

**Strategic Plan for the Creation of an
Integrated Criminal Justice Information System
For Orange County, Florida**

***ICJIS System Architecture
Deliverable #5***

June 14, 2000

Prepared for:



Gale Johnson, Senior Purchasing Agent
Orange County Purchasing Department
Orange County Administration Center, 3rd Floor
21 South Rosalind Avenue
Orlando, FL 32802
Phone: (407) 835-5635

Prepared by:



6360 Interstate 55 North, Suite 300
Jackson, MS 39211
888-664-7400
Fax: 601-957-9492

Rene Claiborne
Computer Sciences Corporation
Account Manager

Ken Slay
Computer Sciences Corporation
Project Manager

Brian Tinsley
Computer Sciences Corporation
Technical Designer

Dr. Tom Wilson
Wilson and Associates
AFIS Consultant



Table of Contents

1.0 Introduction	4
2.0 Architectural Overview	4
2.1 Software.....	4
2.2 Hardware	5
2.2.1 IBM RS/6000 Model F80.....	5
2.2.2 Hewlett-Packard HP9000 N-Class (N4000)	6
2.2.3 Sun Microsystems E4500.....	6
2.2.4 Recommendation.....	7
3.0 Architecture Narrative	7
3.1 Overview	7
3.1.1 Master Person Index (MPI).....	7
3.1.2 Hub Case Management System (Hub CMS).....	8
3.1.3 Software Interfaces.....	8
3.2 Basic Architecture	8
3.2.1 Composition of the MPI.....	8
3.2.2 Collection and Updating of MPI Information.....	9
3.2.3 Updating Agency Systems with MPI Information.....	9
3.2.4 Queries on the MPI	10
3.2.5 Direct Updating of MPI	10
3.2.6 Composition of the Hub CMS.....	10
3.2.7 Collection and Updating of Hub CMS Information.....	11
3.2.8 Direct Updating of Hub CMS.....	11
3.2.9 Updates Between Agencies	11
3.2.10 Synchronization of Hub CMS and Banner Databases.....	11
3.2.11 JIT Statute Table	12
3.2.12 Web Server Software.....	12
3.2.13 Security and Audit Trail Features	12
3.3 Special Processing.....	12
3.3.1 Electronic Filing.....	12
3.3.2 Conviction Packets.....	13
3.3.3 Pre-sentence Investigation Reports.....	13
3.3.4 Court Calendar.....	13
3.3.5 Implementation of Special Processing.....	13
3.4 Event Notifications.....	13
3.5 Interactions between the ICJIS Hub and External Entities	14
3.5.1 Florida Department of Law Enforcement (FDLE).....	14
3.5.2 Florida Department of Corrections (FDOC).....	14
3.5.3 Department of Motor Vehicles (DMV).....	15
3.6 Automated Fingerprinting and Identification System (AFIS).....	15
3.6.1 AFIS.....	15
3.6.2 National Crime Information Center (NCIC 2000)	16
4.0 Network Requirements.....	16



5.0 Security 17

6.0 Configuration Management..... 17

7.0 Data Replication 21

8.0 Hardware and Software Details..... 25

Appendix A - Architecture Diagrams 26

Appendix B - IBM RS/6000 Model F80..... 30

Appendix C - Oracle Application Server 43

1.0 Introduction

This document is the fifth in the series of deliverables for the Orange County Integrated Criminal Justice Information System (ICJIS) Strategic Plan. It describes the general system architecture from the perspective of the necessary hardware and software and includes a narrative describing the system's general operation. In addition, several concepts, such as security, are described both in general terms and how they may apply to the ICJIS architecture. This document does *not* delve into the details of implementing the architecture. That task is left to the detailed design phase of the project.

2.0 Architectural Overview

2.1 Software

After careful evaluation of the available technologies along with Orange County's existing technical infrastructure, it is recommended that the ICJIS system be implemented as a web based 3-tier system. In this architecture, the benefits of traditional client/server systems are realized and many of the shortcomings of traditional client/server systems are overcome.

The first tier, tier 1, is the user interface. Tier 1 is implemented with a standard graphical web browser such as Netscape Navigator or Microsoft Internet Explorer. This makes the system accessible to just about any type of client, from "thin" desktop systems to mainframes, since the only requirement is the ability to run a web browser. Another advantage of implementing tier 1 this way is that the time consuming and tedious process of software distribution that plagues traditional client/server systems (e.g., the "fat client") is virtually eliminated. The second tier, tier 2, is the application server. This is where all the application logic resides, with the possible exception of limited data validation that may be implemented within web pages. It is responsible for servicing client requests and communicating with the database servers. Implementing tier 2 in this way consolidates the application logic usually distributed between the client and server in traditional client/server systems, thus simplifying development and reducing code maintenance costs. The third tier, tier 3, is the database server. Its role is to simply manage the database and service requests from tier 2 systems.

A major benefit of a 3-tier system is scalability. This is very important in a system such as ICJIS since it will likely grow over time with respect to the volume of data and the number of users. In the beginning, both tiers 2 and 3 can reside on a single server. As system demand increases, tiers 2 and 3 may be placed onto separate servers or even distributed across multiple servers within each tier.

Orange County has indicated that Oracle will be the preferred database for ICJIS. This is an excellent choice since Oracle has a first-rate suite of products that support the 3-tier architecture. The *Oracle 8i Enterprise Edition* relational database will be used in tier 3 and *Oracle Application Server* will be used in tier 2. Refer to the **Architecture Narrative** section and **Appendix C** in this document for a description of the features of these products and how they may be used in the ICJIS.



2.2 Hardware

The hardware platform for the ICJIS, in order to effectively support the 3-tier software model, needs to be chosen with regard to several criteria:

- Processor speed
- I/O throughput
- Storage capacity
- Scalability
- Availability features

Systems from three different vendors were chosen for evaluation. The tables below provide a description of how each system meets the above criteria.

2.2.1 IBM RS/6000 Model F80

Criteria	Evaluation
Processor speed	RS64 III 64-bit processor 1, 2, and 4-way SMP @ 450 MHz 6-way SMP @ 500 MHz 128KB data/128KB instruction L1 cache 2MB or 4MB L2 cache per processor
I/O throughput	2.4 GB/sec. memory 2.4 GB/sec. processor 1 GB/sec. I/O (500 MB/sec. each direction)
Storage capacity	254.8 GB Internal 27+ TB External
Scalability	Processors: 1, 2, 4, or 6 Memory: 64 MB min. - 16 GB max. I/O Slots: 10 PCI
Availability features	ECC memory Redundant hot-swap power supplies Redundant hot-swap cooling modules Hot-swap disk bays PCI bus error detection/recovery Dynamic CPU Deallocation System power control network Automatic reboot Separate service processor



2.2.2 Hewlett-Packard HP9000 N-Class (N4000)

Criteria	Evaluation
Processor speed	PA-RISC 8500 64-bit processor PA-RISC 8600 64-bit processor 1 to 8-way SMP @ 360 or 440 MHz (8500) 1 to 8-way SMP @ 550 MHz (8600) 1.5 MB on-chip cache (both models)
I/O throughput	7.6 GB/sec. memory bus 3.8 GB/sec. system bus 5.8 GB/sec. aggregate for PCI slots
Storage capacity	36 GB Internal 71 TB External
Scalability	Processors: 1 to 8 Memory: 512 MB min. – 32 GB max. I/O slots: 12 PCI
Availability features	ECC memory Full parity checking on PCI slots Separate support processor Hot-swap power supplies and fans Hot-plug PCI slots Hot-plug disk bays

2.2.3 Sun Microsystems E4500

Criteria	Evaluation
Processor speed	UltraSPARC II 64-bit processor 1 to 8-way SMP @ 336MHz or 400MHz 16KB data/16KB instruction L1 cache 4MB L2 cache for 336MHz 4MB or 8MB L2 cache for 400MHz
I/O throughput	3.2 GB/sec. system backplane 200 MB/sec. for each SBus board 528 MB/sec. for each PCI board
Storage capacity	72.8 GB Internal 20+ TB External
Scalability	Processors: 1 to 14 Memory: 256 MB min. – 28 GB max. I/O Slots: 2-12 PCI, 1-14 SBus
Availability features	ECC memory Automatic system recovery Dynamic reconfiguration Alternate pathing CPU power control



	Redundant hot-swap power supplies Redundant hot-swap cooling modules Hardware failure prediction Remote power control Modular components
--	--

2.2.4 Recommendation

Each of the above systems would serve the needs of the ICJIS well. Each system excels with respect to the evaluation criteria. However, one vendor’s system beat the other two by a large margin with regard to one factor: price. This vendor is IBM. Based on list prices, the RS/6000 Model F80 was \$90,000 to \$120,000 less than an equivalent HP9000 configuration and \$55,000 to \$80,000 less than an equivalent Sun configuration. Please refer to **Section 8.0** for the configuration used in this comparison (**Appendix B** contains a very detailed description of the system as well). This is also the system and specific configuration recommended for the ICJIS.

3.0 Architecture Narrative

Please refer to the architectural diagrams in **Appendix A** while reading this narrative.

3.1 Overview

The proposed System Architecture for implementation of the ICJIS Hub consists of two new server systems and their associated data stores: the *Master Person Index* (MPI), and the *Hub Case Management System* (Hub CMS). These, together known as the ICJIS Hub, will interconnect all agency systems and will facilitate three major functions necessary for effective information management in the Orange County Criminal Justice System:

- Provide a common store of key information.
- Provide a user interface for querying key information.
- Provide a mechanism for notifying agencies when key information is updated.

The specific functions of the two new server systems are as follows:

3.1.1 Master Person Index (MPI)

- Stores identification information.
- Accesses criminal history information from various sources.
- Contains web server software to implement user interfaces for both regular users and administrators.
- Contains custom software interfaces to facilitate communication with agency systems.
- Contains security and audit trail features to regulate and track activity.



3.1.2 Hub Case Management System (Hub CMS)

- Stores a subset of the Clerk of Court *Banner* database.
- Contains web server software to implement user interfaces for both regular users and administrators.
- Contains the table of statutes
- Contains the court calendar
- Contains custom software interfaces to facilitate communication with agency systems.
- Contains security and audit trail features to regulate and track activity.

3.1.3 Software Interfaces

A key component in each of the above systems is the set of custom software interfaces. These interfaces will facilitate data exchange between the agency systems and the hub as well as data exchange between the different agency systems. These interfaces perform the following functions:

- Updates the MPI or Hub CMS database based on new or updated information in agency systems.
- Notifies users when there is up-to-date information on the MPI or Hub CMS databases when they perform a query.
- Facilitates the update of agency databases with up-to-date MPI or Hub CMS information after applying business rules.
- Performs any necessary data translations such as reconciliation of field lengths and codes.

For each feature mentioned in this architecture narrative, there will likely be a number of viable options for implementation. Where possible, some implementation options are listed. However, the specific solutions will need to be determined during the detailed design phase. During detailed design, many factors must be considered and evaluated before the optimum solution can be identified. Factors likely to be considered include the logistics of inserting new code into existing agency applications, the structure of the various agency databases, and the magnitude of business rules which must be coded in the custom interfaces.

3.2 Basic Architecture

3.2.1 Composition of the MPI

The primary function of the MPI is to provide a common data store for an individual's identification information and criminal background information. The MPI consists of the following:

- Oracle 8i Enterprise Edition RDBMS
- Oracle Application Server 4.x
- Custom Software Interfaces.

