

Configuration Server Overview

Shravan Pargal

ABSTRACT: *As a site adds Cray mainframes or mainframe related hardware to its machine room, the task of accurately describing and managing its configuration becomes challenging. Relationships between connected and shared pieces of hardware need to be described and clearly defined. With the addition of SCX rings and the T3E class of mainframes, this task could have become even more difficult.*

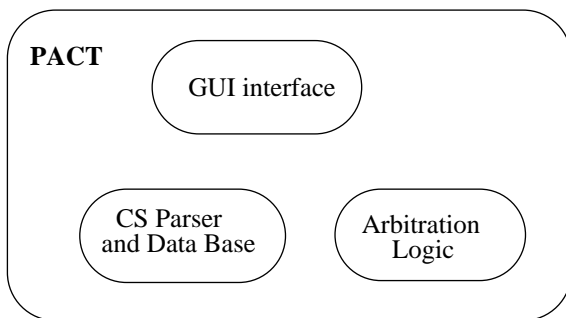
This is a work in progress paper describing the design and user interface planned for future site configuration management. It includes a description of a graphical user interface to define the site's configuration and a discussion on how configuration management will occur from the operating system (UNICOS/mk) perspective.

Introduction

The following paper describes the Configuration Server software on the Unicos/Mk operating system (OS). The Unicos/Mk Configuration Server was written to provide a single point of access for kernel(system) configuration information. The distributed nature of Unicos/Mk denies system servers direct access to each other's memory. As a result, common or shared configuration information needs to be kept at a single well known location, accessible to these servers.

Hardware Platforms

The Configuration Server software runs on two platforms. One piece runs on the SWS (Service Work Station) and the other runs on the T3E running Unicos/Mk.



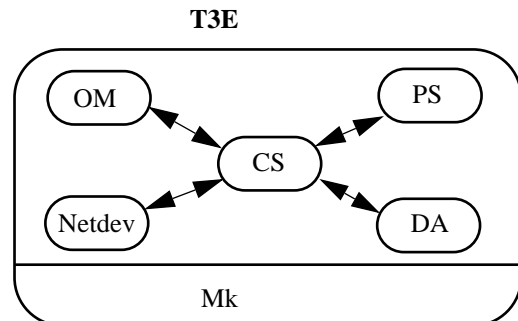
The software that runs on the SWS is controlled via a visual interface tool called PACT (Parameter Configuration Tool). Configuration information may be accessed via PACT or through a programming interface.

Copyright © Cray Research Inc. All rights reserved.

PACT is started as part of the SWS boot process and points to an initial configuration. PACT can be used to display and edit the configuration prior to a mainframe boot. Once the mainframe has been booted, the configuration displayed via PACT is locked in read-only mode.

The programming interface allows clients who need access to configuration information via programs running on the SWS. The access is provided by requiring these client programs to load in a client library which has in it the prototype for a function call to the Configuration Server.

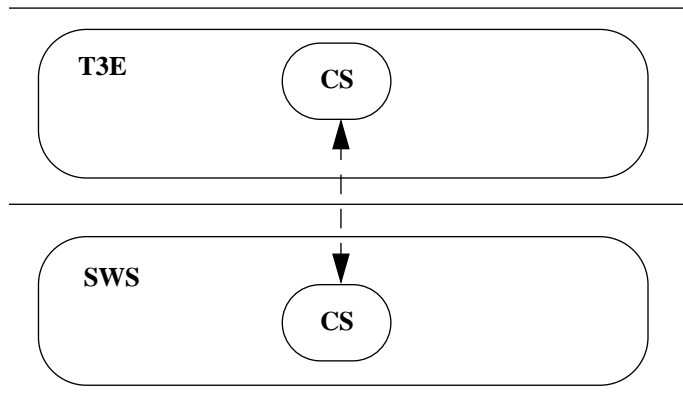
PACT controls the configuration data on the SWS, so requests made via the programming interface are arbitrated through PACT.



The other platform where the configuration server runs is the T3E running Unicos/Mk. The Configuration Server is one of the system servers started at system boot time. Configuration information is made available to other system servers via a library call.

Once a system is up and running, configuration changes can be forwarded from the T3E to the SWS. This allows changes in configuration to be persistent over mainframe reboots. The

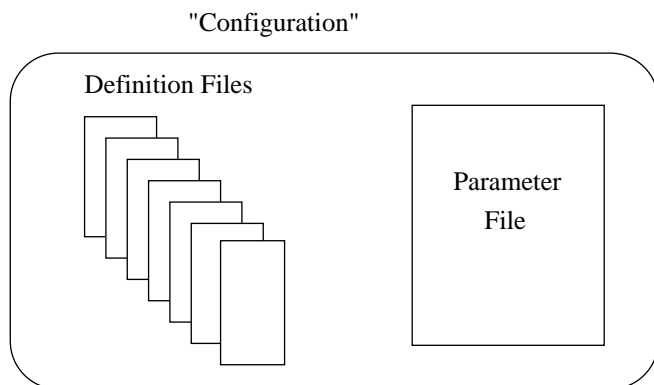
mode of communication between the T3E and the SWS is not described in this document(TBD).



