Biotic Prediction

Building the Computational Technology Infrastructure
for Public Health and Environmental Forecasting

Test Plan

BP-TP-1.2

Task Agreement: GSFC-CT-1

December 3, 2003

1 Overview

1.1 Introduction

This project is developing the high-performance, computational technology infrastructure needed to analyze the past, present, and future geospatial distributions of living components of Earth environments. This involves moving a suite of key predictive, geostatistical biological models into a scalable, cost-effective cluster computing framework; collecting and integrating diverse Earth observational datasets for input into these models; and deploying this functionality as a Web-based service. The resulting infrastructure will be used in the ecological analysis and prediction of exotic species invasions. This new capability, known as the Invasive Species Forecasting System, will be deployed at the USGS Mid-continent Ecological Science Center and extended to other scientific communities through the USGS National Biological Information Infrastructure program.

1.2 Document Overview

This Test Plan has been prepared in accordance with NASA/GSFC's "Recommended Approach to Software Development Revision 3". The sections included are as follows:

Section 1 Overview - introducing the Invasive Species Forecasting System (ISFS) project and describes the sections of this Test Plan.
Section 2 Test Summary – summarizes the system, describes environmental prerequisites for successful tests and notes the requirements needing validation
Section 3 Test Procedures – goes over the objectives, guidelines & methods for testing
Section 4 Test Descriptions – illustrates how the tests are structured and explains each test case section.
Section 5 Input / Output – documents input & output applicable to the various tests.
Section 6 Test Cases – the series of tests to be executed on the system

1.3 Document Versions

Date	Version	Description
August 22, 2003	1.1	Initial submission to CT relating to Milestone F
Dec 3, 2003	1.2	Second submission to CT relating to Milestone F: • Variety of test scenarios presented • System prerequisites & description expanded • Model output documented

1.4 Referenced Documents

Document Title	Version	Date
Software Design (BP-SDD)	1.2	2003-12-02
Concept of Operations (BP-CONOP)	1.9	2002-12-04
Software Requirements (BP-SRD)	1.6	2003-11-30
Software Requirements Trace Matrix (BP-SRTM)	1.0	2003-11-30
Baseline Software Design (BP-BSD)	1.3	2002-11-25

2 Test Summary

2.1 System Summary

The current system handles communication through the firewall between the development server and the compute server, which is the interface with the Goddard cluster.

The subsystem can be broken into three functional layers - Front End, Application and Backend.

Front-end Layer

Consists of the web browser where the Graphical User Interface is presented and the user interacts with the application.

Application Layer

This layer consists of the web server, applications server. This layer is responsible for managing the communications to the backend layer. This layer also provides application & end-user metadata persistence. Web sessions are specific to each user logon. Details regarding the servers, application components & web services may be found in the design document. Some highlights important w/in the testing context:

- The development web server is CARBON. It physically resides at SSAI offices in the Aerospace building, Lanham. Planned: CARBON will be replaced by TAMARISK in December '03.
- The production web server is WEBSERV. It physically resides at NASA GSFC, building 28.
- The architecture lends itself to having web & application services be on separate servers, but currently they are configured on one server. So the web server doubles as an application server.
- Apache web service and Tomcat JSP/Servlet engine must be up-and-running in order for the application to function. No user-specific accounts are necessary, related to these components. The user must only have a valid login to the application.
- Postgres is the RDBMS where user account & application metadata reside. The JSP/Servlet routines access the database under one system-level account. This database access is not related to and is hidden from the user.

Backend Layer

Consists of the host to the Beowulf cluster, where the modeling is processed. Modeled & archived data stores will be housed in this layer, in releases subsequent to Milestone F.

- Statistical modeling & data manipulation programs are written in IDL, ENVI, C & Fortran. The compilers and runtime libraries for these languages must reside on the cluster. These programs are further described in the Software Design.
- The cluster currently used by the web application, as well as for model development & testing is MEDUSA, hosted on the FRIO node.
- Planned: the application is going to be reconfigured to use a recently provided, dedicated small cluster, PIVOT. This transfer should occur in Dec '03. PIVOT doesn't have a dedicated host, rather one of the nodes acts as the host. Model development & testing will continue to happen on FRIO, while application development and testing will use PIVOT.

Specifications on both clusters may be found in the Software Design document.

2.2 Environmental Prerequisites

To test the system, the following conditions must be met:

Front-end Layer

- Tester must have access to the Internet using a web browser.
- Internet Explorer 5.0+ and Netscape 6.0+ are supported. Default browser settings are recommended.

Application Layer

- User has a valid application login registered in the database
- Apache web service & Tomcat servlet engine must be running
- Postgres *'ISFS'* database instance must be running, populated with all application metadata. The Db is accessible to application via the user account *'postgres'*

- All libraries, drivers & frameworks must be installed & configured [see Setup/Deployment instructions in Design document]
- All custom modules, JSP installed [see Setup/Deployment instructions in Design document]
- Active connection to the target cluster must be available.

Backend Layer

- IDL, ENVI, C and Fortran77 must be installed on the cluster, along with IDL extension libraries (e.g. Astronomy IDL lib), modeling routines (e.g. stepreg) & data generation/preparation programs (e.g. varfuncs)
- Merged data set must exists, containing appropriately projected geographic coordinates
- Remote Imagery & GIS layer data files must be present, w/ the filenames matching field names in the merged dataset.
- Valid NCCS account, *'ISFS'*, must exist

2.3 Milestone F Requirements Trace

The Milestone F Requirements Summary section of the Software Requirements document (SRD) lists all the requirements targeted for release for Milestone F. The Requirements Trace Matrix (SRTM) extends these requirements, adding source, dependencies, verification methods & release targets for each. Please refer to these documents.

3 Test Procedure

3.1 Objectives

The overall objective of these tests is to ensure the ISFS is functioning correctly and that all requirements have been satisfied. Specifically, all test scenarios defined within this plan aim to validate one or more of the following criteria:

- Requirements related to the test case have been satisfied
- The application functions correctly, as expected and in an intuitive & helpful manner
- Results are expected and accurate
- Appropriate error & warning messages are displayed
- Useful and comprehensive logging occurs for debugging purposes
- Provide record of tests to ensure problems are fixed in subsequent releases
- Confirm standards have been implemented in a consistent fashion
- Measure Performance & track benchmark metrics
- Planned: Ensure stability & performance isn't compromised in subsequent releases via Regression testing.

3.2 Guidelines

All tests should be conducted independently of each other to verify the results of each test. Each test case will note all input requirements (dependencies) relevant to a successful test run, steps involved in the scenario, the expected results for each step & overall results that will evaluate to a Pass/Fail rating. All results upon execution of a test should be recorded on the test case form.

3.3 Verification Methods

The focus and nature of tests will vary, some targeting a specific component measuring its compute speed while others will focus on application workflow, structure, layout or computed result of the system. Different methods must be applied depending on the circumstances:

Verification Method	Description
Demonstration	The operation of the system, or a part of the system that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
Test	The operation of the system, or a part of the system, using instrumentation or other special test equipment to collect data for later analysis.
Analysis	The processing of accumulated data obtained from other qualification methods. Examples are reduction, interpolation, or extrapolation of test results.
Inspection	The visual examination of system components, documentation, etc.

4 Test Descriptions

Each test case is a standalone template, which provides test instructions and record-keeping for test executions. Each section is described below and presented in the top-to-bottom order they're placed on the test template.

4.1 Test Case

A brief, alphanumeric identifier for the test case.

4.2 Test Case Version

Each test case will have a version and published data. E.g. v1.0 release 1/12/04.

4.3 Title

This serves as a short but descriptive name which uniquely identifies a test. E.g. Application Run (Simple)

4.4 Tracking Information

This is data specific to an execution of a test -- application/Module release version, tester's Name, time & date the test was run must be recorded.

4.5 Purpose

Detailed description of the scenario, why its important to the testing suite & how does relate to other areas of the system. E.g. will verify the integrity of the system and validate the results of the metadata and output image files.

4.6 Input Requirements

Describes the dependencies that must be met in order for this test to succeed.

4.7 Expected Results

When executing a scenario with all input requirements satisfied, this is the predicted outcome. If the actual outcome matches the predicted, then the test has been a success.

4.8 Scenario

The order-dependent, series of steps the tester must follow. For each step the expected result will be provided. Each step will receive an Pass or Fail, depending on whether the actual result matched the expected. If a step fails due to a system bug, the Bugzilla bug number will be recorded.

