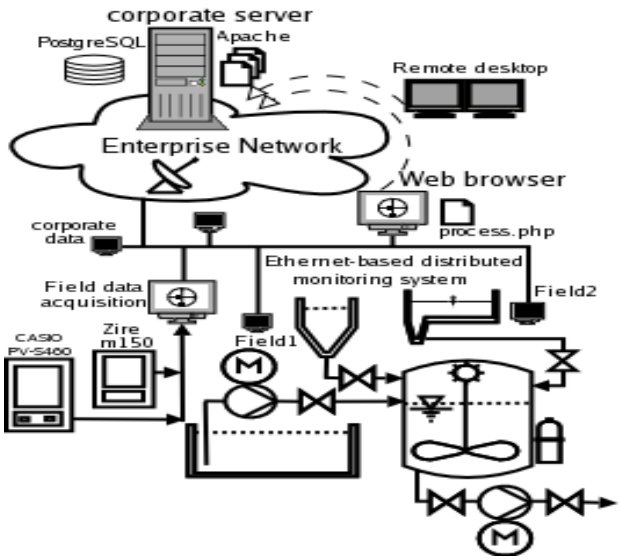

Fig.1 – Distributed monitoring system

C application programmer's interface to PostgreSQL is libpq. libpq is a set of library functions that allow client programs to pass queries to the PostgreSQL backend server and to receive the results of these queries. libpq is also the underlying engine for several other PostgreSQL application interfaces, including libpq++ (C++), libpgtcl (Tcl), Perl, and ECPG [7].

The client and the server can be on different hosts, they communicate over a TCP/IP network connection.

An application program can have several backend connections open at one time, in order to access more than one database.

One can use SSH to encrypt the network connection between clients and a PostgreSQL server, providing an adequately secure network TCP/IP connections with SSH tunnels [2].

There are a few routine maintenance tasks that must be carried out on a regular basis to keep a PostgreSQL server running efficiently [4]. PostgreSQL's vacuuming operation has to be run on a regular basis for several reasons:

a.recovering disk space occupied by updated or deleted rows;
b.updating data statistics used by the PostgreSQL query planner;
c.to protect against loss of very old data due to 'transaction ID' wraparound.

Entire critical level is recommended to be the most secure part of the code. Application modules that interact directly with the supervised process can be developed, par example, using the EZ Widget and Graphics library (EZWGL).

EZWGL includes commonly used widgets; it has integrated support for application resources, geometry management and drag-and-drop; includes OpenGL(TM) like graphics functions.

The system layers that are not interacting directly with the screened process (high-level modules) can be developed using such as the FLTK toolkit.

The Fast Light Tool Kit (FLTK) is a C++ GUI toolkit for UNIX/Linux (X11). FLTK was designed to be statically linked. This was done by splitting it into many small objects and designing it so that functions that are not used do not have pointers to them in the parts that are used, and thus do not get linked in. FLTK works fine as a shared library; so it is now included with several Linux distributions.

2. EXPERIMENTAL RESULTS

Experiments were made using the following test configuration:
a) operating systems: Fedora Core 1, FreeBSD-5.2; b) C compiler: gcc-3.3.3; c) X server: XFree-4.3.0; d) graphic user interfaces use a resolution of 1024x768 pixels, at 16 bits color depth; f) window-managers: fvwm-2.4.17,AfterStep-1.8.11; g) desktop environment: XFce-4.0; h) C GUI toolkit: EZWGL 1.51; i) C++ GUI toolkit: FLTK-1.1.4.

Additionally were used the following packages:
a) PHP language: PHP-4.3.4; b) Apache web server: Apache-1.3.29; c) Database server: PostgreSQL-7.4.1 (C-language library libpq); d) web browser: opera-7.11.

Additional libraries:
a) gtk+-2 toolkit; b) lbpng-1.2.6 library.

For specific data acquisition activities were provided handheld equipment:
a) Pocket Viewer Casio PV-S460 (RS-232 interface); b) Zire-m150 (USB interface).

Pocket Viewer PV-S460 main features are: a) operating system: Casio-OS;
b) memory: 4 MB; c) screen: 160x160 pixels.

Zire-m150 main features are: a) operating system: Palm-OS v4.1;b) memory: 2MB; c) screen: 160x160 pixels.

Zire-m150 handheld can store information in flat-database format – List 1.0.

