Processing Scientific Data: Three Unique Case Studies

Ronald Mak

Presented August 9, 2010
IBM Almaden Research Center
Presentation Agenda

- Case Study 1
  **The Collaborative Information Portal (CIP)**
  Mars Exploration Rovers (MER) Mission
  NASA Ames Research Center

- Case Study 2
  **The Systems Health Information Portal (SHIP)**
  NASA Ames Research Center

- Case Study 3
  **Shot Data Management**
  The National Ignition Facility (NIF)
  Lawrence Livermore National Laboratory

- Important Lessons Learned
Case Study 1:
The Collaborative Information Portal (CIP)
Case Study 1: The Collaborative Information Portal

The **Collaborative Information Portal (CIP)** was a key ground-based application used by the NASA’s Mars Exploration Rovers (MER) mission.

Mission scientists, engineers, and researchers at JPL and around the world used CIP to access mission data in a secure and organized manner over the Internet.

- My role: Senior Scientist
  Research Institute for Advanced Computer Science (RIACS)
  - Architect and lead developer of the CIP middleware, 2002-2003
  - Mission support at the NASA Ames Research Center and JPL, 2004

Mission Control Center, JPL
Mars Exploration Rovers Mission
Mars Exploration Rover Mission

- Twin robot geologists search for liquid water on Mars in the past.

- **Launched:** June 10 and July 7, 2003
- **Landed:** January 3 and 24, 2004
- **Duration:** 90 days

But after over six and a half Earth years, both are still operating (although one is has been stuck in a sand pile for several months).

- **Mission Center:**
  Jet Propulsion Laboratory
  Pasadena, CA
MER Mission Requirements

- Time management
- Data management
- Personnel management

Mission Geology Lab, JPL
The Collaborative Information Portal (CIP)
Clocks

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOL 51</td>
<td>18:31</td>
<td>LST-A</td>
</tr>
<tr>
<td>Feb 24 2004</td>
<td>09:39</td>
<td>Los_Angeles/Americ...</td>
</tr>
<tr>
<td>DOY 55</td>
<td>17:39</td>
<td>UTC</td>
</tr>
<tr>
<td>Feb 25 2004</td>
<td>04:39</td>
<td>Canberra/Australia</td>
</tr>
<tr>
<td>Feb 24 2004</td>
<td>10:39</td>
<td>Phoenix/Americ...</td>
</tr>
</tbody>
</table>

- Mars time displayed in Sols
- UTC shown in Day of Year
- Tick-over at top-of-minute
- Local time
- Time at DSN location
- Time at collaborator’s location
Schedule Viewer

- Multiple time scales
- Unified schedules
- Schedule update indicator
- "Now" bar
Data Product Navigator
New File Notification
Three–Tiered Enterprise Architecture

- **Client**
  - Java application (Swing)

- **Middleware Server**
  - Web services, Enterprise JavaBeans (EJB), Java Message Service (JMS)
  - Service-oriented architecture (SOA)

- **Data Repository**
  - Mission file servers (Unix)
  - File monitor (Java application)
  - Data loader (Java application)
  - Database (Oracle)
Architecture Overview

[Diagram showing the architecture overview with components such as CIP Middleware, Application Server, EJBs, CIP Data Management System, Meta Database, CIP Data Acquisition, Monitor, Loader, and MER Mission Data Servers.]

PC, Mac, Solaris, Linux, MERBoard

Solaris, Linux
CIP Middleware Services

- User management
- Metadata
- Schedules
- Mars and Earth time
- File and directory
- Message
Middleware Technologies

- **Enterprise JavaBeans (EJBs)** to achieve reliability, scalability, security, platform independence, and standards.
  - Stateless session beans
  - Stateful session beans
  - Message-driven beans

- **Web services** to expose the remote methods of the Service Provider EJBs to the client applications.

- **Java Message Service (JMS)** for synchronous and asynchronous messaging.

- **BEA WebLogic** application server.
Middleware Architecture

- Web Services expose the remote methods of the Service Provider beans.
- HTTPS encrypts the transmissions and gets them through the firewall.
- Stateless session beans are the Service Providers.
Caching Data

- Some middleware services cache data to improve performance.
- **Stateful session beans** cache data in memory.
- The cache beans access the backend data store.

Some middleware services cache data to improve performance.

**Stateful session beans** cache data in memory.

The cache beans access the backend data store.
Event Notification

- Publish-subscribe model with topics
- Clients poll for their messages via web services.
- The consumer objects retrieve the messages.
- User proxies look up JMS consumer objects that represent the active subscriptions.
- Messages are converted for return via web services.

“A new pancam image file has just been downloaded from Mars!”
The Publisher bean publishes messages to the Broadcast Messages topic.

