The Athena Palladium Print System

Brian Handspicker
Digital Equipment Corp.
bd%vines.DEC@decwrl.dec.com

Richard Hart
Digital Equipment Corp.
hart@decvax.dec.com

Mark Roman
Open Software Foundation *
marcus@osf.org

February 7, 1989

Abstract

Large-scale application of traditional print systems to distributed computing environments and modern printing hardware have exposed the limitations of those print systems. *Palladium* is a distributed printing system being developed at Project Athena [Athena85] to address the issues of supporting and maintaining a distributed printing system within a distributed computing environment. It will provide access to print services and centralized management of those services as required by Project Athena. The interfaces to these *Palladium* services are designed to conform with developing international standards. *Palladium* supports the features found on current and recently developed printing devices, and provides, in a distributed environment, the print job queuing and management that users have come to expect.

Introduction

Recent advances in document publishing products and printer technology have outstripped the support provided by traditional print systems. Users may create documents with complicated formats, multiple fonts, and embedded graphics on single user workstations. When these advances in printing capabilities are combined with the new distributed computing environments now in use, it becomes evident that printing systems need to be redesigned.

Project Athena [Athena85] is a distributed computing environment for use by students, faculty and staff at MIT. It currently consists of about 800 hundred workstations and 53 file servers. It is planned that Athena will grow to over one thousand workstations in the near future. The Athena Palladium Print System has been designed to provide equitable access to shared printing resources in this environment, and to provide this service in a manner consistent with the Athena distributed services model.

The Athena system is based on UNIX[1] 4.3 BSD. It was originally developed for an environment consisting of a central processor with a well defined collection of user terminals and other devices attached. To adopt 4.3 BSD to the distributed Athena environment, important pieces of software

*Current address: Open Software Foundation, 11 Cambridge Center, Cambridge, MA 02142

[1]UNIX is trademark of AT&T.

were added to provide authentication [Kerberos88], location of services [Hesiod88], and notification [Zephyr88].

Because the Berkeley Spooler [Campbell83] was written for a central processor environment, it has certain shortcomings that make it difficult to use in a distributed environment:

- The desirability of placing printers, users, and spool queues all on different machines in a distributed network environment was initially unforeseen.
- The Berkeley Spooler does not easily support the centralized management and logging needed at Athena, nor does it easily support the network services we wish to incorporate.
- Authentication of users was provided via */etc/passwd*.
- Access controls were not necessary for printers and queued print jobs.
- It was assumed that printers only read and they never write information.
- Tradeoffs were made based on the assumption that very little processor memory is available.

The goals of the Athena Palladium Print System were developed to address these issues. While the Berkeley spooler has served us well, we favor redesign of the printing system over further extensions. This paper first describes our design goals and how distributed systems design concepts have been applied to meet them. Then, we describe the print system model and the function of the constituent pieces.

Palladium Print System Goals

The goals of the Palladium project at Athena can be split into the following categories:

- **Computing Environment**. Palladium is being designed for use within the Athena [Athena85] distributed environment. Within this environment, Palladium users, print queues, and printers, may all exist on different systems. A user may submit a print job from one workstation, then make inquiries about that job from a different workstation. If a printer malfunctions, jobs originally scheduled for that printer may be rescheduled on some other printer on the network.
- **Printer Support** Palladium must eventually support many types and brands of printing devices. Of particular importance are newer printers that both receive text to be printed, and return status and control information. Also, it must be possible to support any page description language. At Athena, the page description language of choice is Postscript[2] [PostScript87], but others will be supported.
- **System Management**. Centralized system management is an important feature of any distributed system having hundreds (and eventually thousands) of workstations. The Athena system is no exception. Palladium will make all printer and print server state information available through standard interfaces to those with appropriate privileges. It must be possible

[2]Postscript is trademark of Adobe Systems, Inc.

to manage individual printers and print servers remotely, without changing the availability of other printing services. It must also be possible for operators to redirect print jobs to different printers, and generally manage print jobs. Finally, reliable performance and usage data must be available to guide system configuration and tuning.

- **Authentication, Authorization and Accounting**. Palladium will not allow access to print system services and information without proper authorization. This implies reliable authentication of a user's identity. Palladium must also be able to log securely all activity and printer usage so that accurate and credible accounting information can be generated.

- **Standards** The Palladium print system will conform with the emerging European Computer Manufacturers Association (ECMA) printing standard [ECMA88]. Palladium is a reference implementation of that standard. It is hoped that this will allow future printing systems to be more easily integrated into the Palladium system.