The Oracle RDBMS will manage the MPI database. The Oracle Application Server (OAS) will facilitate access to the database via its web server, the custom software interfaces, and any other external applications. The OAS web server will provide the primary user interface to the MPI through web pages. It may employ web-based technologies such as JavaScript, Dynamic HTML, Java, and XML to access the MPI database. This will be decided in the detailed design phase. There will be a dedicated custom software interface in the MPI for each agency system. These interfaces will provide the mechanisms for sharing and updating information between agencies and the MPI. Each interface will be custom software and could be written in a language such as C++, Java, or Oracle's PL-SQL, each of which is capable of utilizing the features of the OAS to access the MPI database. Refer to **Appendix C** for an overview of the OAS.

3.2.2 Collection and Updating of MPI Information

Identification information will be collected from agency systems using the custom software interfaces resident on the MPI. These interfaces, one for each agency database, will detect new or changed data in the agency databases and will add or update records in the MPI database accordingly. There are a number of options for implementing this feature, including the regular polling of agency databases by the MPI interfaces, implementing a tool on agency databases to update the MPI or insertion of code in agency applications that transfer new and updated data to the MPI.

Criminal background information will be accessible via the MPI database and will be collected from agency systems and from the *Florida Department of Law Enforcement* (FDLE). Criminal background information may be resident on the MPI database, queried from FDLE through the MPI, or a combination of both. The MPI will facilitate queries of criminal background information by the appropriate agencies and courts.

3.2.3 Updating Agency Systems with MPI Information

The custom software interfaces in the MPI will ensure that users of agency systems have the opportunity to update their systems with the latest MPI information. When a user of an agency system queries MPI data, a mechanism will be employed to inform the user that there is information on the MPI that is newer than the same data on the agency system. This mechanism may require insertion of a trigger in each agency system, either in the application software, I/O routine, or at the database level. The trigger will initiate the custom interface program on the MPI, which will then provide the appropriate information to the user. The user can then choose to update the agency database with the MPI information. When agency information is updated with MPI information, any required business rules will be enforced by the appropriate custom software interface.

Alternatively, agencies may receive notification of MPI changes in real-time or at regular intervals and would then be responsible for making the necessary updates to the agency systems themselves.

3.2.4 Queries on the MPI

The MPI will contain web server software, a part of the Oracle Application Server, which provides connectivity to the various databases for query purposes. This software will allow a user to log in via a web browser and select specific sets of information from a list of queries and reports. The web server will utilize the MPI and Hub CMS databases as well as agency and external databases where necessary to compile the desired information for each query and report.

3.2.5 Direct Updating of MPI

There will likely be the need for direct-update capability on the MPI by certain authorized users. This will allow data to be inserted or changed manually if the need arises. Special web pages will be provided for this purpose.

3.2.6 Composition of the Hub CMS

The Hub CMS provides the primary communications link between the agencies and the new Clerk of Court Case Management System, named *Banner*. It will provide the primary data store of Clerk of Court CMS information by maintaining a database consisting of the subset of Clerk of Court data required for exchange between agencies. It will also contain copies of the JIT statute table and court calendar. The Hub CMS consists of the following:

- Oracle 8i Enterprise Edition RDBMS
- Oracle Application Server 4.x
- Custom Agency Software Interfaces
- CMS Update Utilities

The Oracle RDBMS will manage the Hub CMS database. The Oracle Application Server (OAS) will facilitate access to the database via its web server, the custom software interfaces, and any other external applications. The OAS web server will provide the primary user interface to the Hub CMS through web pages. It may employ web-based technologies such as JavaScript, Dynamic HTML, Java, and XML to access the MPI database. This will be decided in the detailed design phase. There will be a dedicated custom software interface in the Hub CMS for each agency system. These interfaces will provide the mechanisms for sharing and updating information between agencies and the Hub CMS. Each interface will be custom software and could be written in a language such as C++, Java, or Oracle's PL-SQL, each of which is capable of utilizing the features of the OAS to access the Hub CMS database. The CMS Update Utilities will be special

programs used to synchronize the Hub CMS database with the Banner system. Refer to **Appendix C** for an overview of the OAS.

3.2.7 Collection and Updating of Hub CMS Information

As with the MPI, the Hub CMS database will be populated and maintained by utilizing custom software interfaces, which perform the appropriate record generation and updating on the Hub CMS. Basic interaction between the Hub CMS interfaces and the agency systems will follow the same strategies as outlined in paragraphs 3.2.2 and 3.2.3 of the MPI section above.

3.2.8 Direct Updating of Hub CMS

There will likely be the need for direct-update capability on the Hub CMS by certain authorized users. This will allow data to be inserted or changed manually if the need arises. Special web pages will be provided for this purpose.

3.2.9 Updates Between Agencies

In some instances, the custom software interfaces must detect and pass on updates of information between agency databases. This will require the interfaces to be linked such that the originating agency interface will pass on the required data to allow the receiving agency interface to update its database accordingly. Where possible, this updating function would be performed by the application software. If the custom interface software performs any updates, it must apply any business rules resident in the receiving agency application software. In instances where business rules are contravened, notification to users of the receiving agency must be notified. There are a number of possible strategies for performing this notification, such as email, generation of a report, or notification to users as part of the logon process.

3.2.10 Synchronization of Hub CMS and Banner Databases

The Clerk of Court Banner system will provide updates to the Hub CMS at predetermined intervals, generated by a utility associated with the Banner system. These intervals will likely be small, since there is a requirement for very timely information on the Hub CMS. Likewise, the Banner database will be updated at predetermined intervals with Hub CMS information by utilizing a similar utility. These utilities will include security features to ensure information integrity. The specific security functions will need to be defined during the detailed design phase. If the Hub CMS is administered by Clerk of Court Information Services personnel, they will also be responsible for the update utilities.

3.2.11 JIT Statute Table

The Hub CMS will store a copy of statute information in the JIT statute table. The primary statute table will reside in the Clerk of Court CMS and will be maintained by the State Attorney's Office. The Hub CMS statute table will be accessible by all users of the ICJIS Hub. The Hub CMS will notify agencies of statute table changes so that they can update their local systems themselves. It would be possible for agency statute tables to be updated automatically when additions or changes are made to the primary statute table. If this were a required function, specifications would need to be developed during the detailed design phase.

3.2.12 Web Server Software

The Hub CMS will host web server software, a part of the Oracle Application Server, which provides connectivity to the various databases for query purposes. This software will provide redundancy for the MPI web server, and will facilitate quicker access times. For example, queries for identification and criminal history information may be handled entirely by the MPI web server. The Hub CMS web server will likewise utilize the MPI and Hub CMS databases as well as agency and external databases where necessary to display the desired information.

3.2.13 Security and Audit Trail Features

The Hub CMS will contain security and audit trail features to regulate activity between the Hub CMS database and the Clerk of Court Banner database. This is important since there will be a large number of users with the potential to update Clerk of Court data, doing so indirectly via the ICJIS Hub. This security and audit trail mechanism will ensure that only authorized users and processes are capable of accessing the Banner system and should specify exactly what data can be accessed on the Banner system. An audit log will be written on the Hub CMS system, and possibly on the Banner system, each time an update of the Banner database is initiated by the Hub CMS system. The format of the audit log will be decided during the detailed design phase, but should contain the expected fields such as user/process name, date, time, and a description of what was updated.

3.3 Special Processing

In addition to the basic interactions and data exchange between the ICJIS Hub and agency systems discussed above, there are several other functions that can be facilitated by the proposed architecture.

3.3.1 Electronic Filing

The Hub CMS component of the ICJIS Hub could allow for electronic filing by utilization of a browser based application that allows the originator to supply the required

information and electronic signature to the receiving agency. Alternatively, electronic filing could be initiated by an agency system and passed to the Hub CMS application where additional information could be automatically added before sending the file on to the receiving agency.

3.3.2 Conviction Packets

The Hub CMS component of the ICJIS Hub can also allow for processing the conviction packet prepared by the Clerk of Court and sent to the County Corrections system. As with Electronic Filing, this could be accomplished by filling out an electronic form resident in the Hub CMS and accessed by a web browser, or the transmission of a data set from the Banner system to the County Corrections system. Once the Jail Management System is updated with the inmate classification and location, information can be sent to the FDLE and Probation & Parole (P&P) via the ICJIS Hub.

3.3.3 Pre-sentence Investigation Reports

The ICJIS Hub will provide for electronic requests by the Court for Pre-sentence Investigation (PSI) reports. These requests will be transmitted to the P&P office and downloaded into the P&P Case Management System. The PSI report template in the P&P Case Management System will capture demographic data and other pertinent information, then utilize the ICJIS Hub to search the Hub CMS, MPI, FDLE, and other databases necessary to incorporate pertinent information into the PSI report template. When completed, the PSI Report can be electronically submitted to the requesting Court. Again, custom software may be necessary in the Hub CMS to facilitate automated PSI report processing.

3.3.4 Court Calendar

The Court Calendar can be maintained on the Hub CMS and be updated directly by a system owner, such as Court Administration. If an automated feature were required for maintaining the Court Calendar, this function would need to be specified as part of the agency-specific interfaces.

3.3.5 Implementation of Special Processing

Custom software resident in the Hub CMS would be required to implement many of the above features. These functions could be included in the agency-specific interfaces or be part of a separate custom software module designed to process these specific functions.

3.4 Event Notifications

At certain points during processing on the ICJIS Hub, specific agencies need to be notified that an event has occurred. For example, when an arraignment date is set by the Clerk Of Court, notification of this event must be sent to County and/or State

Corrections, the SA/PD office, and the Sheriff and/or Orlando Police Departments. The mechanism used to deliver the notification could be as simple as an email dynamically generated by the Hub and sent to a user or users at the appropriate agencies. Alternatively, a more robust system could be developed using a technology such as Java that supplies a programming interface to a network-enabled messaging system (i.e., the Java Messaging API, which allows the synchronous exchange of critical business data and events between Java objects residing on the same system or different systems). A system built this way would allow further automated processing of the notification beyond just a person reading an email. For example, an entry could automatically be made in a police officer's electronic calendar on receipt of the notification specifying that a court appearance is required on a specific day at a specific time. Java objects would have to be written and placed on both the Hub and on agency systems to enable communication of the event notifications. This would not be a problem since Java runs on nearly every platform from the PC to IBM mainframes.

3.5 Interactions between the ICJIS Hub and External Entities

In addition to the tight coupling of the ICJIS Hub and agency systems, there are a number of systems and entities external to Orange County with which the ICJIS Hub must interact.

3.5.1 Florida Department of Law Enforcement (FDLE)

The ICJIS Hub must communicate with the FDLE databases containing criminal history in order to provide up-to-date criminal history information via the MPI. Connectivity to FDLE is through a wide area network utilizing frame relay. Depending on a number of factors, such as response times, connection reliability, and permissions granted by FDLE, this information could be downloaded to the ICJIS hub periodically, or be accessed in real-time as necessary.

3.5.2 Florida Department of Corrections (FDOC)

The FDOC currently interchanges information with Orange County agencies in paper format. Where necessary, information must then be keyed into the appropriate systems. This process could be automated by establishing a link to the FDOC and establishing a mechanism for submittal of information relating to charging affidavits, arrest warrants, final dispositions, sentence score sheets, and case history. For information submitted from Orange County to FDOC, automation can be accomplished by utilizing the messaging functionality described in Section 3.4. Submission of pre-sentence score sheets and case history information could be accomplished by providing a web browser user interface on the Hub CMS to allow FDOC personnel to directly key in the desired information, or alternatively, implementing software locally on the FDOC system to automatically transmit the pre-sentence score sheet to the Clerk of Court via the ICJIS Hub. Trial date information could be accessed by FDOC by viewing the Court Calendar resident in the Hub CMS via a web browser.

