



A Whale of a Tale: Creating Spacecraft Telemetry Data Analysis Products for the Deep Impact Mission



Kathryn Sturdevant
SpaceOps 2006
June 19-23, 2006

Whale Definition



- Whale- what does it mean?
 - Not an acronym, but a nickname indicative of the size of the task
- What is it? What does it do?
 - Whale is a scripted utility that automatically creates data products such as plots and tables for a given set of telemetry data, then sends an email link to the archive location of the products.

Whale Overview



- Initially envisioned as an archive utility for testbed data.
- Many users not familiar with the existing JPL proprietary ground data system: Deep Impact was a joint mission between JPL in Pasadena, California, and Ball Aerospace in Boulder, Colorado. Many users required assistance with creating data products.
- Whale became a product generation utility and the primary means of analyzing project data for the Deep Impact Mission.
- Was adapted to process both testbed data and operational data during primary mission.
- The project planned to use Whale for the 4 months prior to launch, however it was used an additional 7 months post launch, through end of primary mission.



Whale Challenges

- Network Configuration
 - Support Testbeds and Users at Ball Aerospace and at Jet Propulsion Laboratory (JPL).
- Whale Data Flow
 - What we did to cut processing time
- Skeleton Whale Team with Wide Range of Users
- Creeping Requirements
 - The define-as-you go approach
- Whale Tool Competition-- and another requirement
- CPU and Archival
- The Ripple Effect of More Data
- Data Validation
 - Issues that limited automation of the testbed Whale utility