4.9 Actual Results

The actual results of the a test should be recorded here. It may be as straightforward as "Results were as expected".

4.10 Evaluation Criteria

Further explanation may be necessary on how to interpret the results in order to accurately assess of the success or failure of the test.

5 Input/Output

5.1 Output Generated By Test Cases

Listed below is an explanation for each output produced by the series of test cases associated with this test plan. Not every test will produce output. Detailed description of this input & output may be found in current versions of the SRD & CONOP.

Logged Output from Modeling Routines

These modeling & data preparation programs are launched on the compute server by web application. This program provides the following services:

- Ensures required ENVI/IDL libraries & routines are compiled & loaded
- Performs simple parameter validation
- Loads & interprets merged dataset
- Outputs & logs routine messaging
- Calls routines in IDL/ENVI which run the distance matrix, modeling technique, variogram & spatial fitting. The Kriging program is also called, which is written in Fortran77.
- Generates geographical output

Map of Study Area

JPG & GeoTiff formatted images of the study area, thematically mapped with the predicted, modeled element.

5.2 Input Required By Test Cases

Listed below is the required output produced by the series of test cases associated with this test plan:

Merged Dataset

A Merged Dataset is the primary input to a model run. Its produced by concatenating the measured variable with remote sensed image data at each geographic coordinate in the measured field data. A Response field (i.e. measured field data), e.g. Total Plant Species [*tplant*] is concatenated with the predictor variables (i.e. remote sensed, image data), such as ETM bands or MODIS data. The data values are related via each measured geographic coordinate, in a common projection like UTM. The coordinates for each data point are included in the merged data set, e.g. *xutm* & *yutm*.

Augmented Merged Dataset

This is a merged dataset with added variables that have beeen calculated from the existing set, e.g. vegetation index [*NDVI*]. A model run can either take as input Merged or Augmented Merged data.

Remote sensed imagery data

Satellite imagery data will be primarily provided by external satellite data archives from sensors such as MODIS or LANDSAT, but also user-supplied satellite data or airborne imagery.

Map Mask

To display a map, such as the one found in test case 1, where the area outside the study site's boundaries are black, a mask file must be used. When the mask is combined with the thematic map resulting from the modeling process, all pixels outside the study site can be shaded one color to result in a nice presentation.

Glossary

BP Biotic Prediction project
BSD Baseline Software Design
CGFS Cerro Grande Fire Site in Los Alamos, NM
CT Computational Technologies project
CONOP Concept of Operations
COTS Commercial Off The Shelf
CSU Colorado State University
ENVI Environment for Visualizing Images
ESTO Earth Science Technology Office
GSFC Goddard Space Flight Center
GUI Graphical User Interface
IDL Interactive Data Language
ISFS Invasive Species Forecasting System
NCCS NASA Center for Computational Sciences
NREL Natural Resources Ecology Laboratory
RMNP Rocky Mountain National Park, CO
SEP Software Engineering / Development Plan
SRTM Software Requirements Trace Matrix
TP Test Plan
URL Uniform Resource Locator

6 Test Cases

Title: Application Run (elementary) Related Use Case: Run a pre-defined model

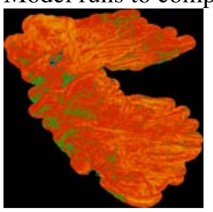
Test by:____Test Date/Time:___________________________

Build/Release:____

Purpose:
To run through the application, from login to output presentation, choosing all defaults. Provides a simple, baseline for the overall workflow & functioning of the application and all that the application depends.

Input Requirements:
1) Web browser access to http://carbon.sesda.com:8088/isfs/modelrun/logon.jsp
2) Application login

Expected Results:
1) User logs-in & runs through modeling app workflow successfully
2) Model runs to completion & displays the following output



Map	1069868804954.jpg	358.0 KB	+view	+download
Model Array	cerroGrande.ma	6.0 KB	+view	+download
Std. Output	1069868804954.txt	7.0 KB	+view	+download

3) Standard Output & Model Array samples may be found in Evaluation Criteria section.

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Direct browser to application login page	Login page displays, prompting user for Account Name and Passowrd. Modeling Workflow has no steps highlighted in Red, all are displayed in bold black. (figure 1)		
2	At login prompt, enter Account Name, Password & Submit	Account is successfully authenticated and the Study Site dropdown is presented. Select Study Site is shown in Red in the Modeling Workflow, all other steps are light grey (figure 2)		
3	Select 'Cerro Grande' Study Site & Submit	Screen is refreshed, showing Cerro Grande selection and presenting Data Set dropdown. Modeling Workflow displayed as shown in (figure 3), with Select Data Set in Red.		
4	Select 'CG_MA' Data Set & Submit	Screen is refreshed, preserving Study Site selection, showing CG_MA dataset selection and presenting Analysis Routine dropdown. Modeling Workflow displayed as shown in (figure 4), with Select Analysis Routine in Red.		
5	Select 'None' for Analysis Routine & Submit	Screen is refreshed, run selections are replaced with message "You Model Run is 0% done" and an image of the cluster computer. Modeling Workflow is displayed as shown in (figure 6), with Run Model in Red. After 15-30 seconds, the results page should automatically display, as seen in (figure 7), with a thumbnail of the map as well as map, model array &		

		standard output files that may be downloaded or viewed. Kriging will not run, so the Enter Parameters screen should not be presented.		
6	Select 'view' on the Map image	A full-size map as show in Expected Results section, matching the thumbnail, is displayed w/in the same browser window		
7	Select browser's Back function	Returned to Results screen		
8	Select 'view' on the Model Array	The Model Array (aka Merged Data Set), is presented w/in the same browser window (figure 8)		
9	Select browser's Back function	Returned to Results screen		
10	Select 'view' on the Std Output	The standard output that was captured when model routines were run on the cluster is presented in the browser (figure 9). Note: Kriging has not been performed & the related output section 'Perform the Kriging of the Residuals', should NOT exist in the std output file.		
11	Select browser's Back function	Returned to Results screen		
12	Select 'download' on the Map image	File Download dialog is displayed (figure 10). Note: the system generated JPG filename should be different for application runs. As well, the From value will change depending on what server you're running the application.		
13	Select 'Save'	File browser 'Save As' dialog should appear (figure 11)		
14	Select 'Save'	Image file should be saved at location indicated. Verify file has been downloaded.		
15	Repeat steps 12-14 for the Model Array & Std Output	Files should successfully download to selected location		
16	Select 'Start Over' button	Application should return to the Study Site dropdown selection screen (figure 2).		
17	Repeat steps 3 & 4, however choose Single Processor Kriging for Analysis Routine step, then Submit	Nearest Neighbors field is presented as shown in (figure 5).		
18	Enter 5 in Nearest Neighbor field and Submit	Kriging step will be run. Processing should take significantly longer than when run w/out Kriging. The std output section 'Perform the Kriging of the Residuals', should exist in the std output file.		
19	Take the resulting map, in GeoTiff format & redisplay in 3rd party products such as ESRI's ArcView or Corel Draw	Map should display & look the same as the JPG displayed in the browser.		

Actual Results:

Evaluation Criteria:

Model results are captured in the "Perform Stepwise Regression" section (see Figure 9)
Model diagnostics are output throughout the process, used for debugging & verification. See sections in Figure 9 such as the "Read the input file containing the merged field and RS data" and "Prepare for modeling steps".