3.5.3 Department of Motor Vehicles (DMV)

The DMV can be linked to the ICJIS Hub allowing information contained therein to be accessible to all ICJIS users. The requirements for DMV information and logistics of data transfer will need to be specified later.

3.6 Automated Fingerprinting and Identification System (AFIS)

A key component of the proposed system architecture is the automated fingerprint processing. There are a number of options for implementing AFIS technology, and the cost and effort associated with implementation will vary greatly depending upon the specific features desired.

3.6.1 AFIS

The AFIS architecture for the Orange County ICJIS, as depicted in Figure 3, consists of hardware, software, and telecommunications necessary to support AFIS identification functions at federal, state, and local levels.

For arrest/booking and forensic latent crime scene identification functions, the AFIS architecture shown utilizes remote terminal access through the host AFIS of the Florida Department of Law Enforcement (FDLE). For the arrest/booking process, a remote live scan terminal at Orange County Corrections Department (OCCD) captures the fingerprint images and transmits them to the remote AFIS Identification Terminal at the Orange County Sheriff's Office (OCSO). Forensic latent work is conducted at the OCSO. A NIST Archive stores a full set of fingerprint images, as well as demographic and arrest data, for each offender arrested or registered in Orange County. The architecture depicted for arrest/booking and latent identification functions represents the functionality that Orange County is in the process of implementing.

The AFIS architecture also depicts the functionality for a future phase of AFIS implementation. Specifically, this future phase envisions a local two finger database and 10 finger database that will allow Orange County to provide identification and verification of identity functions for the courts, for inmate movement and release, and for law enforcement investigation of suspects in the field. At this level of functionality, the AFIS architecture will need an enterprise server to execute the business rules for integrated records and identification transactions, as depicted in the AFIS architecture diagram.

The AFIS architecture also shows the logical relationship between the Orange County identification functions and state and federal systems. FDLE serves not only as the AFIS database and search engine for Orange County's arrest/booking and forensic latent crime scene identification processing, but also serves as the conduit to the Federal Bureau of Investigation's (FBI) AFIS processing. The FBI's Integrated Automated Fingerprint



Identification System (IAFIS) is a repository of the nation's AFIS database. All Florida felony and serious misdemeanor AFIS records are indexed in IAFIS.

NCIC 2000 is the FBI's National Crime Information Center's database of wanted and missing persons, stolen property, firearms, etc. NCIC includes a fingerprint search component. The AFIS Architecture diagram shows the connectivity between FDLE, the FBI, and Orange County for wanted persons searches through the FDLE AFIS.

3.6.2 National Crime Information Center (NCIC 2000)

NCIC 2000 is the FBI's National Crime Information Center's database of wanted and missing persons, stolen property, firearms, etc. NCIC includes a fingerprint search component that can be used by local and state law enforcement to execute positive identifications through Automated Fingerprint Identification Systems (AFIS) in the field. State and local agencies conduct automated name searches of the NCIC "Hot Files" through a dedicated NCIC terminal. Each state maintains an official NCIC Control Terminal Officer.

As part of the Orange County ICJIS enterprise, NCIC 2000 will be an integral part of the AFIS Architecture. The Operational Scenarios developed for this strategic plan call for conducting queries on the NCIC terminal at critical junctures in the identification process. It is recommended that future functionality include automated searches of the NCIC wanted persons file following detection of an alias by the AFIS.

4.0 Network Requirements

Based on the information gathered as part of *Deliverable #2: Review and Assessment*, Orange County already has in place a network infrastructure capable of supporting ICJIS. All agencies have connectivity to one another via the County's high-speed backbone. All relevant agencies external to Orange County are also connected via wide area network links.

The two components that comprise the ICJIS Hub should each be connected to the hosting agency's network with the minimum of a switched, 100-megabit, full duplex Ethernet connection. The connection should be located as "close" as possible to the County backbone in order to minimize latency and reduce the impact on the hosting agency's own network.

All communication with the ICJIS Hub should be standardized on the TCP/IP protocol suite. These network protocols are very robust and flexible and are widely supported by vendors of systems of all types. In addition, most modern network hardware offers features that allow administrators to optimize and closely monitor the operation of these protocols.

5.0 Security

The threat to computer systems is rapidly evolving as systems become more interconnected and as the power and utility of those connections increases. On the surface it appears that security is a simple concept - protect the computer systems. However, it is really much more complex. A security system should provide protection from people outside the organization, as well as from unauthorized or fraudulent use by people inside the organization. It should provide security from physical access to the computers, but also security from remote access through intranets and the Internet. It should also guarantee that unauthorized people can't access functions of the system that are to be secured, such as assigning a court date to a case, even if they are supposed to have access to other aspects of the system. Thus, the idea of security becomes somewhat larger and more complex than simply protecting the computer. In the long run, the only hope for securing access to connected systems is to implement strong authentication technologies, such as those provided by Kerberos, public key systems, and security tokens; integrating them into end systems and applications. Since the ICJIS Hub allows inter-agency data access, both query and update functions, a multitude of security issues arise. Who has access to what data on what systems? Do certain people within a particular agency have access to more data than other people within the same agency? How can a user authentication scheme be developed that covers the ICJIS Hub as well as all other internal and external systems? These issues, and many others, must be defined and addressed as the system design progresses into and through the detailed design.

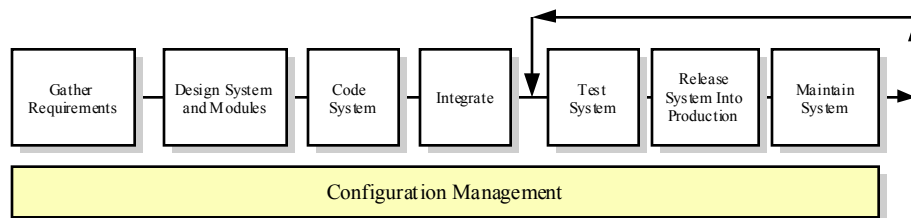
6.0 Configuration Management

Historically, the development process has varied greatly from project to project and system to system. Each project team made their own rules, selected their own tools, and followed their own processes. Unfortunately, this created a number of problems, including:

- Inconsistent quality of systems
- Expectations within the development and user communities were not always aligned
- Increased learning curve as new project teams were formed
- Increased development, maintenance effort, and timeframes

With the ever increasing pressures to develop systems faster, cheaper, and with more functionality and more complexity than ever before, a common set of development tools and processes are an absolute necessity.

Development includes all of the tasks and activities which are involved in gathering requirements for a system, designing the system, coding the modules in the system, integrating the system with the infrastructure or other modules and systems, testing the system, putting the system into production, and maintaining the system throughout its life.

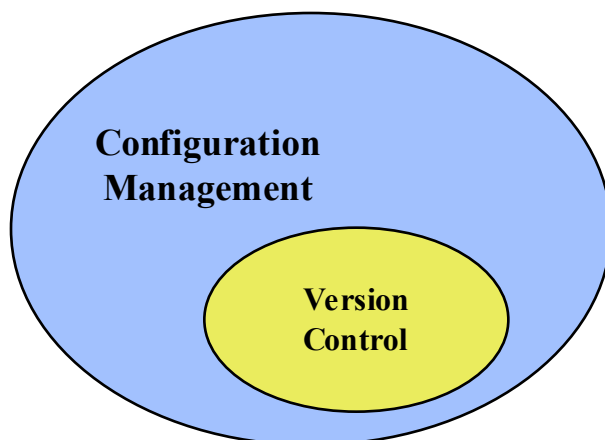


Configuration management and version control spans the entire life of the system, as information is created, used, modified, and stored at each step in the process. It is important that all of this information be captured and stored so that when it is needed later it can be readily accessed. Configuration management and version control refers to the processes and procedures associated with creating, using, modifying, and storing this information as a system moves through its life cycle. Some of the types of information that should be stored include:

- Application Requirements
- System Design Documents
- Program Design Packages
- Testing Plans
- Test Data
- Expected Results
- Old Versions of Everything
- Operator Instructions
- Online Help
- Training Manuals
- Training Data
- Source Code
- Configuration/Profile Files
- Software & Libraries

It is important to note that in addition to capturing this sort of information for each application system, the architecture as a whole also needs to be treated as a system and be subject to the same sort of processes and controls.

Configuration Management (CM) and Version Control (VC) are terms that are often used interchangeably. To some extent they are the same, in that both refer to the control of versions of files, usually source code in the case of version control. However, configuration management usually refers to something bigger than just controlling the versions of source code. Configuration management usually refers to controlling entire systems and sub-systems of files, not just the source code. Also, configuration management refers to managing the process, including having controls in place for checking in and checking out files of different types in different environments. Thus, configuration management is a superset of version control.



The problem which configuration management is supposed to address, more so than any other, is facilitation of the production of a high quality system in the least amount of time. This implies:

- Facilitating reuse of components
- Providing easy to use, high productivity tools
- Supporting the migration of systems or sub-systems from one environment to another (such as from development to a user test area, from the user test area to a training environment, etc.)
- Controlling or ensuring the quality of code
- Addressing conflicts, such as two developers changing the same module
- Providing a mechanism for automated building of environments or systems

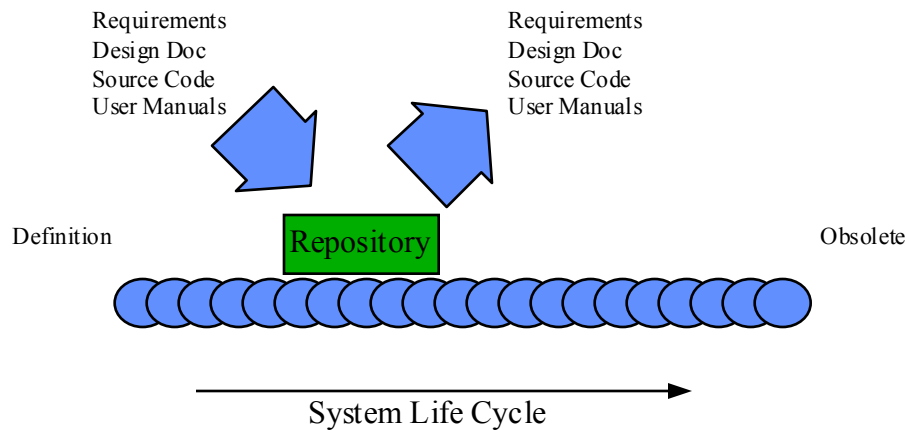
In order to provide these capabilities a number of technologies tend to be needed. They include:

- Workflow software
- Repository or vault software
- GUI environment

Sometimes these technologies are available within a single environment and software system and sometimes it requires the integration of several packages to provide the technologies and functionality required.

The configuration management process can be looked at as a conveyor belt. A system is defined at one end of the belt and is obsolete at the other end of the belt. Everything we know about the system should be put in or taken out of the repository as the system moves through its life cycle (i.e., the belt). Of course, capturing this knowledge in a repository requires rules for what can be stored, who is allowed to store it, who can access it, and what should be done with it. Again, very much like a conveyor in a factory

where specific workers will do specific tasks in a predefined sequence and only certain workers can do certain activities.