Message–driven bean (MDB) archives broadcast messages to persistent storage for later retrieval.

The Delegate bean retrieves old messages for browsing.
Metadata Generation

- File changes are logged by `nfslogd` system program and the File Monitor, or found by the File Detector.
  - A message is sent to the JMS Monitor Topic for each file change.
- The Data Loader generates metadata about the file.
  - Retrieves messages from the monitor topic.
How Reliable Has CIP Been?

- The middleware code remained unchanged since the code freeze in early November 2003.
- CIP became operational in mid–December 2003, two weeks before the first rover landed.
- CIP has run continuously for as long as 153 Earth days without interruption.
  - As of the end of the day on December 31, 2005, the mission had lasted 17,904 Earth hours.
  - During that entire period, CIP has been down less than 10 hours, which is > 99.9% uptime.
- CIP was finally shut down this summer after 6½ years of failure-free service.
Example Log Entries: Streamer Service

2004-12-20 20:23:39,140 DEBUG: Completed upload of file '/opt/bea/user_projects/cip/conf/preferences/sthompso.preferences': 35659 bytes
'/opt/bea/user_projects/cip/conf/global.properties': 13453 bytes
2004-12-20 20:28:57,516 INFO : jdoe: Streamer.getPreferences(user)
2004-12-21 19:17:44,584 DEBUG: Completed download of file '/global/nfs2/merb/ops/ops/surface/tactical/sol/120/opgs/edr/jpeg/1P138831013ETH2809P2845L2M1.JPG': 1876 bytes

- Timestamp, user name, method name
- Information about the uploaded or downloaded file
Middleware Monitor Utility

- Server statistics
### Middleware Monitor Utility

- **Users**

<table>
<thead>
<tr>
<th>User ID</th>
<th>Logged In</th>
<th>Logged Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>masked</td>
<td>Wed 2005.01.12 @ 02:38:19 GMT</td>
<td>Wed 2005.01.12 @ 02:48:27 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 20:01:19 GMT</td>
<td>Tue 2005.01.11 @ 20:01:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 08:39:15 GMT</td>
<td>Tue 2005.01.11 @ 17:10:47 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 16:00:20 GMT</td>
<td>Tue 2005.01.11 @ 16:00:20 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 12:00:19 GMT</td>
<td>Tue 2005.01.11 @ 12:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 08:00:17 GMT</td>
<td>Tue 2005.01.11 @ 08:00:17 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 21:14:02 GMT</td>
<td>Tue 2005.01.11 @ 04:07:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Tue 2005.01.11 @ 08:00:18 GMT</td>
<td>Tue 2005.01.11 @ 00:30:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 06:00:18 GMT</td>
<td>Mon 2005.01.11 @ 20:00:20 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 20:00:20 GMT</td>
<td>Mon 2005.01.11 @ 20:00:20 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.11 @ 20:00:20 GMT</td>
<td>Mon 2005.01.11 @ 20:00:20 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.11 @ 08:00:18 GMT</td>
<td>Mon 2005.01.11 @ 08:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 16:00:17 GMT</td>
<td>Mon 2005.01.10 @ 16:00:17 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 12:00:19 GMT</td>
<td>Mon 2005.01.10 @ 12:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 08:00:18 GMT</td>
<td>Mon 2005.01.10 @ 08:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.09 @ 03:32:18 GMT</td>
<td>Mon 2005.01.10 @ 03:32:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 14:00:19 GMT</td>
<td>Sun 2005.01.10 @ 20:00:20 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Mon 2005.01.10 @ 04:00:19 GMT</td>
<td>Mon 2005.01.10 @ 04:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.09 @ 20:00:18 GMT</td>
<td>Sun 2005.01.09 @ 15:00:16 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.09 @ 19:50:17 GMT</td>
<td>Sun 2005.01.09 @ 19:50:17 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.09 @ 12:00:18 GMT</td>
<td>Sun 2005.01.09 @ 12:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.09 @ 09:00:10 GMT</td>
<td>Sun 2005.01.09 @ 09:00:10 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.09 @ 04:00:14 GMT</td>
<td>Sun 2005.01.09 @ 04:00:14 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.09 @ 03:14:18 GMT</td>
<td>Sun 2005.01.09 @ 03:14:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sun 2005.01.08 @ 20:00:16 GMT</td>
<td>Sat 2005.01.08 @ 20:00:16 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 18:00:18 GMT</td>
<td>Sat 2005.01.08 @ 18:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 12:00:19 GMT</td>
<td>Sat 2005.01.08 @ 12:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 08:00:17 GMT</td>
<td>Sat 2005.01.08 @ 08:00:17 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 04:00:18 GMT</td>
<td>Sat 2005.01.08 @ 04:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Sat 2005.01.08 @ 00:00:16 GMT</td>
<td>Sat 2005.01.08 @ 00:00:16 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 20:00:18 GMT</td>
<td>Fri 2005.01.07 @ 20:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 05:11:08 GMT</td>
<td>Fri 2005.01.07 @ 18:05:03 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 18:00:16 GMT</td>
<td>Fri 2005.01.07 @ 18:00:16 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 12:00:19 GMT</td>
<td>Fri 2005.01.07 @ 12:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 08:00:19 GMT</td>
<td>Fri 2005.01.07 @ 08:00:19 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Fri 2005.01.07 @ 04:00:18 GMT</td>
<td>Fri 2005.01.07 @ 04:00:18 GMT</td>
</tr>
<tr>
<td>masked</td>
<td>Thu 2005.01.06 @ 19:21:50 GMT</td>
<td>Fri 2005.01.07 @ 03:38:13 GMT</td>
</tr>
</tbody>
</table>