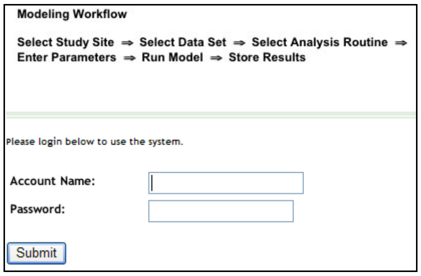


Figure 1 – Login

Figure 2 - Study Site

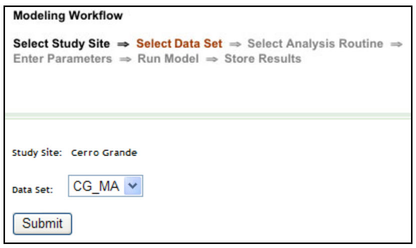


Figure 3 - Data Set

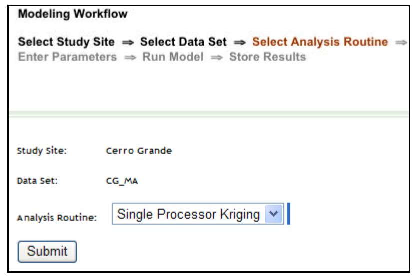


Figure 4 - Analysis Routine

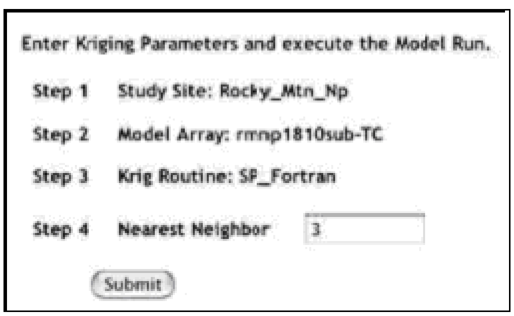
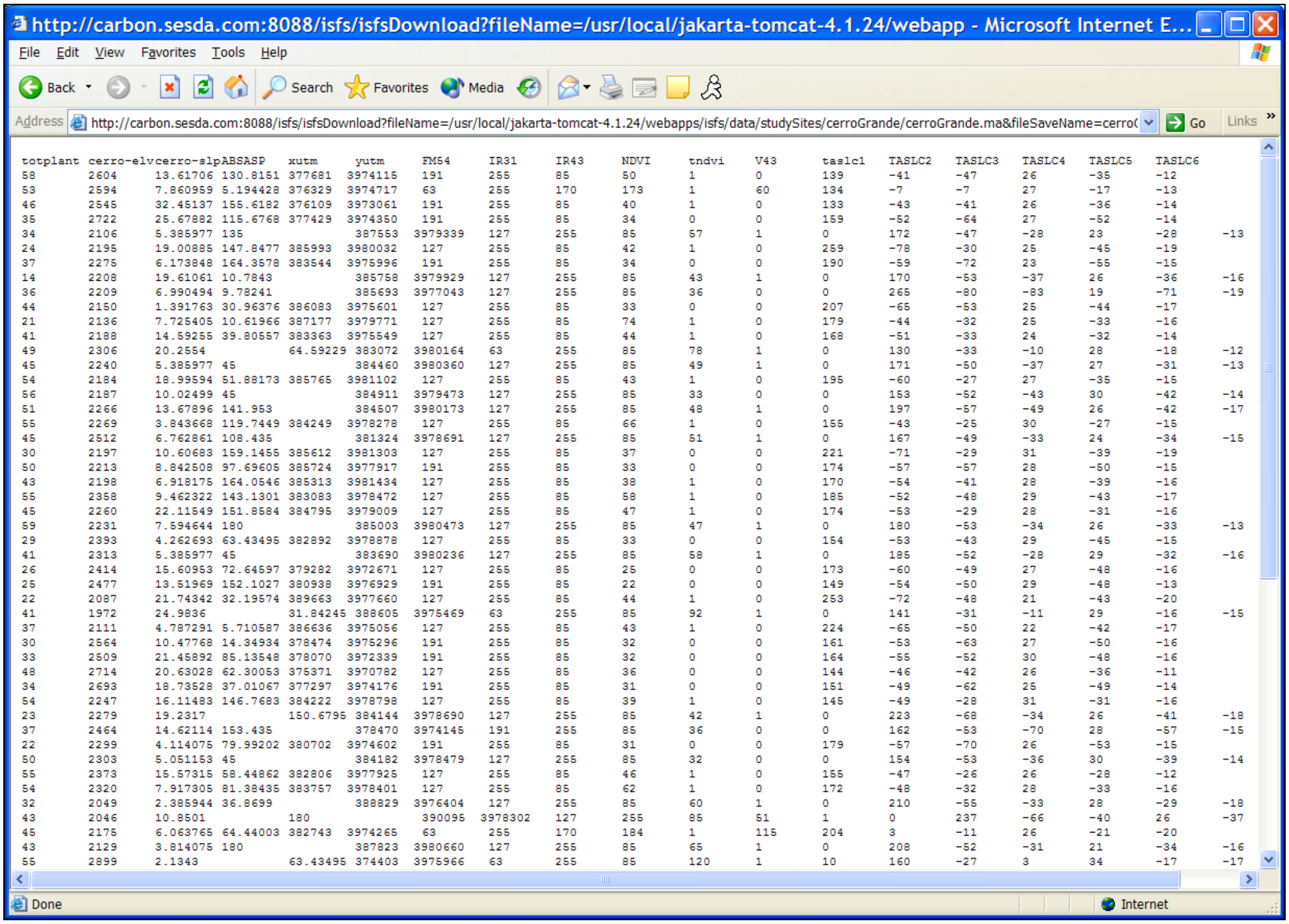


Figure 5 - Nearest Neighbor

Figure 6 - Model Run

Figure 7 - Results



totplant	cerro-elv	cerro-slp	ABSASP	xutm	yutm	FM54	IR31	IR43	NDVI	tndvi	V43	taslc1	TASLC2	TASLC3	TASLC4	TASLC5	TASLC6	
58	2604	13.61706	130.8151	377681	3974115	191	255	85	50	1	0	139	-41	-47	26	-35	-12	
53	2594	7.860959	5.194428	376329	3974717	63	255	170	173	1	60	134	-7	-7	27	-17	-13	
46	2545	32.45137	155.6182	376109	3973061	191	255	85	40	1	0	133	-43	-41	26	-36	-14	
35	2722	25.67882	115.6768	377429	3974350	191	255	85	34	0	0	159	-52	-64	27	-52	-14	
34	2106	5.385977	135		387553	3979339	127	255	85	57	1	0	172	-47	-28	23	-28	-13
24	2195	19.00885	147.8477	385993	3980032	127	255	85	42	1	0	259	-78	-30	25	-45	-19	
37	2275	6.173848	164.3578	383544	3975996	191	255	85	34	0	0	190	-59	-72	23	-55	-15	
14	2208	19.61061	10.7843		385758	3979929	127	255	85	43	1	0	170	-53	-37	26	-36	-16
36	2209	6.990494	9.78241		385693	3977043	127	255	85	36	0	0	265	-80	-83	19	-71	-19
44	2150	1.391763	30.96376	386083	3975601	127	255	85	33	0	0	207	-65	-53	25	-44	-17	
21	2136	7.725405	10.61966	387177	3979771	127	255	85	74	1	0	179	-44	-32	25	-33	-16	
41	2188	14.59255	39.80557	383363	3975549	127	255	85	44	1	0	168	-51	-33	24	-32	-14	
49	2306	20.2554		64.59229	383072	3980164	63	255	85	78	1	0	130	-33	-10	28	-18	-12
45	2240	5.385977	45		384460	3980360	127	255	85	49	1	0	171	-50	-37	27	-31	-13
54	2184	18.99594	51.88173	385765	3981102	127	255	85	43	1	0	195	-60	-27	27	-35	-15	
56	2187	10.02499	45		384911	3979473	127	255	85	33	0	0	153	-52	-43	30	-42	-14
51	2266	13.67896	141.953		384507	3980173	127	255	85	48	1	0	197	-57	-49	26	-42	-17
55	2269	3.843668	119.7449	384249	3978278	127	255	85	66	1	0	155	-43	-25	30	-27	-15	
45	2512	6.762861	108.435		381324	3978691	127	255	85	51	1	0	167	-49	-33	24	-34	-15
30	2197	10.60683	159.1455	385612	3981303	127	255	85	37	0	0	221	-71	-29	31	-39	-19	
50	2213	8.842508	97.69605	385724	3977917	191	255	85	33	0	0	174	-57	-57	28	-50	-15	
43	2198	6.918175	164.0546	385313	3981434	127	255	85	38	1	0	170	-54	-41	28	-39	-16	
55	2358	9.462322	143.1301	383083	3978472	127	255	85	58	1	0	185	-52	-48	29	-43	-17	
45	2260	22.11549	151.8584	384795	3979009	127	255	85	47	1	0	174	-53	-29	28	-31	-16	
59	2231	7.594644	180		385003	3980473	127	255	85	47	1	0	180	-53	-34	26	-33	-13
29	2393	4.262693	63.43495	382892	3978878	127	255	85	33	0	0	154	-53	-43	29	-45	-15	
41	2313	5.385977	45		383690	3980236	127	255	85	58	1	0	185	-52	-28	29	-32	-16
26	2414	15.60953	72.64597	379282	3972671	127	255	85	25	0	0	173	-60	-49	27	-48	-16	
25	2477	13.51969	152.1027	380938	3976929	191	255	85	22	0	0	149	-54	-50	29	-48	-13	
22	2087	21.74342	32.19574	389663	3977660	127	255	85	44	1	0	253	-72	-48	21	-43	-20	
41	1972	24.9836		31.84245	388605	3975469	63	255	85	92	1	0	141	-31	-11	29	-16	-15
37	2111	4.787291	5.710587	386636	3975056	127	255	85	43	1	0	224	-65	-50	22	-42	-17	
30	2564	10.47768	14.34934	378474	3975296	191	255	85	32	0	0	161	-53	-63	27	-50	-16	
33	2509	21.45892	85.13548	378070	3972339	191	255	85	32	0	0	164	-55	-52	30	-48	-16	
48	2714	20.63028	62.30053	375371	3970782	127	255	85	36	0	0	144	-46	-42	26	-36	-11	
34	2693	18.73528	37.01067	377297	3974176	191	255	85	31	0	0	151	-49	-62	25	-49	-14	
54	2247	16.11483	146.7683	384222	3978798	127	255	85	39	1	0	145	-49	-28	31	-31	-16	
23	2279	19.2317		150.6795	384144	3978690	127	255	85	42	1	0	223	-68	-34	26	-41	-18
37	2464	14.62114	153.435		378470	3974145	191	255	85	36	0	0	162	-53	-70	28	-57	-15
22	2299	4.114075	79.99202	380702	3974602	191	255	85	31	0	0	179	-57	-70	26	-53	-15	
50	2303	5.051153	45		384182	3978479	127	255	85	32	0	0	154	-53	-36	30	-39	-14
55	2373	15.57315	58.44862	382806	3977925	127	255	85	46	1	0	155	-47	-26	26	-28	-12	
54	2320	7.917305	81.38435	383757	3978401	127	255	85	62	1	0	172	-48	-32	28	-33	-16	
32	2049	2.385944	36.8699		388829	3976404	127	255	85	60	1	0	210	-55	-33	28	-29	-18
43	2046	10.8501		180		390095	3978302	127	255	85	51	1	0	237	-66	-40	26	-37
45	2175	6.063765	64.44003	382743	3974265	63	255	170	184	1	115	204	3	-11	26	-21	-20	
43	2129	3.814075	180		387823	3980660	127	255	85	65	1	0	208	-52	-31	21	-34	-16
55	2899	2.1343		63.43495	374403	3975966	63	255	85	120	1	10	160	-27	3	34	-17	-17