The factory paradigm doesn't end there. In a configuration management process there is a map of what the appropriate process for a specific product is, similar to a bill of materials in a manufacturing plant. What needs to go in the repository, the order needs to be completed in, who is allowed to do the work, and even who can approve completed work needs to be defined. Also, like an engineering drawing, how the various pieces fit together needs to exist.

The vision for development of the Orange County ICJIS is to eventually have a single repository that can be accessed from all of the strategic platforms/environments within Orange County and in which all of the documents related to a system will reside. All of the documents, as they are created or modified will be placed in the repository. Automated workflow will be used to gain approval for moving subsystems or systems between environments, such as from test to production. The repository will be able to be used to access the actual documentation, source code, etc., but will also have searching capabilities so that if someone wants to determine the impact of a change, they can do so easily and quickly. One aspect of the "system" which will not be stored in this single repository is the data. Data is treated as a separate entity that the architecture and applications share.

As part of designing and implementing this new vision at Orange County, a variety of processes will need to be implemented. While many of these processes may exist today, and some may even be automated, in the future it will be necessary to have all of the processes be as streamlined and as automated as possible. This should maximize everyone's efficiency and lead to higher quality, more quickly developed systems with fewer errors being made in the migration process as information flows through the development and test processes into production.

The major processes that will be developed include:

- Check-in/out of a source file
- Approving migration of a release (workflow, electronic signature, etc.)



- Migrating a release from one environment (such as development) to another (such as test)
- Making/building a release
- Distributing a release into the appropriate environments

These processes will be developed and documented as part of the detailed design and system rollout.

7.0 Data Replication

The purpose of this section is to define the standard mechanism in which data replication can be accomplished within the ICJIS architecture. An example of using data replication in ICJIS is updating the Hub CMS with data from the Clerk of Court Banner system.

There may also be instances where it can be used with the MPI system, such as automatically updating an agency system with MPI data. For the purpose of this section, replication means the copying or synchronization of data contained in an Oracle database across two or more physical systems. It specifically does not include mirroring of data within a single physical system, which is an operational issue. Any application designer, programmer, or maintainer needs to be familiar with and follow this standard for replication. Exceptions to these standards need to be approved by the appropriate group.

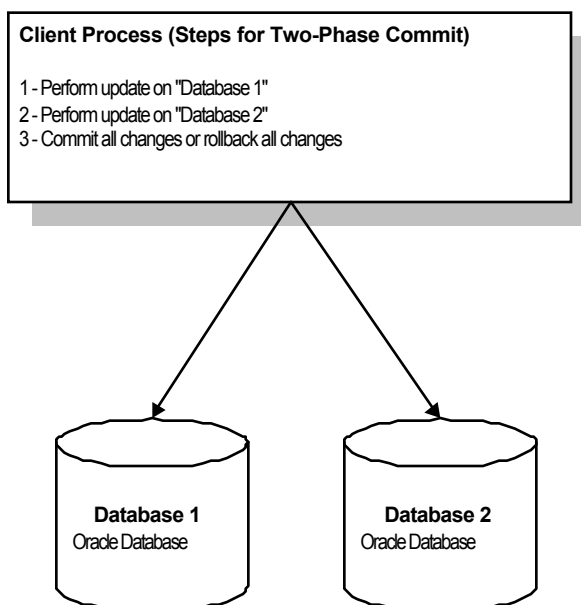
There are many different ways to accomplish data replication across one or more physical database systems. However, none of them are particularly easy to implement. Existing “Off-the-Shelf” products are not currently capable of replicating large numbers of transactions quickly and efficiently.

The only other solution to providing advanced data replication is to build a custom solution. This will enable Orange County to also provide “Smart” replication. “Smart” replication will enable Data Dependent Routing (DDR) techniques to be utilized. This means that data may be replicated to particular databases based on the values of the data. This “Smart” replication is not currently found in off-the-shelf products. The particular method of data replication to use is chosen by answering the following questions:

- How fast does the replication need to occur?
- How much data needs to be replicated?
- How failure of replication needs to be addressed?

In the case of ICJIS, three primary alternatives are feasible: the two-phase commit, Oracle Advanced Replication, and custom triggers.

Two phase commit allows a program to insert, delete, or update data on separate databases on separate physical computer systems within a transaction. Oracle supports two phase commit logic within the SQL used with Oracle v8.x. This control language should be placed in stored procedures, not embedded in application code. A two-phase commit can support two or more physical systems, so this can be extended to transactions that span more than two physical servers. When a failure occurs, the transaction fails and all updates, deletes, or insertions that may have already occurred are rolled back. In other words, either the entire transaction succeeds or it fails. An example of how this would work is depicted below.



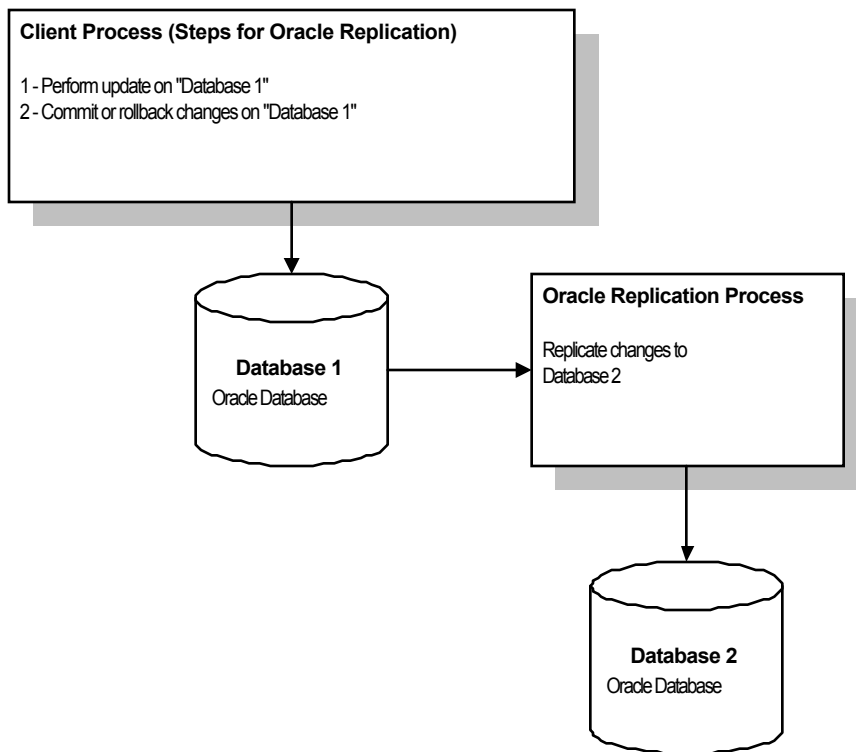
Pros

- Guarantees after the commit that all changes were committed in all locations
- If any part of the transaction fails, all changes for the transaction are rolled back, preventing databases from becoming out of sync with each other

Cons

- In many cases, if failures preclude a transaction from being completed, the entire transaction must be rebuilt, making it impossible for the application to continue normally

Oracle Advanced Replication is the Oracle product/feature that can provide replication of data across systems. The databases can be configured for frequent or infrequent replication intervals that meet the business needs of ICJIS.



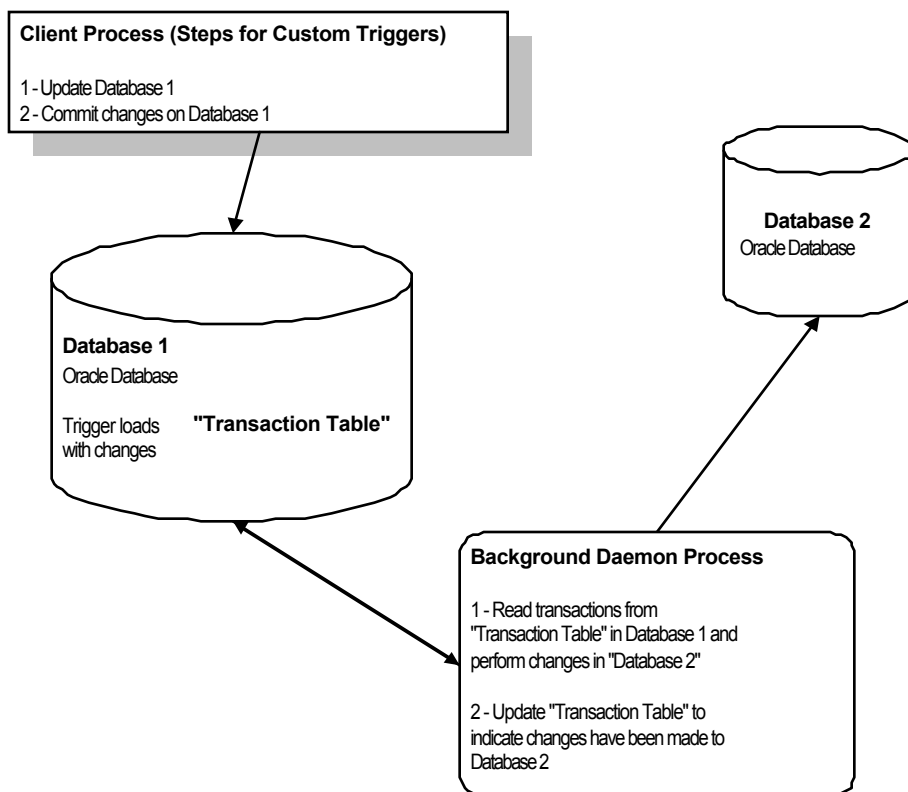
Pros

- Guarantees synchronization across databases after failures have occurred
- Off-The-Shelf package makes it attractive

Cons

- Large amounts of overhead cause network problems

A trigger-based solution would allow any updates, as well as inserts and deletes, to be replicated to another server. When the triggering event occurs, the SQL statement that triggered the event would be inserted into a table on the local server. A daemon (background) process would pick up these SQL transactions and then execute them on the remote server.



Pros

- This approach has been used in the past and has been proven to work
- Daemon processes may be “Smart” to enable data dependent routing
- Daemon processes may be configured to process large or small volumes

Cons

- Additional code must be maintained for both daemon processes and Oracle database triggers
- Additional overhead is required for both CPU, memory, and network bandwidth

8.0 Hardware and Software Details

The following hardware and systems software are likely to be required in order to implement the ICJIS architecture outlined above. These requirements will need refinement during the detailed design phase.

MPI

IBM RS/6000 Model F80

4-way SMP - 450MHz RS64 III processors with 4MB L2 cache per processor

2GB memory (4 512MB DIMMs)

SSA disk bay backplane

SSA RAID adapter

Dual boot-bay option

2 9.1GB Ultrastar 9ZX SCSI hard disks (mirrored boot/system disks)

4 18.2GB Ultrastar 18ZX SSA hard disks (RAID data disks)

POWER GXT130P graphics adapter

P72 color monitor

Keyboard/Mouse

AIX 4.3.3 Operating System

Oracle 8i Enterprise Edition RDBMS

Oracle Application Server

Hub CMS

The Hub CMS hardware and software should be nearly identical to the MPI system specified above. However, since the Hub CMS is not scheduled for implementation until 2003, models and versions will likely change. The system will have to be re-specified at that time.