Distributed Systems Concepts

The design of the Palladium Print System has been realized through the application of the following distributed systems concepts:

- Client-Server Model
- Name Service-based Server Advertisement
- Remote Procedure Call based Server Access Protocols
- Embedded Tasking Systems

The client-server model proves very useful for the relationship between local user applications and shared resources in the Athena environment. This approach allows many clients to obtain central services and facilitates centralized management. The name service allows the client to obtain easily the address and information necessary to get print service. Name service basing minimizes problems with information skew and update, and maintains the notion of a public, stateless workstation.

The Remote Procedure Call (RPC) paradigm can be used to provide a simple interface to distributed services. Using the RPC service, we need only to provide an interface specification to one of the RPC utilities (an RPC Stub Generator) to generate the access protocol code. The RPC approach was used to shorten development time and reduce problems with data representation.

There are some diffculties, however. RPC paradigms enforce a synchronous relationship between client and server. To equitably handle all requests that may be submitted to the server in a short period of time, and to supply reasonable response to each request, the server must have some way of logically doing multiple tasks at the same time. A tasking system will be used to provide this capability by either allocating time slices of a single process to different tasks or by creating new processes for each task that must be accomplished. The current implementation plan calls for the "time-slice" approach.

Palladium Print System Model

The Palladium Print System Model uses these distributed systems concepts to improve the managability of distributed print systems and provide equitable access to shared printer resources in the Athena distributed environment. In Figure 1, the logical design and communication is illustrated. The model is broken up into three distinct parts: client, server and supervisor. Communication is accomplished through the RPC facility.

The functionality is broken up as follows:

- The print client contacts the name server to find an appropriate print server.
- The server advertises its location via the name server.
- The name server tells the client where to find the print server that matches the user's needs.
- The Palladium server interface allows access to the services which it provides through an RPC interface.
- The server's tasking system allows multiple clients to make requests at logically the same time.
- The server database stores all queued print jobs and Palladium print system configuration information.
- The scheduler passes responsibility for a job off to a specialized process called a 'printer supervisor'.
- The printer supervisor takes responsibility for printing a single job, using a single print format on a single (type of) printer. Note that a printer may support more than one print format, but for any specific job we presume that only one of those formats is exploited.

The separation of functionality exploited in this design allows for greater flexibility in extensibility and management of the system.

Palladium Client

The Palladium Client software uses the Hesiod name server [Hesiod88] to select the Palladium server(s) that satisfy the requirements of the user and to locate the desired server in the distributed environment. Hesiod translates the logical printer name supplied by the user into an address for the print server that supports the desired printer. As more that one server may support a given logical printer, it is up to the client routines to chose the best (i.e. shortest queue length, shortest walk, etc.) print server from those offered by Hesiod. The client finally submits the request to this server. The client may also provide some local job management in the case of a print request.

Palladium Server

The Palladium server handles requests submitted by the Palladium Clients and processes those requests. Most requests are processed as they are submitted, returning the results immediately. **Print** and **ModifyPrintJob** requests are validated and, if accepted, added to the Job Info Database.

