

# Bcfg2, Config Management, and You

Narayan Desai

[desai@mcs.anl.gov](mailto:desai@mcs.anl.gov)

Mathematics and Computer Science Division

Argonne National Laboratory





# Overview

- Configuration Management
  - What is it, why is it hard
- Bcfg2
  - Overview and design goals
- Considerations in choosing a configuration tool



# Configuration Management

- Configuration management is an API for programming your configuration
  - Each tool defines a DSL
- Why would I want that?
  - Too many nodes to do it by hand
  - Too many people modifying configuration
  - Too much configuration diversity
  - Higher level configuration programming
- Goals for configuration management tools
  - Efficient representation of diversity
  - Node scalability
  - Programmability



# Sounds simple, right?

- Node count is the easiest thing to scale (IMO)
  - Lots of scalable algorithms for broadcasts, etc
- Configuration diversity is a lot trickier
  - Combinations can be brittle
  - Some tastes don't taste good together
- Increasing administrator population is ever harder to support
  - Larger head count usually means more specialization
  - More coordination overhead
- Matching administrator mental models is key
  - Administrators in charge of ensuring things work
  - Vary from administrator to administrator
  - Often no consensus within groups



# Comparisons to software development

- Configuration management development is similar to software development
- Similar pitfalls
  - Branches are trivial and merges are hard
- Similar approaches help
  - Version control
  - Testing and validation
  - Release management processes
- Still an active area of development, with new buzzwords from year to year
- Most techniques are vastly easier to deploy in new environments
  - Much more costly in pre-existing ones
  - Like adding unit testing to pre-existing codebases



# Bcfg2

- Client/Server system written in Python
  - ~20K sloc
- Lightweight
  - Single server easily handles 1k nodes
- Flexible enough to handle wide range of use cases
  - Start with small static environments requiring auditing
  - Grow to large scale HPC systems
  - Elegantly configure dynamic cloud infrastructures
- Production grade and reliable
  - First external deployment in 2005
  - Used at minimally 100 sites
  - Users cut across all major market segments
- Active and helpful community





# Key Design Goals

- Model system configuration in unambiguous, simple terms
- Close the loop between administrators, configuration specification, and current system states
- Enable a variety of administrative regimen for systems
- Support extensive configuration debugging
- Composition of information from a number of sources
- Expose plugin api to all aspects of the configuration process
- Configuration Meta-programming



# System Configuration Modelling

- Specifications are declarative
  - Describe goals, not process
- Entries describe common system objects
  - Software packages
  - Services
  - Configuration files
  - POSIX filesystem entries
- Validation based on congruency
  - Matches performed against all entries in the configuration
  - And no extra configuration detected on the client
- Design allows translation between specification and current state
  - Specification can be rendered into reconfiguration operations when needed
  - State can be rendered into a declarative configuration specification





# Closing the loop between goals and reality

- Tools fail!
  - Bugs
  - System failures
  - Thinko's
- Need independent verification capabilities
  - Do I have what I said I wanted?
- Built an interface that reports on client state back to the server
  - Full operation log
  - Divergence between stated goals and current state
  - Extra configuration entries
  - Performance data
- And a reporting system that shows overall system configuration health
  - Multi-client patterns in state



# Enable administrator choice of deployment strategies

- It is essential that administrators control how configuration happens
  - Bcfg2 designed as a swiss army knife
- Basic client modes
  - Dryrun
  - Interactive
- Selective rule based deployments
  - Only deploy changes selectively
  - (based on the change itself, the client or the time)
- Support bidirectional specification flow
  - Server -> Client (typical)
  - Client -> Server (delegation)



# Configuration Debugging

- Like in software engineering, debugging is needed
  - Complicated systems don't always act as expected
- Built a full system introspection capability into Bcfg2
- From the server-side
  - Query full client metadata
  - Perform configuration goal construction
  - Drop into a configuration debugger
- Good framework for centralized testing



# Configuration Process/Plugin API

- Bcfg2 is configuration plumbing
- Users can customize all aspects of the configuration process
  - Using Python
  - With a simple API
- Major functional areas
  - Client probing
  - Metadata resolution
  - Configuration goal specification
  - Validation
  - Processing of state information



# Configuration Meta-programming

- The first stage in configuration management is literal configuration
  - “copy this file into that location”
- The next stage is rule based configuration
  - “webservers get this configuration, ftpservers get that one”
- The third stage is meta-programming
  - Configuration patterns used to generate configuration goals
  - “ntp clients should talk to our ntp servers”
  - “the ssh\_known\_hosts file should contain entries for all machines”
- Supported in Bcfg2
  - Through entry templating
  - Query interface to metadata
  - Raw access through Plugin API



# Considerations in choosing and deploying a tool

- Using any robust tool will be better than manual administration
- Find the tool that matches your administrative philosophy
  - Mental models are a big part of that
- Communities have different personalities
  - Find a comfortable one
- Assume deployment will take twice as long as you expect
  - The first 70% goes fast, but the rest..
- Building group consensus is key
  - You will have problems if you force a solution into place by fiat
  - Take the time to teach people how to use the system
  - Listen to their concerns



# Questions?

<http://www.bcfg2.org>  
Irc.freenode.net #bcfg2