Appendix A

Architecture Diagrams

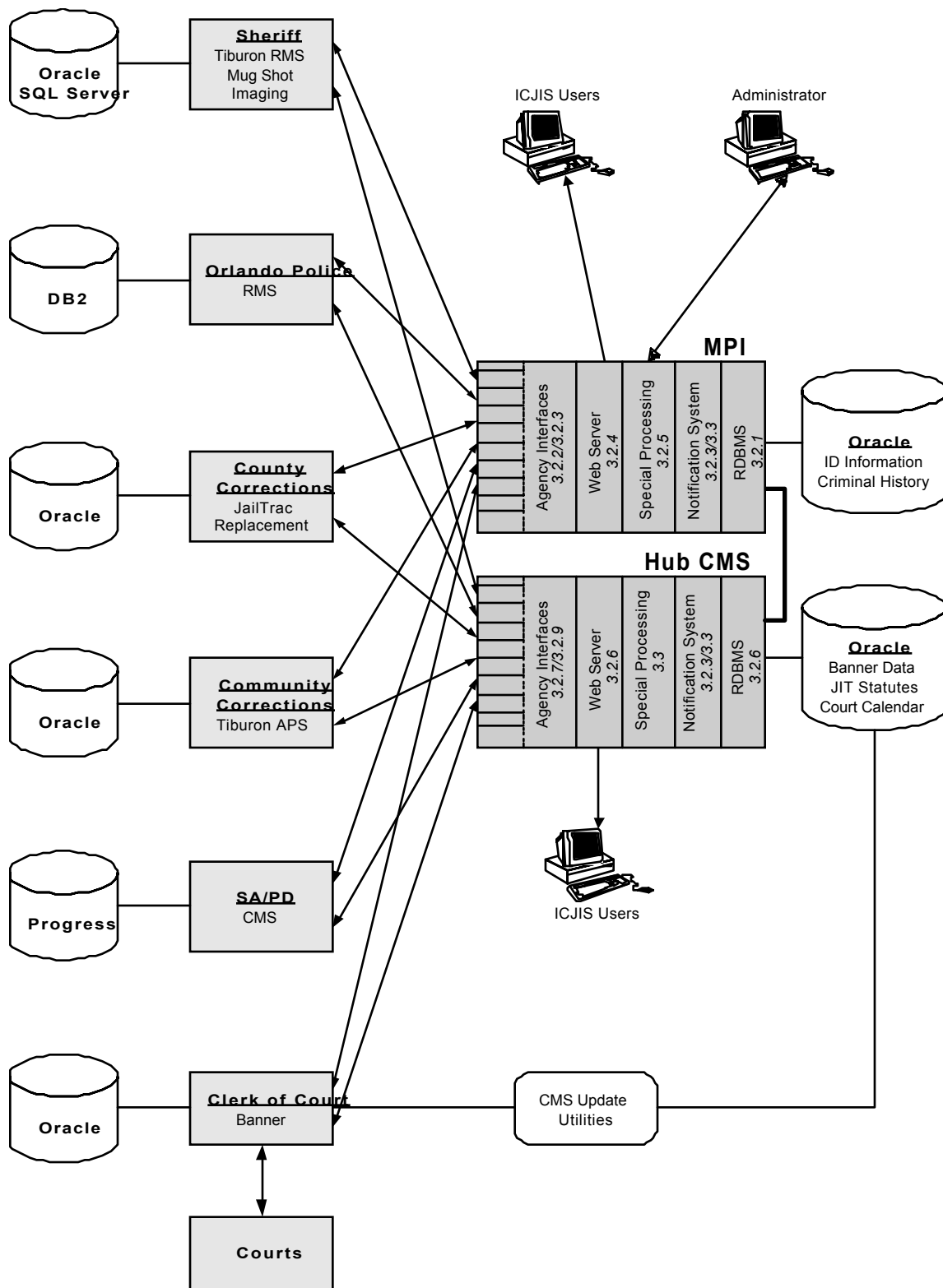


Figure 1 – General System Structure – Internal Agencies

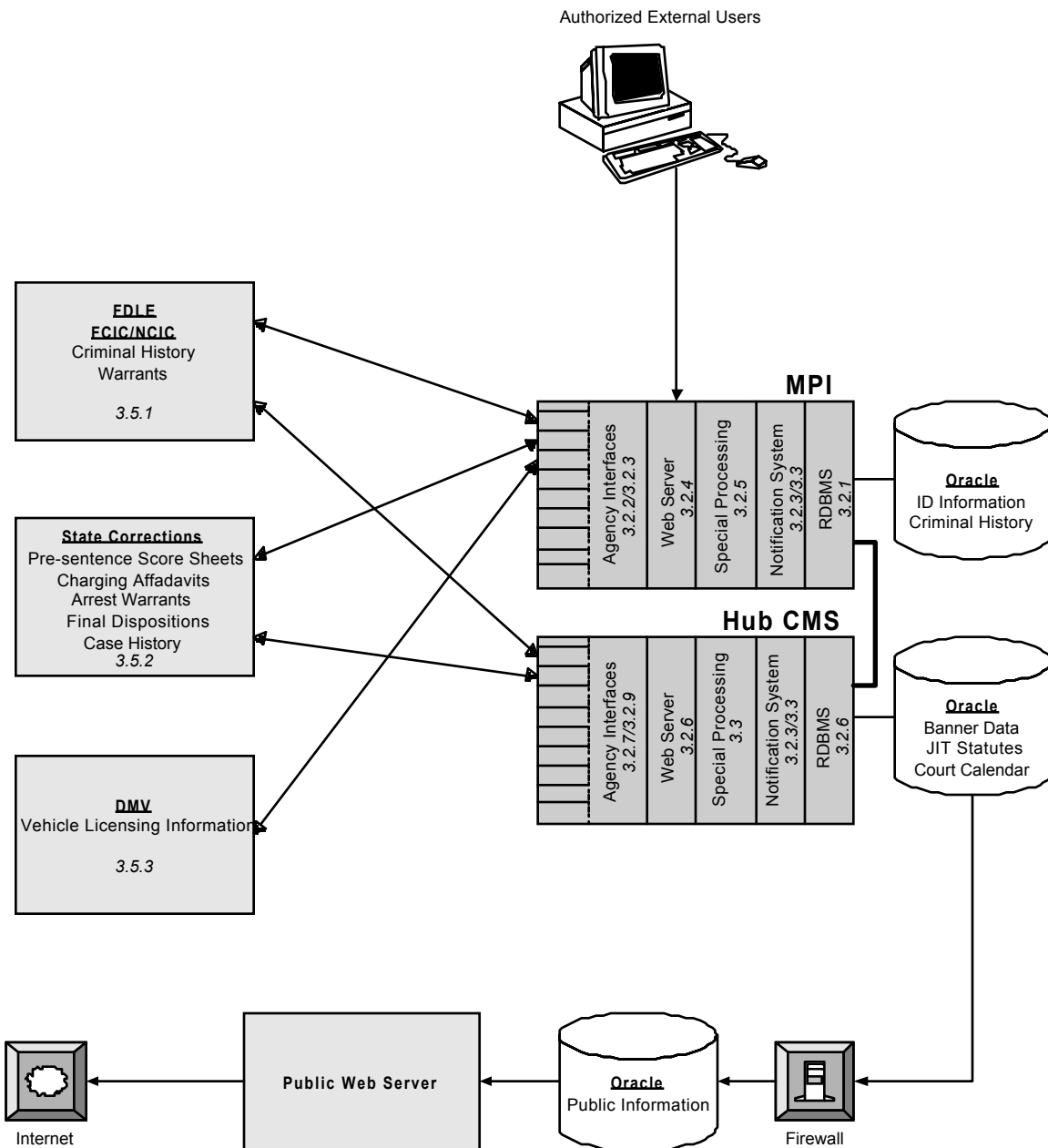


Figure 2 – General System Structure – External Agencies

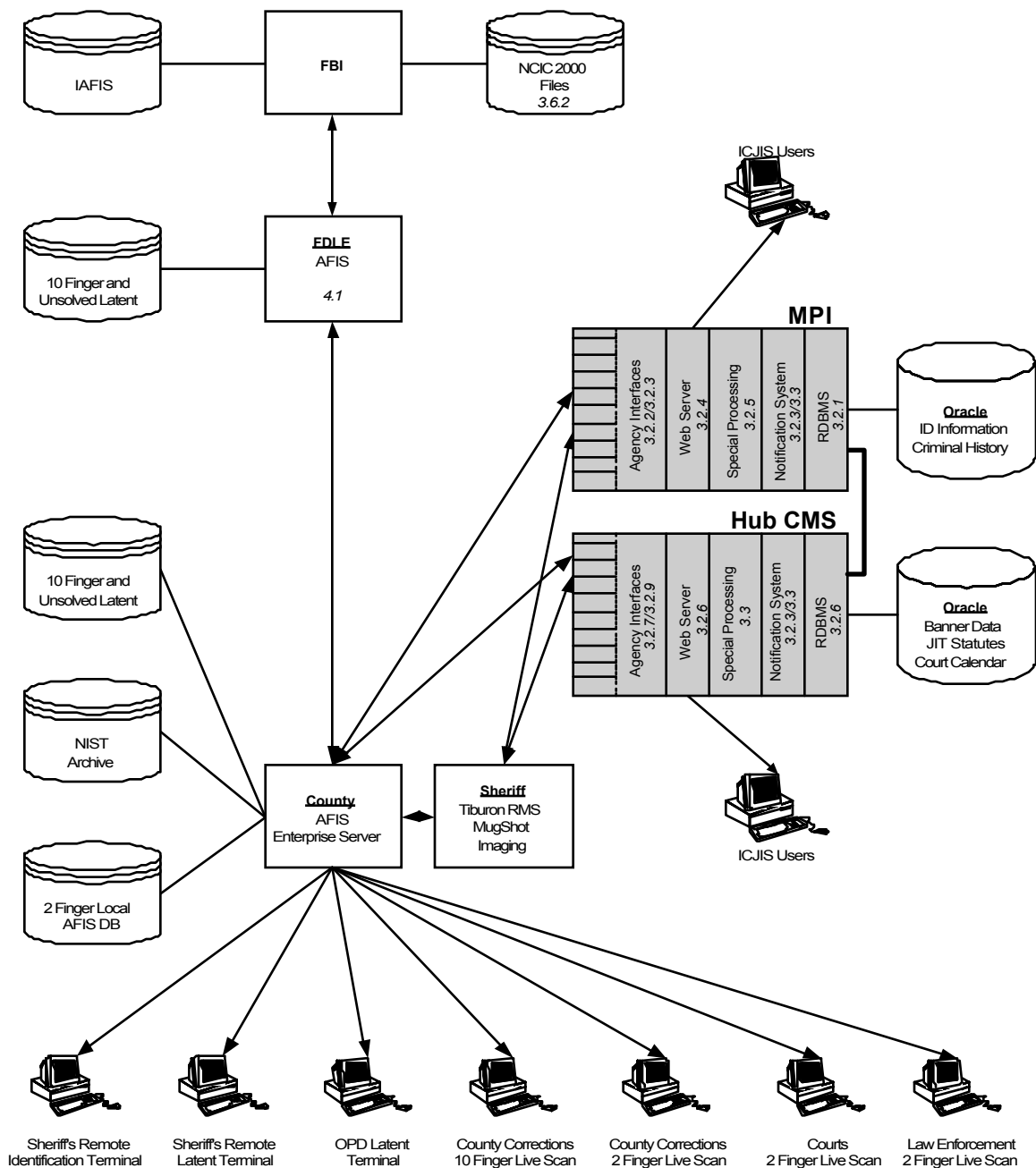


Figure 3 – General System Architecture – AFIS



Appendix B

IBM RS/6000 Model F80

IBM RS/6000 Model F80 Technical Overview and Introduction

Copyright © IBM 2000

IBM achieved leadership performance in the high-end SMP UNIX server marketplace through the balanced design of the 24-way RS/6000 7017 Model S80, which ranked top in the industry standard TPC-C benchmark at 135,815. On May 9, 2000, IBM introduced the RS/6000 7025-F80, 7026-H80, and 7026-M80 to enhance the mid-range server lineup with products using many of the design elements that led to the success of the high-end Model S80.