Figure 8 - Model Array (i.e. Merged Dataset)

```
The Output String generated for this run is:

Starting IDL/ENVI to invoke kriging for run #1069861585769


IDL Version 5.6 (linux x86 m32). (c) 2002, Research Systems, Inc.
Installation number: 10045.
Licensed for use by: NASA/GSFC

% Restored file: ENVI.
% Restored file: ENVI_M01.
% Restored file: ENVI_M02.
% Restored file: ENVI_M03.
% Restored file: ENVI_M04.
% Restored file: ENVI_M05.
% Restored file: ENVI_M06.
% Restored file: ENVI_M07.
% Restored file: ENVI_M08.
% Restored file: ENVI_D01.
% Restored file: ENVI_D02.
% Restored file: ENVI_D03.
% Restored file: ENVI_CW.
% Restored file: ENVI_IDL.
% Restored file: ENVI_IOU.
% Compiled module: GETINPUT.
% Compiled module: ISFS.
% Compiled module: DOVARIO.
% Compiled module: KRIG_GAUSS.
% Compiled module: KRIG_EXPON.
% Compiled module: KRIG_SPHERE.
% Compiled module: JPKRIG.
% Compiled module: BINARY.
% Compiled module: MAKE_KB2D_INPUT.
% Compiled module: MAKEKRIG.
% Compiled module: STAT_REGR_OUT.
% Compiled module: MY_STEPWISE.
% Compiled module: EXPF.
% Compiled module: GAUF.
% Compiled module: SPHERF.
% Compiled module: XML2STRUCT::INIT.
% Compiled module: XML2STRUCT::CHARACTERS.
% Compiled module: XML2STRUCT::STARTELEMENT.
% Compiled module: XML2STRUCT::ENDELEMENT.
% Compiled module: XML2STRUCT::GETSTRUCT.
% Compiled module: XML2STRUCT__DEFINE.
% Error opening file. File: getRunID
% Compiled module: GETRUNID.
% Loaded DLM: XMLSAX.
Opening input file ../data/input/1069861585769_params.xml
Unknown tag name: krigRoutineLabel
Unknown tag name: krigRoutineLabel
Unknown tag name: modelArray
Unknown tag name: modelArray
Unknown tag name: servletWrapper
Unknown tag name: servletWrapper
Unknown tag name: studySiteLabel
Unknown tag name: studySiteLabel
Unknown tag name: willWait
Unknown tag name: willWait
The input values for the Model run are:
** Structure ISFS, 6 tags, length=56, data length=52:
RUNID STRING '1069861585769'
DOKRIG INT 0
KRIGROUTINE STRING 'none'
NNEIGHBORS INT 0
STUDYSITE STRING 'cerroGrande'
FNAME STRING '../data/input/1069861585769.ma'
INPUTDIR STRING = '../data/input/cerroGrande/'
BASENAME STRING = ''

************************************************************
Read the input file containing the merged field and RS data
************************************************************
number of variables = 18
number of lines of data = 79
% Compiled module: STRPARSE.
variables:
column 0: variable = totplant
column 1: variable = cerro-elv
column 2: variable = cerro-slp
column 3: variable = ABSASP
column 4: variable = xutm
column 5: variable = yutm
column 6: variable = FM54
column 7: variable = IR31
column 8: variable = IR43
column 9: variable = NDVI
column 10: variable = tndvi
column 11: variable = V43
column 12: variable = taslc1
column 13: variable = TASLC2
column 14: variable = TASLC3
column 15: variable = TASLC4
column 16: variable = TASLC5
column 17: variable = TASLC6
location of xutm = 4
location of yutm = 5

************************************************************
Computing distance matrix
************************************************************
% Compiled module: W.
max before rescaling = 21803.5
min before rescaling = 81.0987
max after rescaling = 268.852
min after rescaling = 1.00000

************************************************************
Determine boundaries of study area
************************************************************
map: xl, xu, yl, yu = 370733.00 390263.00 3964357.00 3985777.00
field: xl, xu, yl, yu = 371192.00 390095.00 3966876.00 3981434.00
krigsize = 652 715

************************************************************
Prepare for modeling steps
************************************************************
elem = 1 2 3 6 7
8 9 10 11 12 13
14 15 16 17

************************************************************
Perform stepwise regression
************************************************************
% Compiled module: STDDEV.
% Compiled module: MOMENT.
% Compiled module: REGRESS.
% Compiled module: T_CVF.
% Compiled module: T_PDF.
% Compiled module: IBETA.
Final Statistics
Variables in the model:   i.d.Gl.: cerro-slp taslc1 tndvi cerro-elv
Var m R % S(m) Beta Temp Tprob
cerro-slp -3.41e-01 3.1 1.83e-01 -2.16e-01 1.866 96.7
taslc1 -1.29e-01 2.4 4.80e-02 -3.61e-01 2.680 99.5
tndvi 4.05e+00 2.1 2.59e+00 1.69e-01 1.565 93.9
cerro-elv -1.17e-02 0.9 6.53e-03 -2.44e-01 1.794 96.2

Const 94.852 R 14.1% F_emp 3.048
% Compiled module: F_PDF.
F_prob 97.8 J 4 DF 74 n 79

************************************************************
Calculate and plot the variogram of the residuals to the fit
************************************************************
min, max of residuals = -30.471047 30.185684

GAMV Version: 2.000

data file = ./residuals.dat
columns for X,Y,Z = 1 2 0
number of variables = 1
columns = 3
trimming limits = -1.00000002E+21 1.00000002E+21
output file = ./gamv.out
number of lags = 20
lag distance = 1000.
lag tolerance = 0.
number of directions = 1
azm, atol, bandwh = 0. 180. 9999.
dip, dtol, bandwd = 0. 90. 90.
flag to standardize sills = 0
number of variograms = 1
tail,head,type = 1 1 1

xltol is too small: resetting to xlag/2
```

```
Variable number 1
Number = 79
Average = 2.47566767E-08
Variance = 115.706978

Variogram 1 Semivariogram : tail=Residual head=Residual

GAMV Version: 2.000 Finished

initial fit guess = 15.4818 7762.91
Gaussian fit coefficients = 110.733 921.691
dokrg = 0
KRIGING WAS BYPASSED

******************************************
Perform the kriging of the residuals
******************************************
 Kriging routine = Reich Fortran
% Compiled module: MAKEKRIG.
percent done =       0
percent done =       5
percent done =      10
percent done =      15
percent done =      20
percent done =      25
percent done =      30
percent done =      35
percent done =      40
percent done =      45
percent done =      50
percent done =      55
percent done =      60
percent done =      65
percent done =      70
percent done =      75
percent done =      80
percent done =      85
percent done =      90
percent done =      95
percent done =     100
Done with kriging
ncols: 1186
nrows: 1041
xllcorner: 422594.750000
yllcorner: 4450614.500000
cellsize: 29.965000
    max is 34.078171 and min is -35.102051
    ncols: 1186
    nrows: 1041
    xllcorner: 422594.750000
    yllcorner: 4450614.500000
    cellsize: 29.965000
    max is 7.219118 and min is 0.737369
    % Restored file: ENVI_UTL.
***********************************************************
Apply the model to estimate total plants over the study area
***********************************************************
InEQ = 1 9 7 0
number of significant variables = 4
coefficients = -0.34146088
-0.12852968
4.0509230
-0.011723463
opening and reading image file cerro-slp
% Compiled module: READCOL.
% Compiled module: NUMLINES.
% READCOL: Format keyword not supplied - All columns assumed floating point
% Compiled module: GETTOK.
% Compiled module: REPCHR.
% READCOL: Skipping Line 1
% Compiled module: STRNUMBER.
% READCOL: 22 valid lines read
% Compiled module: MEAN.
% Compiled module: CURVEFIT.
opening and reading image file taslc1
opening and reading image file tndvi
opening and reading image file cerro-elv
Min and max of the derived total plant image = 0.00000 69.7406
% LOADCT: Loading table BLUE/GREEN/RED/YELLOW
% Loaded DLM: JPEG.
% Restored file: ENVI_UTL.
0: fid_name = ../data/input/1069861585769.ma
1: fid_name = ../data/input/1069861585769.ma
2: fid_name = ../data/input/1069861585769.ma
3: fid_name = ../data/input/1069861585769.ma
4: fid_name = ../data/input/1069861585769.ma
5: fid_name = ../data/input/1069861585769.ma
FINISHED
% Program caused arithmetic error: Floating divide by 0
% Program caused arithmetic error: Floating underflow
% Program caused arithmetic error: Floating illegal operand
```