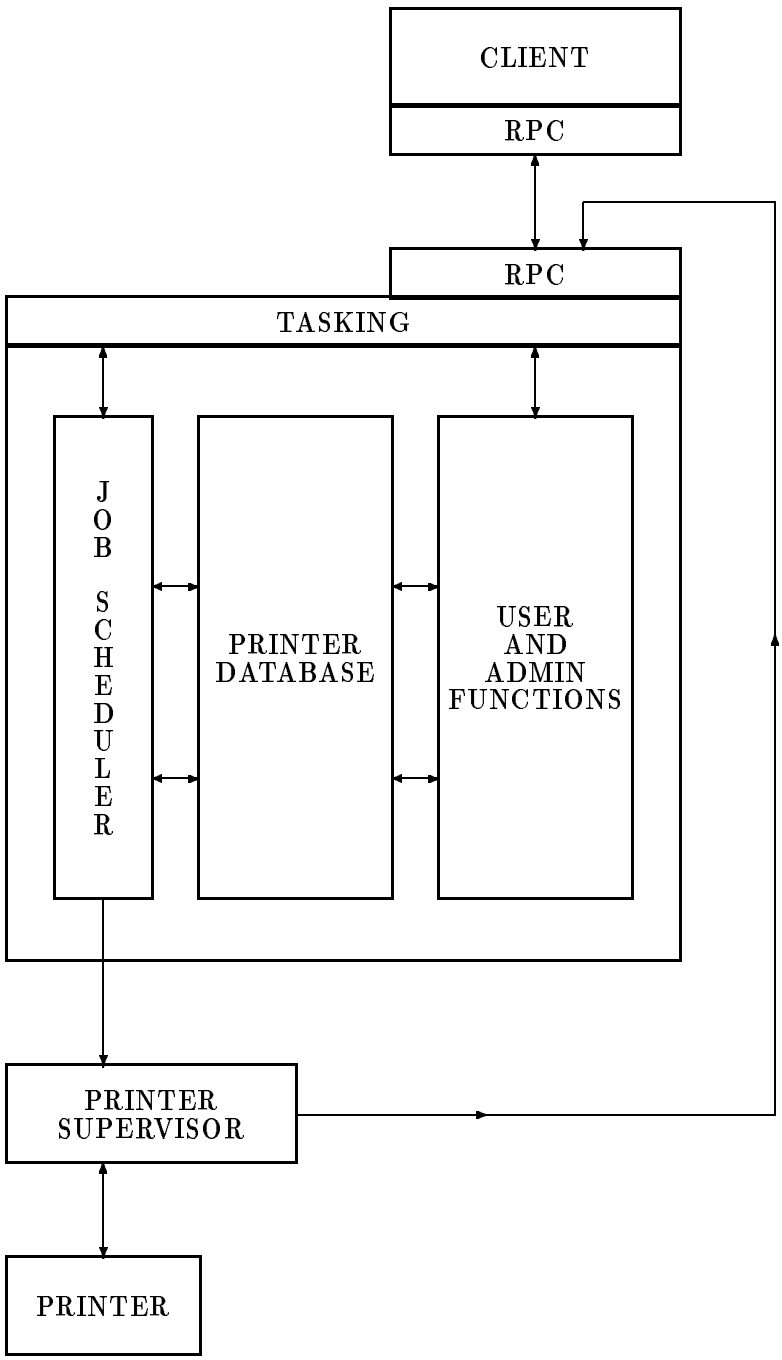

Figure 1: Palladium Print System Model

The Job Scheduler locates the right printer which satisfies the job requirements, schedules the job on the printer and tracks the progress of the job. The server collects events reported to it by the supervisors in the Job Info Database and reports status and significant events to the user. The server also logs accounting and status information reported to it by the supervisors.

The Server does not understand the syntax or the semantics of the file content. The Job Scheduler merely matches the job requirements with the capabilities of the printers managed by this server. In this way the server handles a broad range of file types and printer types.

The server may be implemented as several processes or as several tasks within a single process. However, it must run off of single job, resource and printer databases and use a single centralized scheduling or queuing mechanism for all the jobs and printers that it manages. In doing so, it provides equitable shared access.

Service Interface and RPC

The Palladium Server Interface is presented as a set of library routines (PdLib). These routines are accessed through the RPC protocol described earlier. Palladium Clients use RPC library routines to obtain the following service from a Palladium Server:

1. Print
2. Modify Print Job (that was previously submitted)
3. Read Job Status
4. Cancel Job
5. List Jobs
6. Read Print Server Characteristics
7. Read Print Server Status
8. Read Printer Characteristics
9. Read Printer Status

In addition to these user requests there are a series of administrative requests used to manipulate jobs, servers and supervisors. These functions allow the administrator to:

1. start/stop queueing, printers and servers
2. pause and release jobs for printing
3. change the characteristics of the various server and printer instances.

These operations will be implemented within a single administrative client, although that is not mandatory.

For Palladium, the HRPC software from The University of Washington [HRPC88] was chosen. HRPC is a common interface to a number of RPC systems. Although HRPC is a common subset of the supported RPC systems, it provides the functionality necessary for Palladium and for the ECMA services. HRPC provides a generic interface for remote procedure calls, and its use will help to keep Palladium independent of underlying protocols and RPC definitions.

Authorization and Access Controls

For each request made by a client, the server checks a well-known access list to determine if the user is authorized to make the request. Access lists are maintained for each operation. In a network environment, secure authentication of the client's identity is required; Palladium uses Kerberos for authentication.[Kerberos88].