Overview

While the IBM RS/6000 Models F80 and H80 are the systems that provide a growth path for existing installations of Model F50s and H70s, the Model M80 is designed to provide leadership performance among the mid-range 8-way systems. The target for performance improvement over the existing mid-range Model F50 is over three times with the Model F80 and H80, and over five times with the Model M80 server. In addition to providing CPU and I/O expandability, the Model F80 combined with the latest storage technology provides the maximum internal storage capability available among the current line of RS/6000 mid-range servers. This paper discusses, in detail, the processor, memory, I/O, expandability, reliability, and other technical aspects related to the Model F80.

The IBM RS/6000 Model F80 is a member of the 64-bit family of symmetric multiprocessing (SMP) servers from IBM. The Model F80 is a 64-bit deskside system, which can be configured as a 1-, 2-, 4-, or 6-way SMP with up to 16 GB of real memory. The Model F80 offers flexibility regarding the number of CPUs, memory DIMMs, PCI adapters, and disk drives desired for a specific application or usage.

Physical Package

The Model F80 is packaged in a rugged black deskside steel chassis. The Model F80 server includes a modular hot-swap disk subsystem that allows fast, easy addition and replacement of drives. It has a maximum internal storage capacity of 254.8 GB (218.4 GB hot-swappable) and the possibility to double this as new storage technologies are made available. The Model F80 has a flexible I/O subsystem including ten 64-bit hot-plug PCI slots. It is shipped with the internal adapters and devices installed and configured; software can also be preinstalled if desired.

The Model F80 is designed to operate in a typical office environment with standard AC power at 100-127 volts or 200-240 volts. The Model F80 systems are built with two standard internal hot-swappable power supplies that combine to provide ample power for any configuration. An optional third internal hot-swappable power supply with two hot-pluggable fans can be ordered to provide redundant power and cooling, allowing the system to continue running in the event of a failed fan or power supply. In the event of one of the fans failing, the other fans increase their speed to provide sufficient cooling. The hot-pluggable fans are performance monitored by the System Power Control



Network (see description below). The hot-swappable power supplies and fans can be replaced concurrently if the optional redundant power is installed.

The dimensions of the Model F80 are 483 mm W x 728 mm D x 610 mm H (19.0” W x 28.7” D x 24.0” H). The weight is from 70 kg (155 lbs) to 95 kg (209 lbs) depending on the configuration. Unlike the Model F50, the Model F80 does not provide rollers. The Model F80 is designed for customer setup of the machine and for subsequent addition of most features.

On the bottom of the chassis are three slots on the backplane used for the processor card (top slot) and the two possible memory cards (middle and bottom slots). A dummy memory card is installed in all shipped units for each unused memory slot for safety and proper machine cooling. The following ports are provided by the Model F80:

- One Ultra2 SCSI port for external attachment use (mini 68-pin VHDCI 1 connector) The industry standard VHDCI 68-pin connector on the backside of the Model F80 allows attachment of various LVD and SE external subsystems. A 0.3 meter converter cable, HDCI to P, mini-68 pin to 68-pin, (# 2118) can be used with older external SE subsystems to allow a connection to the VHDCI connector.
- 10/100 Mb/s Ethernet port (RJ-45 connector)
- Four serial ports (max. 230 KB/s, 9-pin D-shell)
- One parallel port (bi-directional)
- Test port
- Keyboard and mouse port

The test port is for diagnostics and is normally covered with a metal plate. It uses the same connector as the parallel port. To avoid confusion, this port should remain covered.

Internal Storage

The system comes preconfigured with a CD-ROM and a diskette drive and one free media bay for customer expansion, such as a tape device. Any devices in the media bays are connected to the internal F/W SCSI controller (no additional cable is required). The Model F80 features two six-packs providing 12 hot-swap disk bays and an additional boot disk bay for two additional disks (not hot-swappable). The two six-packs can be either equipped with a SCSI backplane (# 6553) or SSA backplane (# 6554). SCSI and SSA six-packs can be mixed. The SCSI backplane supports one inch and 1.6 inch drives. If the older 1.6 inch drives are used, these occupy two adjacent bays. The new SCSI carrier has one of two interposers between the drive and backplane, depending on whether the drive is 68-pin or 80-pin SCSI. All SCSI drives sold new with a Model F80 will be 80-pin drives. SCSI RAID is supported for the under-the-cover disks, but requires the addition of a SCSI RAID adapter. To cable a second internal SCSI RAID six-pack, a cable assembly is attached to an external SCSI port on the SCSI RAID adapter, run through a pass through opening in the rear bulkhead on the power supply side of the back panel, and attached to the SCSI backplane. SSA disks require the addition of an SSA adapter. To cable internal SSA, a cable assembly is attached to two external ports on the SSA adapter. This runs through a pass through opening in the rear bulkhead on the power supply side of the back panel and is attached to both ends of the SSA backplane. When

using both SSA six-packs, the cable runs to one end of each of the two backplanes with a short SSA cable in between the two backplanes. These configurations provide a loop, which is part of the SSA architecture. The SSA six-pack requires dummy jumper cards in vacant bays to maintain the SSA loop, and so can only support one inch drives. Booting from SSA disks attached to an Advanced SerialRAID adapter (# 6225) is supported from the six-pack or external SSA disks provided that the disks are arranged in a non-RAID configuration.

The optional dual boot bay holds a two-pack located between the operator panel and the top six-pack of the cabinet. They are SCSI drives attached to the same carriers as are used for the six-packs. They plug into a backplane in the two-pack that does not support hot-swap. The backplane is designed in such a way that the two disks can either be part of the same SCSI bus or attached to different SCSI busses to support mirroring of the boot image. The disks should have 80-pin connectors so that a flex interposer between the disks and the backplane is unnecessary. These two disks do not impact the two six-packs, so the six-packs can be used, for example, in a RAID configuration. If the dual boot bay option is not installed, one of the six-packs will not be able to provide RAID, because booting from RAID disks is not supported.

CPU Architecture

The key components in the CPU include the processor, the processor packaging, memory controller, memory subsystem, and the I/O interface.

RS64 III RISC Processor

The RS64 III processor card used in the Model F80 has the following attributes:

- 450 MHz or 500 MHz operating frequency
- 128 KB on-chip L1 instruction cache with parity and refetch
- 128 KB on-chip L1 data cache with ECC
- On-chip L2 cache directory
- 4 MB of off-chip L2 cache using ECC double data rate (DDR) SRAM per processor for 2-, 4-, and 6-way SMP and 2 MB of off-chip L2 cache using ECC Single Data Rate (SDR) SRAM for a 1-way system.
- PowerPC 6xx bus architecture, 16-byte (128-bit) wide bus interface

The RS64 III processor is available in two operating frequencies: 450 MHz and 500 MHz. The frequency is accomplished by leveraging IBM's copper technology (CMOS 7S) along with an innovative design of timing-critical paths. The copper technology and an improved manufacturing process allow the chip to operate at 1.8V. The lower operating voltage coupled with the smaller circuit dimensions result in reduced wattage in the RS64 III and allow additional function to be placed on the chip.

The size of the level one (L1) instruction and data caches is 128 KB each. Innovative custom circuit design techniques were used to maintain the one cycle load-to-use latency for the L1 data cache. The level two (L2) cache directory was integrated into the RS64 III chip, reducing off-chip accesses that impact performance.

IBM used double data rate (DDR) SRAM technology for the L2 cache in the RS64 III processor. DDR technology provides two transfers of data on the 16-byte (128-bit) wide L2 data bus every SRAM clock cycle. The DDR SRAM technology also reduced L2 access latency as measured by nanoseconds.

Processor Boards

The processor boards used in the Model F80 for 1-, 2-, 4-, and 6-way SMP configurations come in the form of a single book and are described below.

Single Processor

A single processor board consists of a single RS64 III processor operating at 450 MHz, on-board memory slots, and a memory controller in a single book. Upgrades to additional processors require changing of the processor book. However, the single processor board is a cost-reduced package.

2- and 4- Way SMP

A 2-way SMP configuration is provided by a processor board consisting of a pair of RS64 III processors operating at 450 MHz and a memory controller. Expansion to 4-way SMP is provided by interfacing an additional processor board consisting of a pair of RS64 III processors operating at 450 MHz. However, the upgrade from 2-way to 4-way SMP is offered as a book swap, since the addition of the processor card on the processor book is too delicate for field handling.

6-Way SMP

A 6-way SMP configuration uses two processor boards that are interfaced to each other. One processor board consists of a pair of RS64 III processors operating at 500 MHz and a memory controller. The other processor board consists of four RS64 III processors operating at 500 MHz. Upgrades from a 2- or 4-way SMP to a 6-way SMP are offered as a book swap.

Memory Controller

A single custom chip provides the function of the memory controller and the I/O hub in the Model F80. The controller chip provides interfaces to processors, memory, and the I/O subsystem.

The RS64 III processors on the processor boards are connected to the memory controller through the PowerPC 6xx bus. The controller chip is a part of the first processor board. The memory controller provides a single 6xx bus interface in a single processor configuration. For 2-way SMP configurations, the controller provides a 6xx bus interface to the pair of RS64 III processors present in the same board. The memory controller provides another 6xx bus interface for CPU expansion using an additional processor board. The 4- and 6-way SMP configurations consists of a total of two processor boards which uses the two 6xx bus interfaces provided by the memory controller installed together in a book.

In the Model F80, the 6xx bus is a 16-byte (128-bit) wide bus and the operating clock rate of the bus depends upon the processor clock speed. The 6xx bus operates at a clock rate

of 150 MHz for a processor clock speed of 450 MHz, and 125 MHz for a processor clock speed of 500 MHz.

Memory Subsystem

The memory controller provides two memory bus interfaces and provides the reliability functions of ECC as well as memory scrubbing. Memory scrubbing provides a built-in hardware function that is designed to perform continuous background reads of data from memory, checking for correctable errors. The memory configuration for a single processor configuration and 2-, 4-, or 6-way configurations is explained as follows:

- In a single processor configuration, the on-board memory, consisting of eight DIMM slots, is interfaced to one of the two memory interfaces in the controller. The other interface is used by a separate riser memory card. The riser memory card provides 16 DIMM slots. While the DIMM slots in the on-board memory are populated in pairs, the slots in the riser memory card are populated in quads. The minimum configuration requires a pair of DIMMs in the on-board memory. Once the on-board memory slots are filled and more memory capacity is desired, the DIMMs are moved to the riser memory card and the next increment is made as a quad. The single processor configuration can provide a maximum memory of 8 GB by using the riser memory card and populating each of the 16 slots using 512 MB DIMMs.
- In 2-, 4-, or 6-way SMP configurations, the memory is provided using two separate riser memory cards, each with 16 DIMM slots and populated with DIMMs in quads. The two riser cards are interfaced to the two memory interfaces in the memory controller. The minimum 2-way SMP configuration requires a single riser memory card populated with a quad of DIMMs. The second riser memory card with minimum of a quad of DIMMs can be configured only after the 16 DIMM slots in the first riser memory card are fully populated. 2-, 4-, or 6-way processor configuration can support up to 16 GB maximum memory by using the two memory riser cards fully populated with 512 MB DIMMs. The Model F80 uses 200-pin 10ns SDRAM DIMMs. DIMMs of equal sizes must be used, while populating in pairs or quads. DIMM size used in one pair or quad can, however, coexist with a different DIMM size used in another pair or quad. In the Model F80, the bus interface from each riser memory card to the memory controller is 8-bytes (64-bits) wide and operates at clock rate double that of the PowerPC 6xx bus. No additional memory bandwidth can be achieved by splitting memory between cards.