What is a "Configuration"

The description of hardware and software in a machine room (site) is described by a "Configuration". A configuration consists of hardware and software components.

A "configuration" is made up of "definition files" and a "parameter file".



The definition files are header files which describe data structures used by system servers to communicate with the Configuration Server. Each server has one or more of these header files included in its list of header files. The normal case is that each server that needs to communicate with the Configuration Server only needs to include the file that described data structures of interest to this server. For example, the disk server includes a single disk definition file in its source.

The parameter file contains in it instantiations of a data structure defined in the definition file. It also describes the initial values of the data in this data structure.

For example one of the header files could have the following data structure:

```
typedef struct {
    int    type;
    char  *name;
} disk_t;
```

the parameter file would then have the following instantiation and initialization:

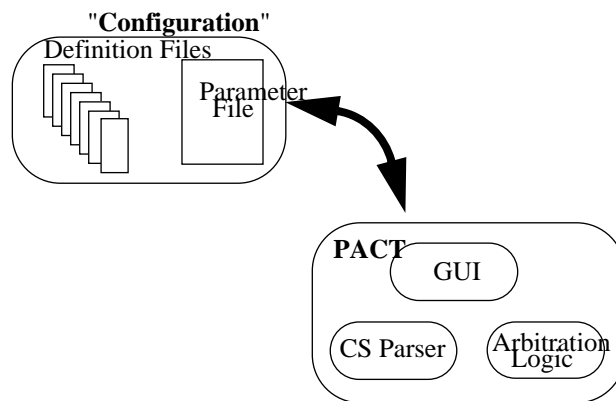
```
disk_t  my_disk;
my_disk.type = 1;
my_disk.name = "Tom"; /* note that is a "C"
like initialization, but not valid "C" code */
```

One way of visualizing the relationship between the definition files and the param file would be to think of the definition files to contain the building blocks, while the parameter file contains the combination of these blocks to form a structure. Together they describe a complete structure.

Configuration Processing

An important feature of the Configuration Server lies in the form in which it is driven. The Configuration Server interprets its input. This means that the configuration can be changed, new definition files can be added to the configuration and the Configuration Server does not need to be recompiled. There is a decoupling between the data and the data processing software.

On the SWS, PACT calls the Configuration Server parser to initially process the configuration passed to it. Since the Configuration Server software is independent of the input, PACT too is made independent of input provided to it. This implies that once the Configuration Server and PACT are developed they do not need to be recompiled to match the new input.



On the T3E, the Configuration Server parser code is run as part of the Configuration Server initialization logic. Once the configuration is loaded, the Configuration Server creates its request port and is in a position to process configuration information requests from other servers.

Storage of configuration information

The storage of configuration information can be visualized as an inverted tree with several branches, as shown in Figure 3.

Summary

The following list summarizes the functions of the Configuration Server:

- The mainframe configuration server is a single point of access for configuration information on the mainframe.
- Use of this configuration server provides consistent information.

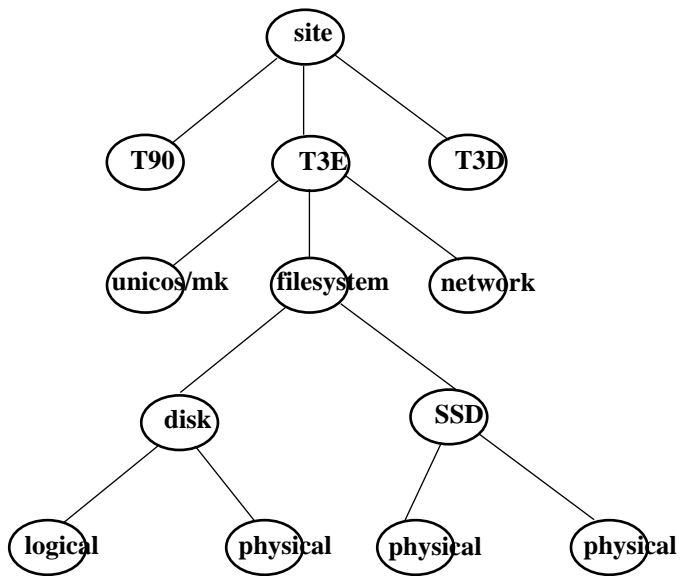


Figure 3: Storage of configuration information

- Because the configuration server has access to all configuration information, it can coordinate and inform other servers of configuration changes.
- The configuration server provides the ability to maintain configuration information between system restarts by communicating with the Boot Machine Configuration Server.
- The utility program and Boot Machine Configuration Server handle configuration issues on the boot system. Programs such as bootsys, dumpsys and sysclear use the Boot Machine Configuration Server request/reply interface to get the pieces of the configuration picture.