---

August 9, 2010

Presented at the IBM Almaden Research Center
Middleware Monitor Utility

- Files uploaded or downloaded
Middleware Monitor Utility

- Cache contents
Middleware Stress Tester

- Simulate multiple users performing client operations simultaneously.

- Can the server handle a heavy load?
  - What is its behavior when it starts to fail?
CIP Summary

- Provided **situational awareness** for MER mission personnel
  - Download images, data, schedules, current time
  - Collaboration
  - Alerts

- **Service-oriented architecture (SOA)**
  - Java Swing and C++ desktop client applications
  - J2EE technologies
  - Web services
  - Mission file servers
  - Metadatabase
  - BEA WebLogic, Oracle RDBMS
  - Open source software
Case Study 2:
The Systems Health Information Portal (SHIP)
Case Study 2: The Systems Health Information Portal

The **Systems Health Information Portal (SHIP)** constantly monitors and analyzes sensor data gathered from a manned NASA space vehicle.

If there is a **fault** with the vehicle, SHIP quickly **analyzes** the situation and recommends **corrective actions** for the astronauts on board, even if contact with ground control is lost.

- My role: **Project Scientist**
  University of California at Santa Cruz
  - Architect, lead developer, and systems integrator, 2004-2005
SHIP Requirements

- **Access to disparate data sources**
  - Databases, files, live instrument streams, web services, web pages, programs

- **Multiple data formats**
  - Relational tables
  - XML, text, and binary files
  - Proprietary and legacy data formats

- **Generate and manage reports**
  - Fault analyses
  - Prognostications
  - Procedure manuals

- **Test bed**
  - International Space Station (ISS)

---

Rendezvous in low Earth orbit (LEO) of a manned space capsule with the main rocket engines in preparation for a trip to the Moon or to Mars.
SHIP Architecture
SHIP Architecture
SHIP Architecture
SHIP Architecture

Primary Data Sources

Databases

Instruments

Files

Programs

Matching Engine

Case-Based Reasoning

Web Server

Application Server

Business objects

Service objects

Web services

Enterprise Information Integration Server

Browser-based applications

Web objects

Desktop applications

Casbase
SHIP Architecture

Primary Data Sources

- Databases
- Instruments
- Files
- Programs

Case-Based Reasoning

Rule Engine

- Analyses, Prognostications, and Procedures

Enterprise Information Integration Server

Application Server

- Business objects
- Service objects
- Web services

Web Server

- Web objects

Matching Engine

Browser-based applications

Desktop applications
SHIP Architecture

[Diagram showing the architecture of SHIP with primary data sources, Web Server, Application Server, Database, Instruments, Files, Programs, Matching Engine, Case Base, and various other components connected through arrows and lines.]
SHIP Architecture
SHIP Architecture

Raw data ➔ Integrated information ➔ Analysis ➔ Knowledge ➔ Action
SHIP Screen Shot

- ISS: International Space Station
- PUI: Program Unique Identifier (part number)
- PRACA: Problem Report and Corrective Action
SHIP Summary

- **Problem Resolution and Corrective Action (PRACA)** system for manned space vehicles
  - Disparate data sources: live sensor feeds, archived legacy data, XML files, Word documents, web pages, web services, databases, etc.
  - Case-based reasoning and rules-based diagnoses
  - Semantics-based knowledge management

- **Service-oriented architecture (SOA)**
  - Web-based and desktop client applications
  - J2EE technologies
  - Web services
  - Enterprise information integration (EII)
  - Application integration
  - JavaServer Faces, BEA WebLogic, Composite Software
  - Open source software
Case Study 3: Shot Data Management
Case Study 3: Shot Data Management

The National Ignition Facility (NIF) is a major laser-based fusion energy research project at the Lawrence Livermore National Laboratory.