Bus Bandwidth

The following are the theoretical maximum bandwidths as applicable for a 6-way 500 MHz SMP configuration:

- Total memory bandwidth: 2 GB/s
- Total processor bandwidth: 2 GB/s
- Total I/O bandwidth: 1 GB/s (500 MB/s bi-directional)

The following are the theoretical maximum bandwidths applicable for 2-, or 4-way 450 MHz SMP configurations:

- Total memory bandwidth: 2.4 GB/s
- Total processor bandwidth: 2.4 GB/s
- Total I/O bandwidth: 1 GB/s (500 MB/s bi-directional)

I/O Hub Function

The memory controller also functions as the I/O hub. The controller provides two RIO (remote I/O) ports. The two RIO ports are attached to an I/O host bridge chip. Each RIO port has two unidirectional 1-byte (8-bit) wide links. All the I/O transfers take place using one primary RIO port, which operates at 500 MHz (500 MB/s bi-directional or an aggregate of 1 GB/s). The controller uses the other RIO port, which operates at 250 MHz (250 MB/s bi-directional or aggregate of 500 MB/s), as a fail-over to the primary RIO port. In contrast to Model H80 or Model M80, these RIO connections in the Model F80 are not visible outside the cabinet, but the function is identical.

Internal I/O Architecture

As already discussed, the system includes one I/O host bridge chip managing all the I/O between the I/O adapters and the memory controller using RIO connections. On the other side, the I/O host bridge provides two primary PCI busses, operating at 66 MHz and 64-bits wide. The service processor and a PCI-to-PCI bridge chip are connected to the first primary PCI bus. The PCI-to-PCI bridge provides three 64-bit hot-plug PCI slots and the onboard dual SCSI adapter (F/W SCSI internal, Ultra2 SCSI external). A PCI-to-ISA bridge chip is connected to the service processor providing an ISA bus. The ISA bus is used by the National Super I/O chip providing the floppy drive controller, two of the four serial ports, keyboard and mouse ports, and the parallel printer interface. A 16552 DUART chip is also connected to the service processor providing the other two serial ports. The second bus is connected to two additional PCI bridges, which provide seven more 64-bit hot-plug PCI slots. The onboard 10/100 Mb/s Ethernet adapter is connected to this chip. Each slot represents a separate PCI bus, which simplifies the hot-plug functionality.

Hot-Plug PCI Adapters

The function of hot-pluggable PCI adapters is to provide concurrent additions or removals of PCI adapters when the system is running. This function is explained in the following paragraphs.

In the chassis, the adapters installed inside the slots are protected by plastic separators designed to prevent grounding and damage when adding or removing adapters. The hot-plug LEDs outside the chassis indicate if an adapter can be plugged in or removed from the system. These LEDs are also visible inside the chassis. Inside, the light from the LED is routed to the top of the plastic separators, using light pipes, which makes it very easy to locate the right slot. The hot-plug PCI adapters are secured with retainer clips on top of the slots; therefore, you do not need a screwdriver to add or remove a card and there is no

screw to drop inside the chassis causing damage to the system. The function of hot-plug is not only provided by the PCI slot, but also by the function of the adapter. Most adapters are hot-pluggable, but some are not. Be aware that some adapters must not be removed when the system is running, for example, the adapter with the operating system disks connected to it, or the adapter that provides the system console. Refer to the *PCI Adapter Placement Reference Guide, SA38-0538* for further information.

Software Requirements

The Model F80 requires AIX 4.3.3 with the AIX 4330-03 recommended maintenance package (APAR IY09047), which is included on all pre-installed systems and on the 04/2000 Update CD that ships with AIX 4.3.3 as of April 2000. In addition, there is APAR IY09814, which includes additional fixes that were not available before the 4330-03 package was shipped. In order to install the Model F80 from CD, you need an AIX 4.3.3 CD dated 04/2000 (LCD4-0286-05) or later, because the system will not boot from older AIX 4.3.3 CDs. You can also download the actual maintenance level from the Internet to install the machine using Network Installation Manager.

Investment Protection and Expansion

The following sections discuss how configurations, upgrades, and design features help you lower your cost of ownership.

High Availability

Using the HACMP clustering solution, which is available across the entire range of RS/6000 servers, reliability of the system is extended. The HACMP solution exploits redundancy between server resources and provides application uptime. The Model F80 is available in a high-availability cluster solution package named the HA-F80. This solution consists of the following components:

- Two Model 7025-F80 Enterprise Servers
- AIX Version 4.3.3 operating system (unlimited user license), or later
- HACMP 4.3.1 cluster software, or later
- One 7133-T40 SSA disk subsystem with at least four disk drives
- All necessary redundant hardware and cables

This solution is sold at a price lower than the sum of its parts. Ask your IBM Business Partner or IBM representative for further information.

Reliability, Availability, and Serviceability (RAS) Features

Some RAS features such as redundant power supplies or N+1 hot-plug fans are already discussed. Additional topics are covered in the following sections.

Error Recovery for Caches and Memory

The RS64 III processor L1 cache, the L2 cache, system busses, and the memory are protected by error correction code (ECC) logic. The ECC codes provide single bit error correction and double bit error detection for the L2 cache and the memory. All recovered

error events are reported by an attention interrupt to the service processor, where they are monitored for threshold conditions.

The standard memory card has single error-correct and double-error detect ECC circuitry to correct single-bit memory failures. The double-bit detection helps maintain data integrity by detecting and reporting multiple errors beyond what the ECC circuitry can correct. In many cases (using DIMMs with 18 DRAM chips and when memory is configured in quads, for example), memory chips are organized such that the failure of any specific memory module only affects a single bit within an ECC word (bit scattering) thus allowing for error correction and continued operation in the presence of a complete chip failure (chip kill recovery). Another function, named memory scrubbing, provides a built-in hardware function, which performs continuous background reads of data from memory, checking for correctable errors. Correctable errors are corrected and rewritten to memory and a threshold counter is maintained that will signal the service processor with a special attention when the threshold is exceeded.

Dynamic CPU Deallocation

The processors are continuously monitored for errors such as L2 cache ECC errors. When a predefined error threshold is met, an error log with warning severity and threshold-exceeded status is returned to AIX. At the same time, the service processor marks the CPU for deconfiguration at the next boot. In the meantime, AIX will attempt to migrate all resources associated with that processor (tasks, interrupts, etc.) to another processor, and then stop the failing processor.

The capability of dynamic CPU deallocation is only active in systems with more than two processors because device drivers and kernel extensions, which are common to multi-processor and uniprocessor systems, would change their mode to uniprocessor mode with unpredictable results.

Persistent CPU and Memory Deconfiguration

CPUs and memory modules with a failure history are marked bad to prevent them from being configured on subsequent boots. This history is kept in the VPD 3 records on the FRU 4, so the information moves physically with the FRU and is cleared when the FRU is replaced, and stays with the failed FRU when it is returned to IBM. A CPU or memory module is marked bad when:

- It fails BIST 5 /POST 6 testing during boot (as determined by the service processor).
- It causes a machine check or check stop during runtime and the failure can be isolated specifically to that CPU or memory module (as determined by the service processor).
- It reaches a threshold of recovered failures (for example, ECC correctable L2 cache errors, see the preceding) that result in a predictive call-out (as determined by service processor).

During CEC initialization, the service processor checks the VPD values and does not configure CPUs or memory that are marked bad, much in the same way that it would deconfigure them for BIST/POST failures.

I/O Expansion (RIO) Recovery

The RIO interface supports packet retry on its interface, which means that it will automatically try to resend a packet if it gets no acknowledgment or a bad response until a time-out threshold is reached. RIO also supports a closed loop topology configuration, which is required for RS/6000 products. RIO hubs will automatically attempt to reroute packets through the alternate RIO port if a successful transmission cannot be completed (for example, the retry threshold is exceeded) through the primary port. Therefore, no single link failure in the RIO loop will cause the system to go down, although the failure will be reported for deferred maintenance.

PCI Bus Error Recovery

As described in the PCI slot section, every slot is connected through a PCI-to-PCI bridge chip to a primary PCI bus; thereby, each slot is logically and physically isolated onto its own individual PCI bus. This fact provides a special error-handling mode that allows the bridge chip to freeze access to an adapter when a PCI bus error occurs on the interface between that adapter and bridge chip. In this frozen mode, DMA 7 is blocked, write operations to that device address space are discarded, and read operations result in a return value of all 1s. Device drivers can be programmed to look for these dummy responses on loads and can attempt recovery. The AIX support for this function is not available yet.

System Power Control Network (SPCN)

SPCN consists of a set of power/environmental controllers, interconnected by a set of serial communication links. In Model F80 systems, the SPCN function is integrated into the service processor and provides the following functions:

- Powering all the system parts up or down, when requested. The SPCN hardware has connections to the VPD that is resident on each of the pluggable cards and the backplane. The VPD is located on each of the cards in the form of an I²C chip. This chip is accessed during initial power on sequence and the data contents are read by the service processor. Using this function, the service processor decides not to use components that are marked bad.
- Powering down all the system parts on critical power faults
- Monitors power, fans, and thermal conditions in the system for problem conditions, which result in an EPOW. EPOW stands for environmental and power off warnings and is a function to inform the service processor or the operating system early, about an event that happened in the hardware. There are different warnings; such as cooling warnings or power fail warnings which result in entries in the error log. If there is a serious error, such as the temperature reaches a specific limit, the system will be shutdown.

- Reporting power and environmental faults, as well as faults in the SPCN network itself, on operator panels and through the service processor
- Assigning and writing location information into various VPD elements in the system

Disk Redundancy (Mirroring, RAID, Dual Controllers)

RS/6000 and AIX provide a number of options for increasing the robustness of storage subsystems, all of which involve some level of redundancy of disks and/or adapters. AIX disk mirroring provides the ability to define transparent double or triple redundancy of disk data by mapping disk write data to two or three physical disks. On disk reads, the request is issued to all disks in the mirror group, and the first error-free response is returned, which also has some performance benefits. If one of the disks fails, the data is still readable from the other disk(s). There are also customer options for SCSI and SSA RAID controller adapters, which can provide the same protection with better performance and less redundancy overhead. Also available are storage subsystems that provide under-the-covers redundancy for high availability. To provide protection against adapter failures, AIX also supports dual-controller options where the same disk subsystem can be accessed through both a primary adapter path and through a backup adapter path if the primary fails.

Hot Swap Disk and Service Aid

The hardware within the system is designed with the capability to remove and install disks without powering down the system. An AIX Diagnostics Service Aids provides positive identification (a blinking LED) at the disk device as a visual aid for removal.

Service Processor

The Model F80 has an integrated enhanced service processor. When the system is powered down, but still plugged into an active power source, the service processor and SPCN functions are still active under standby power. This function provides enhanced RAS by not requiring AIX to be operational for interfacing with a system administrator or service director for RS/6000. This means that all service processor menu functions (using the local, remote, or terminal concentrator console), as well as dial out capability, are available even if the system is powered down or unable to power up. The next sections talk about selected features of the enhanced service processor.

