Bcfg2 and XML

Narayan Desai
Argonne National Laboratory
desai@mcs.anl.gov

Overview

- Bcfg2 concepts
- Bcfg2 model
- Status
- Failures
- Research directions
- XML
- Desired functionality

Bcfg2 Overview

- Vehicle for Configuration Management Research
- Client/Server Architecture
- Server Generates literal configuration
  - One step removed from “bits on disk”
- Generator Interface
  - Provides access to configuration data of arbitrary formats
  - Can implement adminstrative policy or procedures
  - How hosts can affect changes on one another
- Metadata system
  - Aspect based classing system
  - Currently hand managed

Bcfg2 Model

- Client configuration is represented declaratively
- Configurations can be represented as a set of interrelated configuration elements
- Client configurations are comprehensive
  - All aspects of system configuration included in repository
  - No blind spots
- Installation process abstracted away from the user
  - User worries about end states, not transition process
- Client initiated operations
  - With server maintained client states
- Generators sequester arbitrary logic on the server side
  - Validates wire-format model

Status

- Two deployments completed
  - ~500 machines total
- Three more in progress
- Deployments correspond to responsibilities
- Administrators trust Bcfg2 in general
  - More willingness to integrate complex systems and processes
- Impact
  - Reduced administration requirements
  - Wholesale OS upgrades became a background task
  - Radically different mindset for adminstrators
    - Far more like programming
    - Activities far more controlled – less firefighting

Failures

- Metadata system too complicated
  - too many ways to do things
- File caching semantics are tricky with POSIX
  - no transactions
- Administrators are really sensitive to complexity
  - Don't like XML

Research Areas

- Trying different approaches to usability and outreach
  - Focusing on concrete features to aid in adoption motivation
  - What is the goal, anyway?
- Administrative Applications
  - IP Management
  - User Management
- Delegation
- Transactions
  - Client Assertions
- Portability of configuration specification
  - Templating
  - Static respositories

Research Areas (cont)

- Scalability
  - What are the requirements?
- Uniform interface to configuration data
  - Are hosts special aggregations of configuration?
  - Should the system accommodate devices unable to run programs?

XML

- Used for client configuration descriptions
  - Format with semantics
  - Client programmed to use those semantics
- System metadata
  - Currently hand managed
  - Need something better
  - Maybe Ed's compiler?
- Generator Data
  - Structured on a task by task basis
  - Paired with logic
- Schemas used for repository validation
- Tools written to automatically generate XML where possible

Desired Functionality

- Higher order information management tools
  - Current data managed by hand
- Data Management is getting complicated
- Constraint compiler?
- Automatic reconfigurations based on state changes
- Higher order configuration descriptions
  - Client assertions can provide some functionality, but language is still an issue