Each simultaneous firing of its 192 powerful lasers at a BB-sized target generates gigabytes of data that Shot Data Management routes in near real time to the project scientists and researchers.

- My role: Enterprise Software Strategist
  - Architect, catalyst, lead developer, and systems integrator, 2006-2007
  - High security clearance (Level P)
Shot Data Management Primary Services

- **Data provisioning**
  - Integration of disparate and heterogeneous data sources
  - Secure access and downloads
  - Metadata-based queries

- **Data management**
  - Version management and revision control
  - Archive data from NIF’s expected 30-year lifespan

- **Data marts**
  - Historical and trend analysis of specific subject areas
  - Support ad hoc data reporting

The target: A polished 2 mm capsule filled with cryogenic hydrogen fuel.
Requirements

- Application insulation
- Eliminate duplicate data
- Near real time data access
- Reusable data services
- Decision support
- Quality control
- Hierarchical storage management
- Data security
- Integrate legacy data silos
- Workflow management
Architecture Overview

- Industry standards
  - Web services
  - XML data sources
  - JDBC and ODBC (database interfaces)
  - Java Message Service (JMS)
  - BPEL (Business Process Execution Language)
- Java programming language
- SOA (Service-Oriented Architecture)
- Linux server clusters

- Oracle RDBMS
- Oracle CMS
  (Content Management System)
Workflow: Set Up Campaign & Approve Experiment

Execute Service Order

Set Up Campaign & Experiments

Approve Experiment

Generate service order

Create campaign

Export to ICCS

Generate Settings

Approved Settings

Diagnostic Configuration

Run Expert Group Tools

Generate service order

Create campaign

Export to ICCS

Generate Settings

Approved Settings

Diagnostic Configuration

Run Expert Group Tools

Generate service order

Create campaign

Export to ICCS

Generate Settings

Approved Settings

Diagnostic Configuration

Run Expert Group Tools
Workflow: Manage Configuration & Calibration

- Request calibration
- Calibrate Data
- Service Order
- Install
- Diagnostic Configuration
- Diagnostic Calibration
Workflow: Analyze & Visualize Shot Results
Data Archiving

- Hierarchical storage management system
Visualizing Workflows

- Large display of many nodes in multiple “swim lanes”
  - Each node represents a unit of work to be performed.
  - Each swim lane represents a category of work.
  - Results flow from one node to the next based on rules.

- The display is maintained in real time.
  - Currently active nodes are highlighted.
Creating a Workflow

- Create workflow diagrams using an IDE.
- Compile workflow diagrams into BPEL scripts.
  - Business Process Execution Language (XML)
Shot Data Management Summary

- Large, complex distributed system based on SOA
  - Integration of legacy data silos
    - data warehouses
    - relational databases
    - content managers
  - Data archiving
    - HSM
    - 30 years of data

- Workflows
  - Shot approval and setup
  - Configuration and calibration
  - Results analysis and visualization

“This system saved 6 months off the software development schedule.”
Important Lessons Learned

Sunset on Mars
Important Lessons

- Scientific data management systems share common features
  - Distributed and web-based
  - Integration of legacy subsystems
  - Integration of disparate and heterogeneous data
  - Near real-time data access with security
  - Data is unstructured
  - Data queries driven by metadata
  - Users collaborate to analyze data and write reports.
Important Lessons

- Keys to project success and high reliability
  - Well-understood requirements
  - Highly dedicated and competent developers

- Good software engineering
  - Solid but flexible service-oriented architecture (SOA)
  - Industry standards and best practices
  - Unit testing
  - Monitoring tools
  - Create alternate paths around risky areas of code that you can “hot switch” at run time.
Important Lessons

- Keys to project success and high reliability cont’d
  - Strong project leadership
    - Realistic project schedule with clear milestones
    - Agile (iterative) development methodology
    - Maintain constant communication with users
    - The leader must actively drive the project from milestone to milestone
  - Obsessive stress testing
    - Discover the system limits (before the users do)
    - Learn the system’s behavior under extreme stress.
Questions?
ALIEN SKULLS FOUND ON MARS!

TOP SECRET NASA REPORT: ‘ANCIENT ASTRONAUTS WERE OUR ANCESTORS’

PLUS

SADDAM’S PRISON LOVER TELLS ALL!
The Retrieve-Reuse-Revise-Retain Cycle

New fault

Retrieve

CASE BASE

Retain

Confirmed solution

Revise

Proposed solutions

Similar faults

Reuse

Old cases

New case

Fault Analysis System