Automatic Reboot

The system will automatically reboot (if the appropriate policy flags are set) in the following conditions:

- Power is restored after a power loss during normal system operation
- Hardware checkstop failures
- Machine check interrupt
- Operating system hang (Surveillance failure)
- Operating system failure



Surveillance

The service processor, if enabled through service processor setup parameters, performs a surveillance of AIX through a heartbeat mechanism. If there is no heartbeat within the time-out period, the service processor does the following:

- Creates a system reset to allow an AIX dump to occur
- Upon receiving a reboot request, either after the dump or immediately if dump is not enabled, the service processor captures scan debug data for the system.
- Reboots the system

Dial-Out (Call Home), Dial-In

If enabled, the service processor can dial a preprogrammed telephone number to report errors. When enabled, it is also possible to access the service processor remotely through a modem connection. When the service processor is in standby mode, because the system is powered off, or an error occurred, the service processor monitors an incoming phone line to answer calls, prompts for a password, verifies the password, and remotely display the standby menu. The remote session can be mirrored on the local ASCII console if the server is so equipped and the user enables this function.

Processor and Memory Boot Time Deconfiguration

As described previously, processors can be dynamically deconfigured by the system. It is also possible to deconfigure processors and also memory with menus of the service processor for benchmarking reasons. For further information, refer to the *RS/6000 Enterprise Server Model F80 Service Guide, SA38-0568*.

Fast Boot

This feature, set as the default, allows you to select the IPL type, mode, and speed for your boot capabilities using service processor menus. Selecting fast boot results in several diagnostic tests being skipped and a shorter memory test being run; therefore, the startup process is faster, but possible problems might not be discovered at startup.

Service Processor Restart

The service processor design for the Model F80 includes the ability to reset the service processor. This enables the system firmware to force a hard reset of the service processor if it detects a loss of communication. Since this would typically occur while the system is already up and running, the service processor reset will be accomplished without impacting system operation.

Boot to SMS Menu

The Boot Mode menu allows one to select, among other things, to boot to SMS menu. This function provides booting into SMS menu without pressing a key. This function is useful, because it is not necessary to wait in front of the system and press the **F1** (graphic display) or **1** (ASCII terminal) at the right moment.

External Storage Expandability

The storage expansion for the Model F80 is can be provided through several IBM storage options. The storage subsystems can be connected externally as stand-alone tower or from within a rack. External disk storage capacity can also be provided by attaching the Model F80 to storage servers. Using differential Ultra SCSI, the Model F80 can be attached to the IBM Enterprise Storage Server. By using the Fibre Channel Adapter, the Model F80 can be attached to the IBM Fibre Channel RAID Storage server or the IBM Enterprise Storage Server.

Reference

The following sections list additional materials available for further research.

System Documentation

For more detailed information, refer to the following documents:

- RS/6000 Enterprise Server Model F80 Installation Guide, SA38-0569
- RS/6000 Enterprise Server Model F80 User's Guide, SA38-0567
- RS/6000 Enterprise Server Model F80 Service Guide, SA38-0568
- PCI Adapter Placement Reference Guide, SA38-0538

Select IBM Redbooks

The following IBM Redbooks are related to the material discussed in this paper:

- RS/6000 Systems Handbook 2000, SG24-5120 (Available June 2000)
- RS/6000 S-Series Enterprise Servers Handbook, SG24-5113
- IBM Enterprise Storage Server, SG24-5465
- Monitoring and Managing IBM SSA Disk Subsystems, SG24-5251
- AIX 4.3 Differences Guide, SG24-2014
- NIM: From A to Z in AIX 4.3, SG24-5524
- AIX Logical Volume Manager, from A to Z: Introduction and Concepts, SG24-5432
- Understanding IBM RS/6000 Performance and Sizing, SG24-4810

Select Internet Links

- <http://www.rs6000.ibm.com/>
- <http://www.rs6000.ibm.com/hardware/enterprise/>
- http://www.rs6000.ibm.com/resource/hardware_docs/index.html
- http://www.rs6000.ibm.com/cgi-bin/ds_form
- <http://www.rs6000.ibm.com/support/micro/>
- <http://www.ibm.com/servers/aix/>
- <http://www.chips.ibm.com/>
- <http://www.research.ibm.com/topics/serious/chip/>
- <http://www.storage.ibm.com/>
- <http://www.hursley.ibm.com/~ssa/rs6k/>
- <http://www.redbooks.ibm.com/>



Appendix C

Oracle Application Server



Oracle Application Server: Executive Summary

Copyright © Oracle Corporation 1999

Introduction

This document discusses Oracle Application Server (OAS) and its many benefits for today's rapidly evolving corporate, Web-based computing environments and e-businesses. OAS is specifically designed to address the many challenges faced by IT managers – such as scalability, performance, manageability, reliability, and security while offering faster time-to-market for sophisticated, transaction-oriented applications. OAS is the only available solution that embraces an integrated vision for creating fully featured, enterprise-class applications and extending them to the Web. With OAS, organizations can benefit from a streamlined, component-based deployment platform that leverages open standards (such as CORBA, IIOP, and Java) and readily supports all major Web servers, databases, and legacy systems. The integrated services approach used in OAS helps to eliminate the extra layers of logic (and hidden development costs) often found in 'bolted-on' solutions, while providing a solid foundation for growth and a high quality of service to users.

The Challenge for Web-based Business Applications

As companies move to exploit the benefits of the Web, they look for solutions that will help to ease the development and deployment of new and existing applications. Today, the typical HTTP Web server, first used for publishing content in the form of static information, falls short of satisfying these contemporary requirements. Although HTTP servers can accommodate simple application scenarios, they lack the ability to manage transactions and scale to a larger base of users – two critical requirements for deploying full-featured applications on the Web.

Early Computing Architectures

Since the early years of computing, corporate applications were written as monolithic 'chunks' of code, combining the business and database logic into a single back end repository that often ran on a mainframe. Since application logic could not be easily isolated, software upgrades and regular maintenance placed a heavy burden on programmers and system administrators. Even a simple feature change in an application could take months before it was implemented. In more recent history (the last ten years), client/server or two-tier technologies have proven more successful for faster development and deployment of business-critical applications. However, the client/server (C/S) model, which places some of the application logic on both client and server, can be vexing when it comes to maintenance. While C/S-based Web applications are often suitable for medium to large workgroups, they do not scale well. As the number of users and transactions increases into the thousands and beyond, C/S systems can quickly exceed the performance limits of their deployment environment. A separate but equally important concern in C/S architectures is their inability to maintain a persistent connection between the client and the data source. This is an especially important consideration for today's Web-based applications. Although certain unrefined methods can be used to connect Web clients to databases, the resulting "stateless" connection requires that a client make a new



connection to the server for every transaction, thereby increasing network traffic and diminishing performance. Issues related to scalability, maintenance, and connection persistence can be further exacerbated as enterprise managers attempt to distribute applications across diverse platforms and operating systems.

Finding New Solutions by Expanding the Role of the Middle Tier

Enterprise architects and programmers are now moving the business logic out of the front and back end tiers into a more flexible middle tier. This approach isolates the mission-critical application logic and allows developers to create one or more middle tier servers to help further segment discreet application functions. The resulting environment, referred to as an n-tier architecture, offers the ultimate in flexibility for both development and deployment. In this scenario, developers can partition application logic into components, which can be easily modified or re-used and deployed across several physical machines on different platforms. Middle tier servers can also offer connection management and allow clients to share and maintain ‘stateful’ connections. By using the n-tier model, system designers can overcome the past limitations of client/server architectures. With the expanded middle tier, clients and servers can maintain connections across multiple transactions, and applications can be distributed across multiple machines and platforms. The end result – improved scalability, performance, and overall flexibility.

Refining the Middle

The use of n-tier architectures is an essential first step in the creation of robust, enterprise class, Web applications. However, the complete benefits of n-tier architectures can only be realized when there exists a correctly articulated and implemented infrastructure that addresses the many issues surrounding development and deployment. To meet these requirements, a new product category has emerged – the application server. By logically situating the application server between the client and the back-end database(s), an additional layer of independence and functionality is created. The application server is the centerpiece of the middle tier and provides a unified platform for shared enterprise applications such as order entry. The application server provides vital services such as security, message brokering, database connectivity, transaction management, and process isolation while delivering enterprise quality of service. Today, enterprise managers are recognizing the application server as the best way to address the many challenges of large scale, transaction-oriented, development and deployment environments.

Oracle Application Server

Just as there are several ways to implement an n-tier architecture, there are also several application servers that have widely varying approaches and capabilities when it comes to managing the middle tier. In contrast to the many point solutions that claim to be application servers, Oracle Application Server (OAS) is the only solution that truly integrates all of the core services and features required for building, deploying, and managing high performance, n-tier, transaction-oriented, Web applications within an open standards framework. These essential capabilities include:

- HTTP Web Server and support for popular Web servers
- Database and legacy access middleware for connection to all major databases
- CORBA ORB support for scalable, cross platform, distributed object deployment
- TP Monitor capability for load balancing, pooling and transactions
- Network Services like security and directory
- Message-oriented middleware for extensibility that simplifies the enabling of new applications

Through the convergence of the above facilities within OAS, Oracle has defined the enterprise application server category. OAS offers cross-platform support for all types of network clients (HTML, Java, CORBA), Web servers, and databases, which preserves existing investments in legacy and client/server systems.

Enterprise managers can also realize these additional development and deployment benefits:

- RAD (Rapid Application Development): OAS maintains a completely open development space that facilitates use of a wide variety of popular IDEs and development tools including tight integration with Oracle's Jdeveloper that help developers build network ready applications in Java. In addition, Oracle Designer and Oracle Developer are unique in offering model-based development for the Web. The OAS development environment hides and automates lower-level system 'plumbing' so that developers can focus on creating business logic.
- Integrates Latest Component-Based Industry Standards: OAS accelerates the convergence of an assortment of middleware products by incorporating defacto standards such as CORBA, Internet Inter ORB Protocol (IIOP), and Enterprise JavaBeans. Object-oriented presentation (GUI) and business components reside on the application server and can be deployed independently. This modularity provides major benefits in the areas of scalability, maintenance, and code reuse and readily facilitates distributed, cross platform deployment.
- System Administration: Centralized deployment and management of applications occurs on a standard server platform rather than on thousands of individual clients. OAS offers specialized, easy-to-use administration tools for security, configuration, start/stopping, monitoring server components, and performance optimization.

SUMMARY

Oracle Application Server already offers a comprehensive solution for building, deploying, Web-enabling, and managing enterprise-class applications. The integrated yet open character of OAS within an n-tier architecture results in better performance and easier manageability for transaction-based applications. Using OAS, enterprises can run products such as Oracle Applications, Oracle Enterprise Manager, 4GL Tools, Internet Commerce Server, and other packaged applications from Oracle and/or third party suppliers – all on a common platform. For enterprise managers that are looking for a fully



featured Web solution with tight database integration, OAS has the right answer. Because of its flexibility, OAS helps information architects and developers shape and deliver applications around the special needs of the business – not vice versa. By offering the highest levels of reliability, scalability, and integration, OAS ensures a clear path for future growth. OAS is the platform of choice for transaction oriented, n-tier deployment of all kinds and is backed by the solid support of Oracle Corporation – a world leader in business applications and database technologies.