Figure 9 - Standard Output from Model run w/ Kriging

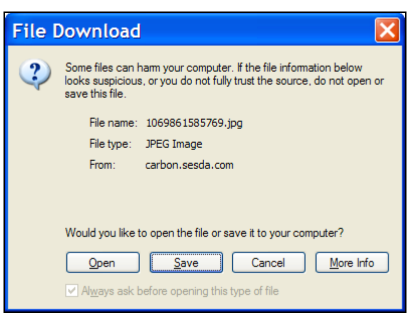


Figure 10 - Download Map

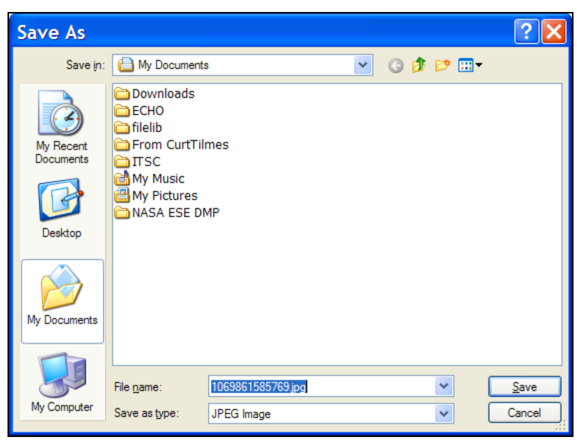


Figure 11 - Save File

Title: Running & Fitting the Model Related Use Case: Create new model for analysis

Test by:______________________________Test Date/Time:________________________

Build/Release:____

Purpose: Test the underlying modeling & spatial fitting routines on the computer server, independent of the web application. The goal of this test is to provide a high confidence level that the computational code that runs the models, spatial fitting & model array are working properly. These routines are used when building a new model and running a pre-existing model.

For the near-future, this test will need to be run by a member of the team knowledgable in the geostatistical algorithms used by the various modeling & analysis programs and adept at interpreting the results. Consequently, the scenario does not consist of explicit, step-by-step instructions, rather guidelines to follow with expected results. In the future a more user-friendly test scaffolding will be built and supporting documentation drafted around these routines, to allow less experienced team members to run these tests. This test case will be updated at that time as well, to reflect more specific & detailed steps.

Input Requirements:
1) NCCS Account on compute server
2) ENVI/IDL installed & configured on compute server
3) Access to & knowledge on how to use Proto_ISFS, varfuncs, stepreg, kriging programs
4) Properly formed merged dataset available
5) Remote sensed imagery data, located in same directory as programs are run
6) Mask of study area boundary for map visualization. So that the area of the map outside the study boundaries are displayed black.

Expected Results:
1) Model formula
2) Variogram plot
3) Variogram fit
4) Map of predicted variable over study area

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Logon to FRIO host node of the Medusa cluster	Access is granted		
2	Launch Proto_ISFS IDL/ENVI program, passing in parameter file containing the location+name of the merged dataset, columns where coordinates reside, listing of remote image bands	Merged & remote sensed data is read, all fields are identified, distance matrix & study area boundaries are calculated. Please refer to "Logged Output from Stepwise Regression with Kriging" section in the SDD for explanation of the standard program output.		
3	Program presents dialog to choose modeling technique (statistical method) to determine variable significance. Choose Stepwise Regression.	Regression is run and model formula is produced. See "Perform stepwise regression" portion of the "Logged Output from Stepwise Regression with Kriging" section in the SDD. See Figures 12 & 13 in Evaluation Criteria, related dialogs to this step.		
4	Set variogram parameters & run all variogram models [guasian, exponential & spherical] presented in the selection dialog. This will fit a model to the variogram. Choose one of the variogram models that "best fits" & continue.	Variograms are generated and display. See "Calculate & Plot the Variogram …" portion of the "Logged Output from Stepwise Regression with Kriging" section in the SDD. See also Figures 14 & 15 below which show the setup dialogs related to this step, as well as Figures 16-18 which show the variogram charts.		
5	Based on the results from the variogram, the user chooses (or not) to … run the Parallel Reich Kriging routine, selecting Nugget (non-zero value, e.g. 0.5), Range (e.g. 401 m), Sill (e.g. 27), # of Nearest Neighbors (e.g.10).	Kriging runs, produces an array of residual values that are applied to the model formula. See Figures 19-21 for dialogs related to this step.		

7	Apply the model to the pixels, multiplying by the coefficients and adding pixel-by-pixel, the residuals that came from Kriging.	Coefficients output should equal original coefficients presented when Regression was run.		
8	Present the geographical output	A map is presented, thematically shaded for the value predicted by the model. User provided opportunity to view all layers that were used in the model. See Figure 22.		

Actual Results:

Evaluation Criteria:

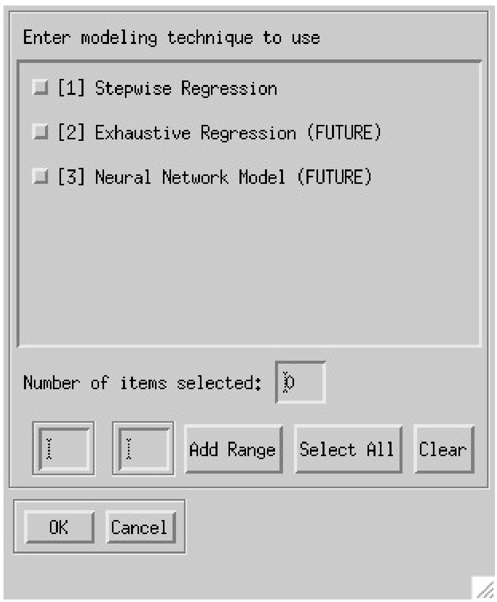


Figure 12 - Choose Modeling Technique

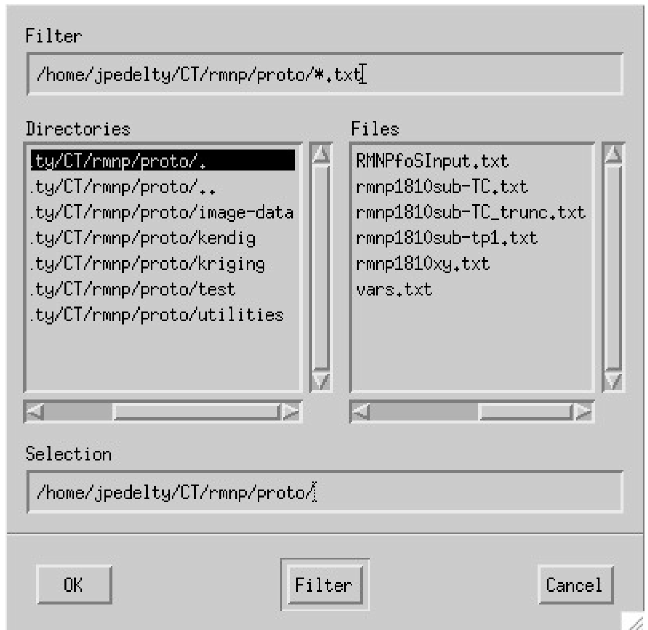


Figure 13 - Load Model Parameters

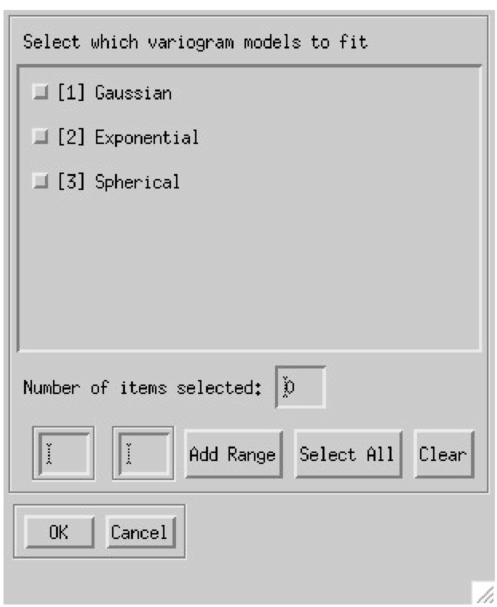


Figure 14 - Choose Variogram Type

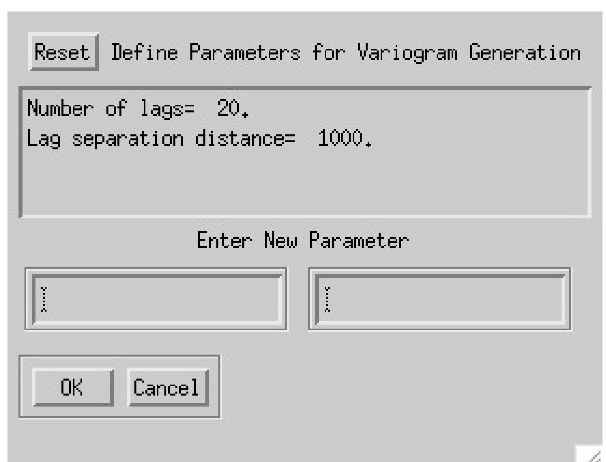


Figure 15 - Set Variogram Parameters

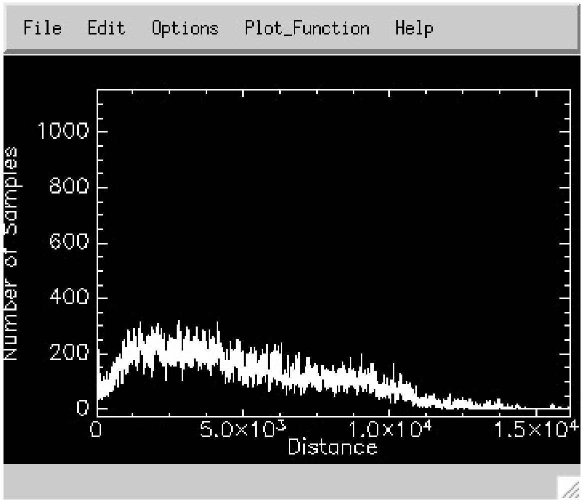


Figure 16 - Variogram Chart

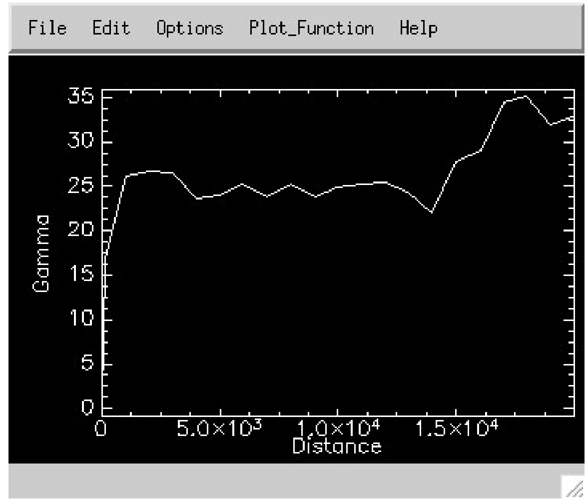


Figure 17- Variogram Chart

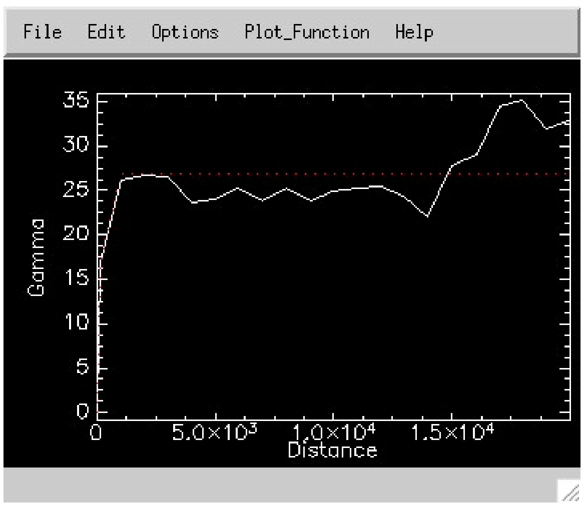


Figure 18- Variogram Chart

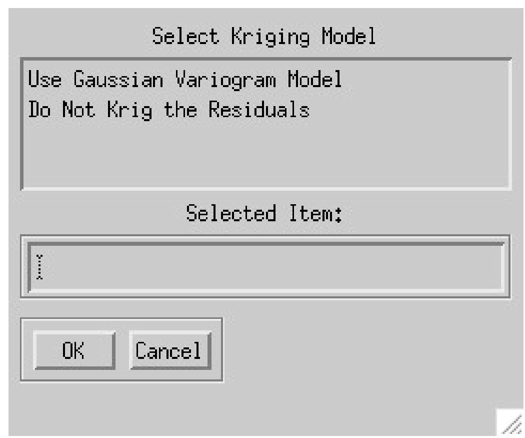


Figure 19 - Choose kriging

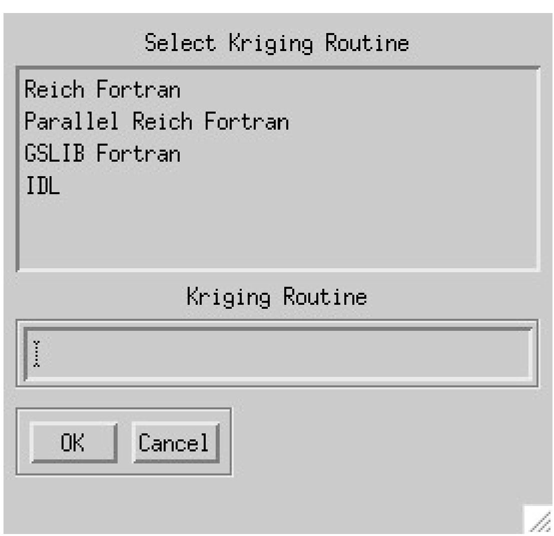


Figure 20 - Choose kriging Routine

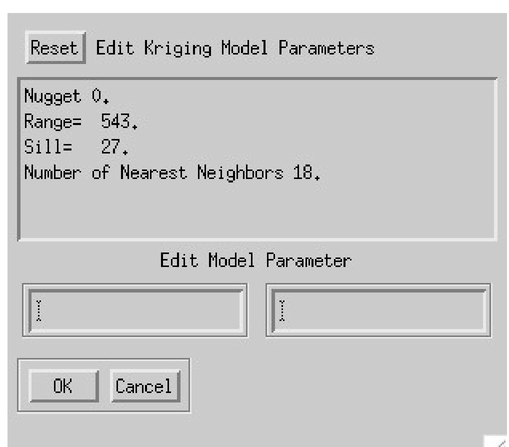


Figure 21 - Set kriging Parameters

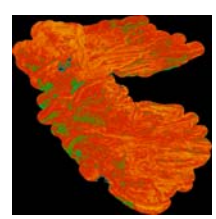


Figure 22 - Modeled Map

Title: User-supplied Data Ingest Related Use Case: Create a New Data File for Analysis

Test by:____Test Date/Time:____________________________

Build/Release:____

Purpose: This test one type of data ingest, user-provided, raster layers (typically image data or GIS "grids"). The design of this feature is not complete, consequently this test scenario reflects a functional overview and reasonable system behavior based on early design discussions & understanding of the requirements. These steps will evolve and become much more specific.