File Transfer Paradigms

When a print job is submitted to a print server, the contents of the file(s) to be printed must be made available to the server at some point. There are three paradigms for print job file transfer:

1. Pass with Request (as a Print Operation parameter)
2. Retrieve by Server
3. Copy by Client

Each of these paradigms has its own advantages and disadvantages when used in the distributed Athena environment. The Palladium print system uses each of these paradigms where it is most appropriate.

The "pass by request" paradigm can be used for communications between clients and servers and between the server and the supervisor. This method will be used as a default when other, hopefully more efficient, methods do not exist.

The "retrieve by server" method is used for communications between Palladium supervisors (the "server") and Palladium servers (the "client"). Palladium servers will make a publically readable directory available that can be mounted by supervisors to read files to be printed. The supervisor will then copy the file to the printer when it wishes to do so.

The "copy by client" method is used for communications between the Palladium client and the Palladium server. Palladium servers will make a writable space available where clients may copy files to be printed. Essentially, the server returns to the client a file handle and permission to write that file. Such permission is revoked after the copy is complete.

Palladium Server Database

The Print Server Database contains high-level definitions of logical printer capabilities (e.g. names of media supported, finishing features supported, etc.), definitions of resources (e.g. sizes and colors of media, location of stapling, etc.), and print job request information. The print job information consists of the text to be printed and the parameters which will control the processing of the print request. The database also contains capability and location information for the supervisors which the server controls.

The information in the Palladium database is split into two sections. The first section contains static, relatively slow to change, information. This information describes the Palladium configuration as a set of definitions of printers, logical printers, servers, and supervisors. Each server and supervisor reads a configuration file when it begins operation and whenever requested thereafter to get the configuration information it needs.

The examples show the kind of information that a server or a supervisor might read from a configuration file. These configuration files may be edited by a system manager, or, more normally,

they are generated automatically from the SMS System Management database. The first example shows an entry for a printer that will be used by its supervisor. It contains all the information necessary to configure and run a particular printer.

```
!
! A Configuration entry for a printer and its supervisor
!
begin entity = printer(satchmo)
        Type = ln03r
        GeographicLoc = E40-358
        DeviceAddress = /dev/ttycp
        HostName = castor.mit.edu
        ProgramNumber # 10489
        PrintFileDirectory = /mit/pss2/spool/Pd
        SetupProgram = /mit/pss2/etc/Pd/Setup.sh
        BannerDir = /mit/pss2/etc/Pd/BannerDir
        begin info
                Speed # 9600
                AsciiFilter = /mit/pss2/etc/Pd/AsciiFilter
                DviFilter = /usr/lib/tex/dvi2ps
                LocalFlagsClear # 0
                LocalFlagsSet # 040040
                SgFlagsClear # 0376
                SgFlagsSet # 001
        end info
        begin DefaultChars
                LogicalPrinters = (branford, wynton)
                Media = (8.5x11-white)
                Server = dizzy
        end DefaultChars
end entity
```

Each printer has a particular type. The printer **satchmo** has a type of **ln03r**. Each printer type supports a specific set of of Data Syntaxes, Fonts, Media, and Forms. These are listed in the configuration file entry for the ln03r type entry. Individual printers may restrict these features if they require operator intervention, disk space, or something else that may not be locally available. For example, **satchmo** restricts Media to be just 8.5 x 11 white paper.

Finally, logical printers group desired functional capabilities that may be supplied by some set of physical printers. The configuration file entry for the logical printer **branford** is shown below. The server **dizzy** makes the logical printer **branford** available to the world. When a client makes a request of **dizzy** for services on **branford**, the printer **satchmo** will be used based on the configuration file entries shown here.

```
!
! A Configuration entry for the ln03r printer type.
```

```
!
begin entity = type(ln03r)
        DataSyntax = (PostScript, Ascii, Dvi)
        Fonts = (AvanteGarde-Book, AvanteGarde-BookOblique, \
                 AvanteGarde-Demi, AvanteGarde-DemiOblique, \
                 ....
                 Times-Italic, Times-BoldItalic)
        Media = (8.5x11-white, 8.5x11-green, 8.5x14-white, 8.5x14-green)
        Forms = (MIT-LetterHead, Athena-LetterHead)
end entity
!
! A sample configuration file entry for a logical printer
!
begin entity = logprinter(branford)
        begin LogPrinterChars
                DataSyntax = (PostScript)
                Media = (8.5x11-white, 8.5x14-white)
        end LogPrinterChars
        begin JobDefaults
                DocumentFormat = PostScript
                StartPage = Full
                SeparatorSheet = Brief
                Media = (8.5x11-white)
        end JobDefaults
end entity
!
! A sample server Configuration file entry
!
begin entity = server(dizzy)
        ServerHost = zippy.mit.edu
        SpoolDir = /usr/spool/Palladium
end entity
```

Besides the static information found in the configuration files, print job request information is also stored in the Palladium server database. This more dynamic information is stored by the server in the spool directory specified in the configuration information for the server. The interface to the Palladium database is the same regardless of whether static or dynamic information is requested. A considerable portion of the database information is cached in memory so that disk accesses are kept to a minimum.

Palladium Printer Supervisor

The Printer Supervisor provides a static process to drive a specific printer (or set of related printers) for the printing of one print job at a time. The Printer Supervisor is entirely responsible for interpreting the job request. This includes interpreting the content of the file(s) and reconciling the instructions provided in the print request parameters with the contents of the file. The supervisor

has complete control over the printing of the job, including the loading of fonts, forms and other electronic resources into the printer, as well as the transfer of the file(s) to the printer.

Supervisor Interface

The printer supervisor provides the same RPC-based interface that the print server provides to clients. this allows the printer supervisor to be accessed by either a print server or by individual clients. The printer supervisor may be configured to either: allow exclusive access by a single Print Server or allow access by 'all comers'. The latter configuration allows a print supervisor to be installed on a workstation to drive a personal printer attached to that workstation. The same interface used by the client to access shared printers through a print server can be used by the client to access a local (potentially non-shared) printer. The only difference to the client is the address advertised in the name server.

Conclusions

The Palladium distributed print system has been designed to use existing software technology to construct a truly distributed printing system that can support current and future printing devices. We are currently implementing and testing the design.

The incorporation of centralized management into the system will allow for easier and more effectively management of print services as advancing technology and user requirements place greater demands on them. It is hoped that basing our development work on an emerging standard will permit our work to be easily augmented and extended by Project Athena and other members of the MIT network community.

Acknowledgments

We would like to acknowledge the efforts of the Palladium design team, Pradeep Chetal, Dave Gray, Jack Woychowski, and Gerhard Messmer, who contributed significantly to the design of the print system. We would also like to thank George Champine and Dan Geer for their persistent support of our efforts.

References

[PostScript87] Adobe Systems, Inc., "PostScript Language Reference Manual," Addison-Wesley, Reading, Massachusetts, (April 1987)

[Athena85] E. Balkovich, S. R. Lerman, and R. P. Parmelee, "Computing in Higher Education: The Athena Experience," *Communications of the ACM*, 28(11) pp. 1214-1224, (November 1985)

[Campbell83] R. Campbell, "4.2BSD Line Printer Spooler Manual," Dept. of Electrical Engineering and Computer Science, Univ. of Calif., Berkeley, (July 1983)

[Nectar88] E. C. Cooper, "The Nectar Threads Package," Computer Science Department, Carnegie Mellon University, (May 1988)

[Hesiod88] S. P. Dyer, "The Hesiod Name Server," in *Usenix Conference Proceedings*, (Winter 1988)

[ECMA88] "Standard ECMA - Document Print Service Description and Print Access Protocol Specification," Preliminary Draft, European Computer Manufacturers Association, Group TC35-TG5, (September 1988).

[PdTeam88] Handspicker, et al. *The Athena Pd Design Document* MIT Project Athena, (September 1988)

[HRPC88] J. Sanislo, "An RPC/LWP System for Interconnecting Heterogeneous Systems," in *Usenix Conference Proceedings*, (Winter 1988)

[SMS88] M. A. Rosenstein, D. E. Geer, Jr., P. J. Levine, "The Athena Service Management System," in *Usenix Conference Proceedings*, (Winter 1988)

[Kerberos88] J. G. Steiner, C. Neuman, J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," in *Usenix Conference Proceedings*, (Winter 1988)

[Zephyr88] T. DellaFerra, *et al.*, "The Zephyr Notification System," in *Usenix Conference Proceedings*, (Winter 1988)

[Apollo87] T. H. Dineen, P. J. Leach, N. W. Mishkin, J. N. Pato, and G. L. Wyatt, "The Network Computing Architecture and System: An Environment for Developing Distributed Applications," in *Usenix Conference Proceedings*, (June 1987)