Application modules have been generated in ELF binary format (fig.2). Some of the remote monitoring activities can be performed over a TCP/IP network using a web browser (mechanism is based on using 'PHP-Apache-PostgreSQL').

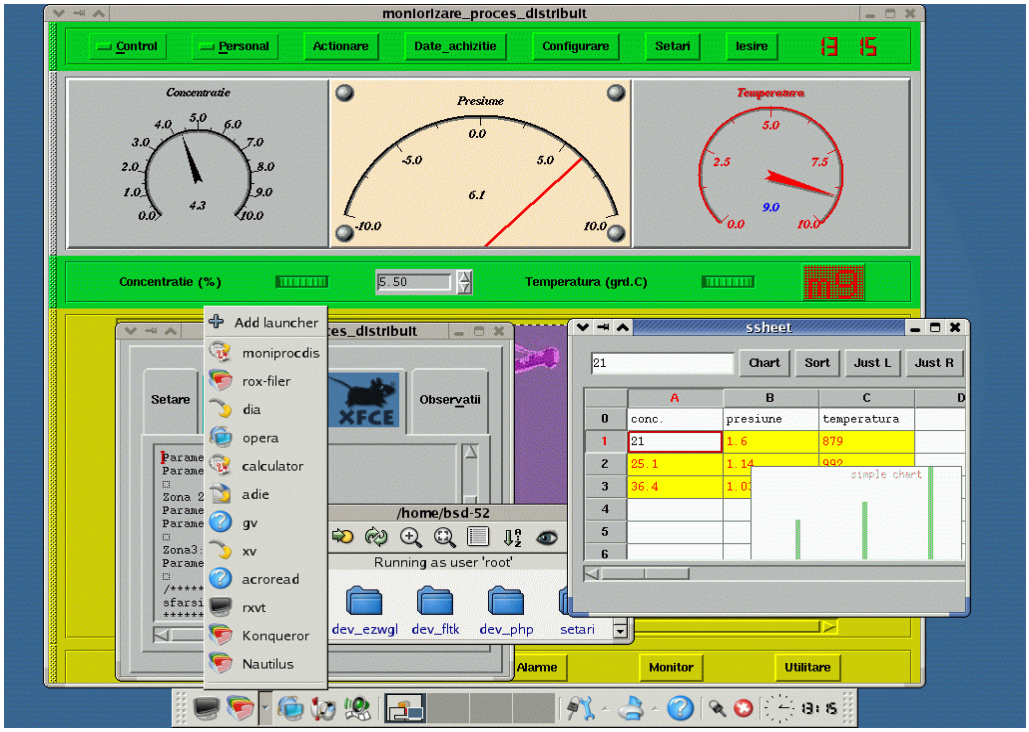

Figure 2 – Monitoring software modules

Certain monitoring information (related to some parameters evolution) is gathered using handheld devices, having data logging capabilities (fig.3). Periodically, information stored in handheld devices is downloaded into a computer for future processing.

Fig.3 – Data logging sub-system

In figure 4 - left side is presented a 'Pocket sheet' application launched on a PV-S460 device (screen capture from); right side is a database (on List 1.0 format) for Zire-m150 devices (imported into a spread sheet application).

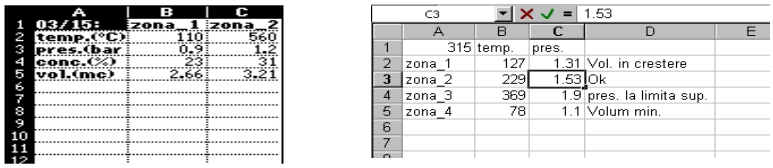

Fig.4 – Parameters acquisition based on handheld devices

Compared with Linux, FreeBSD is richer and more mature. From all flavors of BSD available (NetBSD, OpenBSD, etc.), FreeBSD has remained consistent on i386. FreeBSD for the most part, has very good support for Intel hardware. It is possible to use many different types of hardware like network cards and RAID controllers.

Additionally, FreeBSD has a very large port tree, and has good support for low level system calls. Almost any program that runs on Linux will have a FreeBSD port available. If not, FreeBSD's Linux support lets user run many Linux programs almost natively.

MySQL seems to be focused on the needs of Web developers, and PostgreSQL seems oriented to a wider range of application developers. Postgres in most instances is nearly the same speed as mysql; PostgreSQL can handle more concurrent connections; postgres's security is far more advanced than mysql's.