Design discussions involved in a workflow where:
1) The application will provide a screen for the user to "browse" to the location of the data file, type annotative text to be stored with the data file and initiate ingest.
2) The system uploads the data file to the back end archive file system via FTP
3) An entry is made in the relational database containing the user_id, the location+name of the data file and annotated notes. Probably ingest state info is logged as well, e.g. time of upload, file size.

Input Requirements:
1) Web browser access to http://carbon.sesda.com:8088/isfs/modelrun/logon.jsp
2) Application login
3) System accessible data file

Expected Results:
1) Data file resides on back end file system
2) Source data file is unaffected (e.g. file hasn't been moved, just copied)
3) RDBMS contains reference data to the file

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Login to application using user account with a Model Builder role	User successfully logs into the ISFS and is authorized as a Model Builder		
2	User chooses an option to upload, or ingest data to the ISFS	Browse dialog presented		
3	The user will select a file from their file system and choose to upload to the ISFS	The ISFS will upload the data via FTP and notify user when complete. Field presented to user to type annotative text to be stored with data file.		
4	The user submits annotative text with the file	The ISFS will ingest the data file to the relational database containing the user id, location + name of the data file and annotated notes		
5	The user receives a status stating the file has been ingested successfully and describes the name/location of the file	The log will show that the RDBMS contains reference data to the accessible file		

Actual Results:

Evaluation Criteria:

Title: Model Run Performance Related Use Case: Run a Pre-define Model

Test by:____Test Date/Time:_____________________________

Build/Release:____

Purpose: A primary objective for the Milestone F phase of the project was to improve the performance of the computationally-heavy statistical Kriging routines. The goal was a 25x speed up. By applying cluster computing and parallelizing the Kriging code, this performance improvement has been realized. The background, methods employed & results are documented in the BP-BSD document. This test case serves to validate and track these improvements as the system evolves and new releases are developed.

Currently, the code to run these test have not been stitched into a formal test framework – this will be done. But the code does reside with the science team and has been documented & validated by them. A knowledge transfer to the Engineering team will be scheduled, with the goal to incorporate the Kriging performance test code into a more user-friendly test tool/scaffolding. In conjunction with this effort, the test plan will be updated adding detailed instruction (expand on "*Run Kriging Routines*" step) on how to run the code and interpret the results.

The domain expert on the Kriging code, test methods & results is Jeff Pedelty.

Input Requirements:
1) Account access to Medusa cluster, where these test were performed
2) Run on Cerra Grande & Rocky Mtn study areas
3) "cerrotp.asc" input file (see figure 24) variogram method, sill, nugget, NN, variation figures per point. This file is copied to each node of the cluster before kriging program is run.
4) The 'doscaling' CShell script runs the kriging program 'krigmpi' several times, over 1, 2, 4, 8, 16 & 32 processors, for both Cerro Grande & Rocky Mtn study sites.

Expected Results:
1) Kriging runs successfully and outputs timing results, which should mirror those fond in the Evaluation Criteria section.
2) Kriged residuals files generated, named `krigtp+<#processors>`, e.g. 'krigtp16' {file generated when run over 16 processors}
3) Maps generated having had the kriged residuals applied to them, named se`tp+<#processors>`, e.g. 'setp8' {file generated when run over 8 processors}

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Logon to FRIO, host to MEDUSA cluster	Access granted		
2	Load & compile Kriging Program 'krigmpi'	Successful. May already be in compiled state.		
3	Run Kriging script 'doscaling'	Run in expected time, files & maps generated.		
4	Compare maps using CMP command	All generated maps should be identical.		

Actual Results:

Evaluation Criteria:
Performing a 'CMP' [bit-for-bit comparison] between 2 maps generated from two separate DoScaling steps, e.g. one from a 2 processor run & another from a 16 processor run, should yield an exact match – no differences should be reported between the two files.

The overall goal for Kriging code improvement is to reduce processing times and increase the amount of data handled by the model. Performance results from running the improved, parallel Kriging code on the Medusa cluster are shown in the table below. Detailed explanations may be found in the Baseline Software Design doc, Performance Improvement Plan, where these results were taken. Shown in Table 12 are the performance results (in seconds) for both the CGFS and RMNP baseline scenarios, as explained In the Baseline Scenario section of the BSD. These times should remain consistent as long as they're run on the Medusa cluster, using the same datasets and Kriging test code.

Number of Medusa Processors	Baseline Scenario			
	CGFS (715 rows, 652 cols)		RMNP (1041 rows, 1186 cols)	
32	127.5	98.0%	17.6	78.6%
16	255.9	97.7%	31.8	87.0%
8	508.2	98.4%	59.2	93.5%
4	1009.1	99.1%	113.0	98.0%
2	2016.0	99.2%	223.9	98.9%
1	4000.4	100%	442.8	100%

Figure 23 - Baseline Scenario Timing and Scaling Results

```
715
652
30
370733
3964357
79
79
5.12282390436951
455.410188253038
125.827632624413
gau
FALSE
377681   3974115   12.1404620252437
376329   3974717   4.41509570826466
376109   3973061   5.10877968928783
377429   3974350   1.2639663739676
387553   3979339   -16.2669362380078
385993   3980032   -9.38978806199168
383544   3975996   -4.65218666136289
385758   3979929   -30.4710466455864
385693   3977043   3.49264307347079
386083   3975601   1.43448996167306
387177   3979771   -27.2167014709079
383363   3975549   -5.67604644068282
383072   3980164   0.756836302466806
384460   3980360   -3.82452183411592
385765   3981102   12.2509463605737
384911   3979473   9.87556491557819
384507   3980173   8.65378903517916
384249   3978278   3.93234560863192
381324   3978691   -0.679706502860922
385612   3981303   -7.06850693099133
385724   3977917   6.47572679961798
385313   3981434   -5.92225129111595
383083   3978472   10.7501746344824
384795   3979009   2.50801067298626
385003   3980473   11.9809074498098
382892   3978878   -16.5484709729914
383690   3980236   -5.16929356014792
379282   3972671   -12.9857134760107
380938   3976929   -17.0454461010789
389663   3977660   -12.4933514719016
388605   3975469   -8.13047867517827
386636   3975056   -6.7292036520593
378474   3975296   -10.5218761006579
378070   3972339   -4.03141369115698
375371   3970782   10.5183546312577
377297   3974176   -3.47519873608697
384222   3978798   5.57925442701092
384144   3978690   -13.955990901807
378470   3974145   -3.15086325907611
380702   3974602   -21.4879818706616
384182   3978479   3.66564557672414
382806   3977925   9.15674502438412
383757   3978401   7.1062343875608
388829   3976404   -15.0754398875648
390095   3978302   2.24986910506317
382743   3974265   -0.113629574814838
387823   3980660   -2.90697130930975
374403   3975966   11.3770935503781
374563   3976570   -6.20833089650938
371335   3969606   -0.0960185622577104
385931   3977403   6.19391663906016
386706   3978244   0.476113505554024
387105   3978316   4.62085243907425
385700   3977459   4.49503881690016
385615   3977936   12.5392945511707
381247   3974853   7.92417388379023
383525   3977873   -2.38539061106023
383187   3977834   -0.0179980501900339
385162   3979529   30.1856843348419
376624   3975024   -9.68298039335884
376386   3973371   1.04605093170771
375040   3970477   -8.35053159818662
385461   3976940   5.30386267588058
387364   3976995   2.98076337868692
385911   3975803   8.92712449216745
385975   3978468   4.86294939840116
385231   3979175   22.5720691978511
385498   3979045   4.3828770769065
385579   3979049   7.66727259353406
385366   3980513   5.45320036927015
385439   3979279   12.0416462963239
386598   3979786   10.2602409070779
375674   3971833   9.92202988722483
374093   3970471   15.0443136085833
376906   3971368   12.8012190993337
379233   3975130   -21.9087068722515
373043   3966876   -4.10587978725426
371192   3967436   -2.59982267366625
373297   3973086   -14.048954634091
```

Figure 24 - 'cerrotp.asc' input file to kriging test

Title: Application Protocol with Cluster Related Use Case:

Test by:______________________________Test Date/Time:______________________

Build/Release:____

Purpose: The Application interacts with the compute server (host node to the cluster), via a secure internet protocol. Currently SSH is being used, but a move to RMI or SOAP will be forthcoming. This will preserve the secure nature of the interaction but also provide asynchronous communication between the two systems.