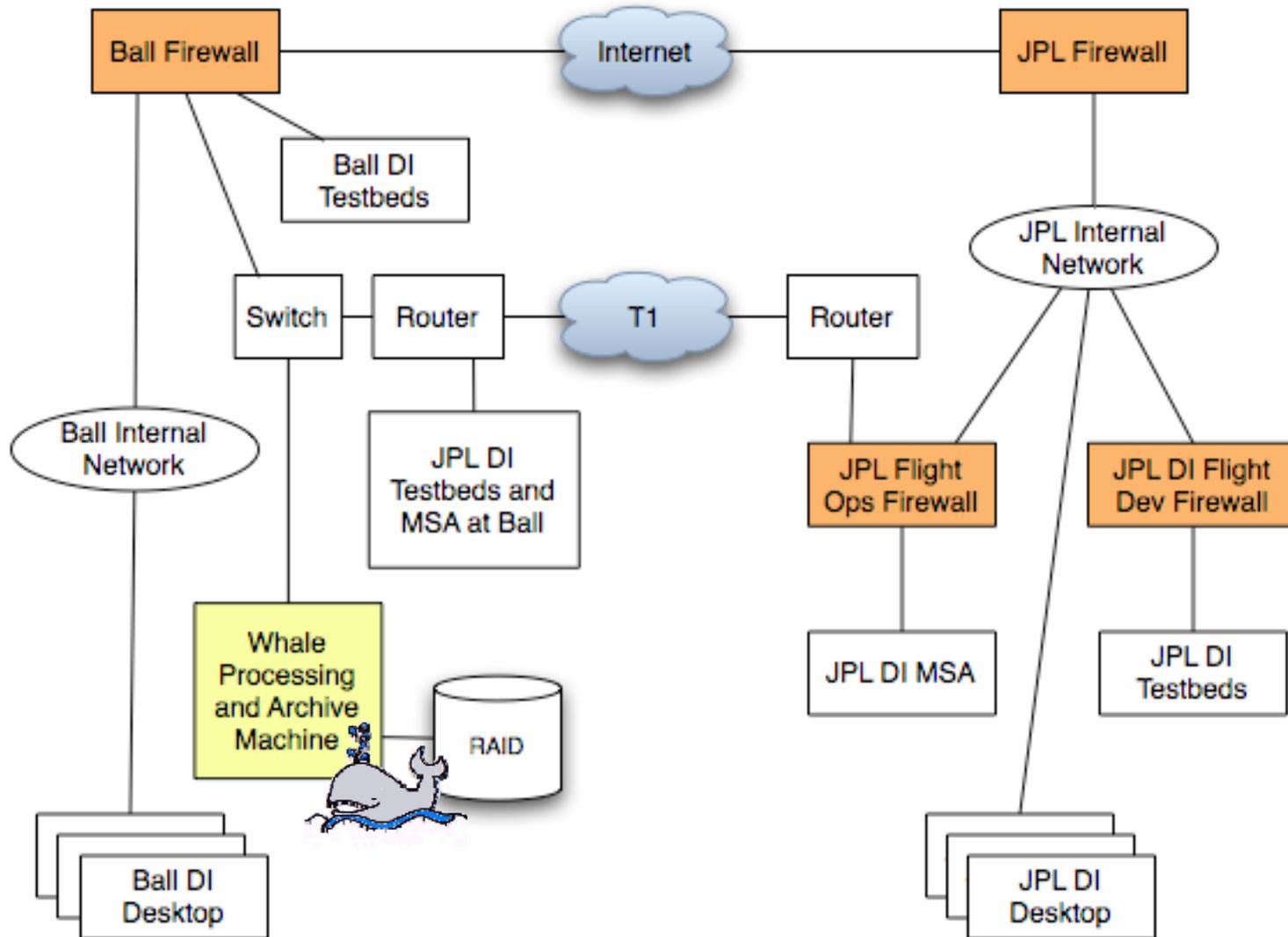


Network Configuration

- Network had to support data submissions to Whale from testbeds located at Ball Aerospace and at JPL.
- Network had to allow Users at Ball and at JPL to access to archived Whale data products for review.
- Firewall and Security issues
 - Whale machine made its own “island”
 - Limited access via Firewall to certain machines
 - Database queries across the network were restricted, so large files had to be transferred from testbed machines to the Whale processing machine.



Network Diagram



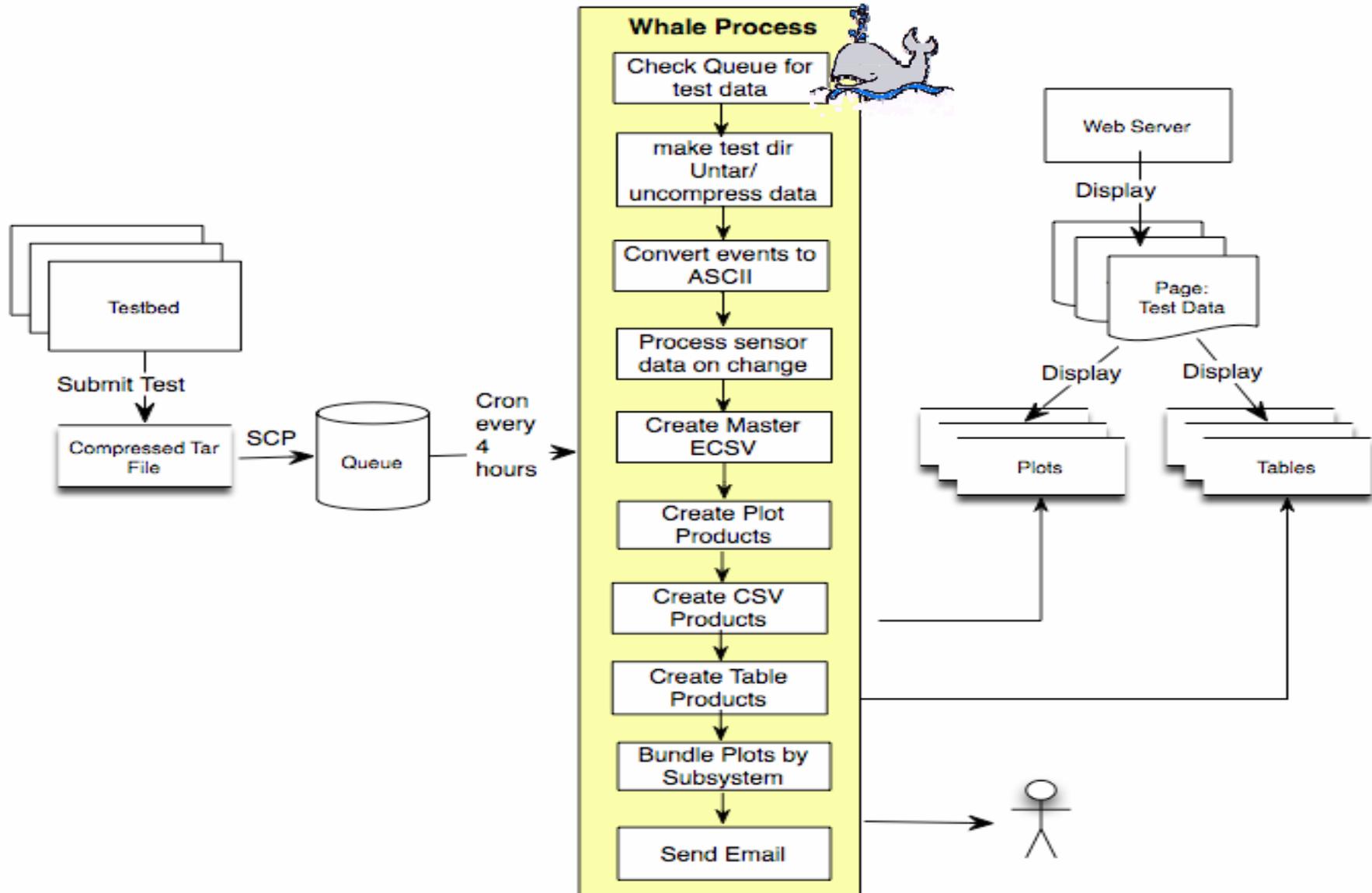


Whale Data Flow: Design Decisions

- Design decisions made to cut data processing time:
 - Process sensor values “on change” versus “on sample” to reduce the data to 1/10th of its previous size.
 - Parallel processing was used to collect multiple sensor values from the data file at once, thus reducing file read times. Collecting 250 values at a time to stay under the Unix/Perl 256 open file handle limit.
 - Reduce the sensor value data set list to only those values requested by the subsystems. Reduced list of sensor values from 21,710 to 3709.



Whale Data Flow Diagram





Whale Data Flow

The Testbed Whale Data Flow is the following:

1. The test conductor initiates the Whale submission script, providing a test identifier and data directory location.
2. The Whale submission script executes as setuid to the Whale team login, compress-tars the test data, and secure copies it to the queue on the Whale processing machine.
3. The Whale processing script runs via cron every 4 hours and checks the queue for test submissions.
4. The Whale processing script untars all queued tests and processes them one at a time, creating a subdirectory to be accessed via the web and installing the test data there. It then begins to process the data.
5. The event data is run through a script that converts it to ASCII text.
6. The sensor data is converted into an “expanded” comma separated value format (ECSV).
7. Plot products are generated by the plotting tool (gnuplot) using the ECSV and a list of channels.
8. CSV products are created from the ECSV in a parallel processing fashion (due to a 256 file limit) and are then used to generate table products.
9. The plots are concatenated into subsystem-specific groupings.
10. An email message notifies the Whale list that a test is available and a web link is provided.



Whale Team and Users

- Whale Team: Count = 2
 - Initially, one part time developer
 - Three months prior to launch
 - One full-time developer
 - Two half-time data management team members
 - Cruise Phase
 - One full-time developer
 - One full-time data management team member
- Whale Users: Count = approx. 40
 - JPL Users: most familiar with JPL ground data system (GDS), some not familiar.
 - Ball Users: Not familiar with JPL GDS



Creeping Requirements

- Whale primary focus changed from archival to data product generator. Requirements were sketched out but never fully explored.
- Whale became a high priority 4 months prior to launch as data analysis became critical. Project was in its critical pre-launch phase, thus making requirements gathering more challenging.
- Test conductors were understandably hesitant to change existing procedures, which could have been modified to improve to the data directory structures, and therefore the Whale processing.
- Manual test data collection methods caused inconsistencies in data given to Whale and sometimes required human data manipulation/filtering prior to Whale processing.



Whale Competition

- A subsystem engineer created a product and plot utility that gained a small, but vocal following.
- Whale was required to incorporate this functionality into its processing, thus requiring a second pass over the data using “on sample” methodology.
- On Sample is more time-consuming, however, because the list of the sample values was smaller than the full Whale set, the net effect was that it doubled the Whale processing time for a given data set.



Statistics: Processing Time

- Processing Time:
 - An 11 hour test with 13 MB of data took under 5 hours to process.
 - A 28 hour test with 381 MB of data from two sources (SCU_A and Impactor) took 26 hours to process.
- Testbed Data Rate:
 - Maximum data rate was 200 kilobits per second in the testbeds.



CPU and Disk Usage

- Testbed Whale Machine:
 - A planned, dedicated machine for data archive and processing.
 - Sun Ultra 2000 with dual CPUs
 - Initially, 2-880 GB RAIDs.
 - Final, 2.8 TB RAID to avoid running out of disk prior to impact.
- Operational Whale Machine:
 - A late addition to the Whale task, added to offload the severely overloaded testbed machine.
 - Sun Ultra 60
 - 2-539 GB RAIDs.
- Total disk usage was 3051 GB.



The Ripple Effect of More Data

- Processing demands grew significantly post-launch.
 - Test runtimes increased from 8 to 72 hours, with most in the 8-20 hour range.
 - Testbeds were often running 24/7.
- Multi-day tests were submitted to Whale in logical parts due to the size of the files and in order to get early products for analysis.
- Whale components broke as files grew > 2 GB and, and Whale had to be fixed quickly. For example, data was tarred and zipped. Zip failed on files > 2 GB, so Whale was modified to use tar with compress.
- The Netscape browser failed to display files > 2.1 GB.
- Whale staging area disk allocation was exceeded when 3 large testbed runs were queued simultaneously.

Data Validation and the Automation Challenge



- Automation Issues for Testbed Data:
 - No queries allowed over network so data collection had to be started/stopped by the test conductor. Data from a previous test could be inadvertently saved at the beginning of a new test.
 - Whale was designed to recognize set vs. unset clock values, not to determine that valid data was out of clock range for a given test.
 - Many conductors, manual procedures, operator fatigue.
 - Separation of SCU_A and SCU_B data was not automated.
- Automation of Operational Data:
 - Mostly automated once developed due to data format consistency-- data was queried and formatted prior to Whale processing.
 - Manual intervention required when flight software was upgraded and data had to be processed separately for each version of flight software.

Conclusions



- What we did right
 - Email notice
 - Reduced processing time
 - Automated “Daily Ops Whale”-- better automation because the data was collected by a program that queried a database.
- What we could do better
 - More requirements gathering early on
 - Make testbed data format more consistent
 - Have Whale format the data as much as reasonable/possible for consistency
 - Query the data to make cleaner data sets
 - Bundle data results by subsystem and email the bundles directly to each user.
 - Index tests by date in addition to testbed and name
 - Setup an archival process for DVD burning