Postgres offers considerably more options than MySQL, which does not meet all the standard requirements for a relational data management system. Postgres on the other hand, with object-relational features such as inheritance and free datatype definitions, actually goes beyond them.

Compared with Motif or with Xaw/Xaw3d/Xawnet toolkits, EZWGL has a rich widget set, and applications developed have a better functionality and look. Motif is a widget set that is intended to work in conjunction with the Xt/Intrinics, EZWGL is a C library written on top of Xlib, that simplifies the task of graphical user interface (GUI) programming under the X window system environment.

3. FUTURE DEVELOPMENT

In future, is expected that industrial processes will be depicted using color Petri nets. Colored Petri Nets (CP-nets) is a graphical oriented language for design, specification, simulation and verification of systems. It is in particular well-suited for dispersed systems in which communication, synchronization and resource sharing are important.

With the purpose of achieving a better distributed processing, the strength of Petri nets will be combined with the strength of C/C++ programming language. Petri nets will provide the primitives for the description of the synchronization of concurrent processes, while programming languages provide the primitives for the definition of data types and the manipulation of data values.

In future are envisage modules compatibility with other X11R6 platforms; the efforts will be focused on providing robust network services under the heaviest loads and uses memory efficiently to maintain good response times for simultaneous user processes. In this respect will be offered advanced networking, performance, security and compatibility features.

For PV-S460 handheld communication with serial interfaced data logging devices will be developed a protocol (based on tools for pockets, like OWBasic-400).

4.CONCLUSION

The differences between Linux and freeBSD may seem minor, perhaps even insignificant. For security and speed, FreeBSD is the way to go. In any case, Fedora Core, or other Linux distribution, are not inferior. Linux has high-quality desktop uses,

and works well in some server situations. However, the FreeBSD tree is undoubtedly one of the fastest operating system running on Intel hardware (including compilation, file access, network stack, running X).

Linux as a development platform has the following key advantages:
a) is based on thirty years of Unix experience; b) is accompanied by a large base of free software development libraries and tools; c) is backed-up by free environments available for languages like C, C++, Objective C, Tcl/Tk.

X, is actually a convergence of various components: X server, window manager, desktop, etc. The X Window System is a network-transparent GUI, allowing use of remote displays. X is particularly a bandwidth consumer. Troubles do not appear in LAN situations, only if trying to use X over the Internet.

XFce desktop environment has reputations of performing well with low memory. Whatever libraries are used, application has to be modular. Well-designed applications keep the user-interface code separate from the functional code.

Based on a limited feature set, MySQL is very fast. As far as durability is concerned, you might lose some data if the plug is pulled in the middle of a transaction. With PostgreSQL, you have many features and can be confident that application data is safe. PostgreSQL is low-maintenance compared to some other database management systems [5].

Unlike traditional database systems which use locks for concurrency control, PostgreSQL maintains data consistency by using a multiversion model (Multiversion Concurrency Control - MVCC) [1], [3].

For databases, the frequency and scope of the vacuuming operations performed for each of these reasons will vary depending on the needs of each site. Consequently, database administrators have to appreciate these and build up an suitable maintenance stratagem [6].

FLTK comes with complete free source code. FLTK is available under the terms of the GNU Library General Public License (it can be used in commercial software).

5.REFERENCES

[1] Ramez Elmasri and Shamkant Navathe, *Fundamentals of Database Systems*, 3rd Edition, Addison-Wesley, ISBN 0-805-31755-4, August 1999.
[2] C. J. Date, *An Introduction to Database Systems*, Volume 1, Sixth Edition, Addison-Wesley, 1994.
[3] Nels Olson, *Partial indexing in POSTGRES: research project*, University of California, UCB Engin T7.49.1993 O676, 1993.
[4] Stefan Simkovics, *Enhancement of the ANSI SQL Implementation of PostgreSQL*, Department of Information Systems,Vienna Univ. of Technology,November 29, 1998.
[5] A. Yu and J. Chen, The POSTGRES Group, *The Postgres95 User Manual*, University of California, Sept. 5, 1995.
[6] M.Stonebraker,L.A.Rowe,M.Hirohama, "The implementation of POSTGRES", Transactions on Knowledge and Data Engineering 2(1),IEEE,March 1990.
[7] Jim Melton and Alan R. Simon, *Understanding the New SQL: A complete guide*, Morgan Kaufmann, ISBN 1-55860-245-3, 1993.