Code exists to validate the exchange of information. The package is "rmiComputeServer" and is used to confirm the RMI transport layer is working properly. Today, this code is targeted to be setup & run by a Java developer. An effort will be made to encapsulate the code into a tool that a non-developer could run.

Input Requirements:
1) "rmiComputeServer" code
2) Access to file system on both Compute & Application/Web servers
3) Java compiler

Expected Results:
1) Routines compile
2) Server & client launch successfully.

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Place the code on both the client and compute server machines. Run: > mkdir isfs; cd isfs; scp -r carbon.sesda.com:isfs (or unpack tarball)	Code created in appropriate dir structure. File content & structure should mirror listing in Evaluation Criteria.		
2	Edit the engine/ComputeEngine.java and put in the correct host name. Then compile and build stubs > javac engine/ComputeEngine.java > rmic -d . engine.ComputeEngine	Java code compiles		
3	Make sure you don't have a $CLASSPATH defined, then start rmiregisty: > rmiregistry	May experience problem starting the server if this step is skipped.		
4	Set $CLASSPATH: > setenv CLASSPATH /.../isfs/src:/.../isfs/src/compute.jar	Variable set in environment		
5	Start server (on carbon.sesda.com): > java -Djava.security.policy=isfs/java.policy -Djava.rmi.server.hostname=carbon.sesda.com -Djava.rmi.server.codebase=file://home/dkendig/isfs/src engine.ComputeEngine	Server starts		
6	Start client (on gcmd.sesda.com): > java -Djava.security.policy=isfs/java.policy client.ComputePi carbon.sesda.com 20	Client starts		

Actual Results:

Evaluation Criteria:

If client fails to start, possibly with error message "…connection refused…", necessary ports in the firewall or the network connection to the server are likely causes.

```
Directory of C:\rmiComputeServer\isfs
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
05/23/2003  09:49 AM               913 compute.jar
10/29/2003  02:18 PM               160 java.policy
10/29/2003  03:43 PM             1,245 Notes.txt
10/29/2003  03:42 PM             1,212 Notes.txt~
12/01/2003  09:04 AM    <DIR>          rmi
12/01/2003  09:04 AM    <DIR>          src
               4 File(s)          3,530 bytes

 Directory of C:\rmiComputeServer\isfs\rmi
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
12/01/2003  09:04 AM    <DIR>          com
12/01/2003  09:04 AM    <DIR>          META-INF
06/11/2003  12:38 PM            18,337 rmicb-1.2.1.jar
               1 File(s)         18,337 bytes

 Directory of C:\rmiComputeServer\isfs\rmi\com
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
12/01/2003  09:04 AM    <DIR>          css
               0 File(s)              0 bytes

 Directory of C:\rmiComputeServer\isfs\rmi\com\css
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
12/01/2003  09:04 AM    <DIR>          rmi
               0 File(s)              0 bytes

 Directory of C:\rmiComputeServer\isfs\rmi\com\css\rmi
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
06/11/2003  01:50 PM             6,073 ClientTwoWaySocketFactory.java
06/11/2003  01:50 PM             2,450 EndpointInfo.java
06/11/2003  01:50 PM               300 Makefile
06/11/2003  01:50 PM             8,070 ServerTwoWaySocketFactory.java
06/11/2003  01:50 PM             4,038 SignallingChannel.java
06/11/2003  01:50 PM             1,943 SocketAdapter.java
06/11/2003  01:50 PM             1,371 SocketPool.java
12/01/2003  09:04 AM    <DIR>          test
06/11/2003  01:50 PM             2,830 TwoWay.java
               8 File(s)         27,075 bytes

 Directory of C:\rmiComputeServer\isfs\rmi\com\css\rmi\test
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
06/11/2003  01:50 PM             2,316 Client.java
06/11/2003  01:50 PM               918 Hello.java
06/11/2003  01:50 PM               915 Hello2.java
06/11/2003  01:50 PM             1,432 Hello2Impl.java
06/11/2003  01:50 PM             1,423 HelloImpl.java
06/11/2003  01:50 PM               773 Makefile
06/11/2003  01:50 PM               832 Sender.java
06/11/2003  01:50 PM               834 Sender2.java
06/11/2003  01:50 PM               974 Sender2Impl.java
06/11/2003  01:50 PM               968 SenderImpl.java
06/11/2003  01:50 PM             2,320 Server.java
              11 File(s)         13,705 bytes

 Directory of C:\rmiComputeServer\isfs\rmi\META-INF
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
06/11/2003  01:50 PM                62 MANIFEST.MF
               1 File(s)             62 bytes

 Directory of C:\rmiComputeServer\isfs\src
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
12/01/2003  09:04 AM    <DIR>          client
12/01/2003  09:04 AM    <DIR>          compute
12/01/2003  09:04 AM    <DIR>          engine
               0 File(s)              0 bytes

 Directory of C:\rmiComputeServer\isfs\src\client
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
10/29/2003  11:41 AM             1,406 ComputePi.class
05/30/2003  02:41 PM               819 ComputePi.java
06/11/2003  06:32 PM             1,469 ComputePi.java.2way
05/30/2003  03:03 PM             1,485 Pi.class
05/30/2003  02:41 PM             2,984 Pi.java
05/23/2003  09:30 AM             2,913 Pi.java~
               6 File(s)         11,076 bytes

 Directory of C:\rmiComputeServer\isfs\src\compute
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
05/23/2003  09:48 AM               238 Compute.class
05/23/2003  08:22 AM               183 Compute.java
05/23/2003  09:48 AM               166 Task.class
05/23/2003  08:23 AM               124 Task.java
               4 File(s)            711 bytes

 Directory of C:\rmiComputeServer\isfs\src\engine
12/01/2003  09:04 AM    <DIR>          .
12/01/2003  09:04 AM    <DIR>          ..
05/29/2003  11:52 AM             1,366 ComputeEngine.class
05/29/2003  11:51 AM               911 ComputeEngine.java
05/23/2003  08:48 AM               899 ComputeEngine.java~
05/29/2003  11:54 AM             1,764 ComputeEngine_Skel.class
05/29/2003  11:54 AM             3,284 ComputeEngine_Stub.class
               5 File(s)          8,224 bytes
```

Test by:______________________________Test Date/Time:____________________

Build/Release:____

Purpose: No plans exist to be portable to other platforms such as SGI, Solaris. This test simply serves to as a visual confirmation that the correct version of Linux is installed on the servers and that the application runs correctly. Note: licensing may need to be validated in the future with the expected changes in the industry.

Input Requirements:
1) File System access to web server
2) File System access (NCCS account) to Compute
3) Web browser access to http://carbon.sesda.com:8088/isfs/modelrun/logon.jsp
4) Application login

Expected Results:
1) Redhat Linux 7.2+ installed on both Web & Compute servers
2) Web application runs and model produces valid results.

Scenario:

Step	Description	Expected Outcome	Pass/Fail	Bug Tracking
1	Log onto FRIO: > ssh -l <username> frio.gsfc.nasa.gov	System authenticates user and prompts: Last login: Thu Dec 4 12:38:10 2003 from <site>		
2	Query system for Linux Kernel version: > uname -a	System returns: Linux frio 2.4.9-31smp #1 SMP Tue Feb 26 05:55:20 EST 2002 i686 unknown		
3	Validate version of Redhat Linux installed: > cat /etc/redhat-release	System confirms: Red Hat Linux release 7.2 (Enigma)		
4	Repeat steps 1-3 on PIVOT, WEBSERV & CARBON servers	Results match		

Actual Results:

Evaluation Criteria: