Argonne NATIONAL LABORATORY

DevOps and Bcfg2

Sol Jerome

Enterprise Systems
The University of Houston

February 19, 2011

BCFG2

Introduction

About Bcfg2

- Written in Python
- First release in 2004
- Included with most linux distros
- Supports Linux/Solaris/OSX/etc
- BSD licensed

BCFG2

Introduction

About Bcfg2

- Centralized repository
- Self-documenting
- Declarative semantics model
- Varied client modes (allowing for incremental adoption)

BCFG2

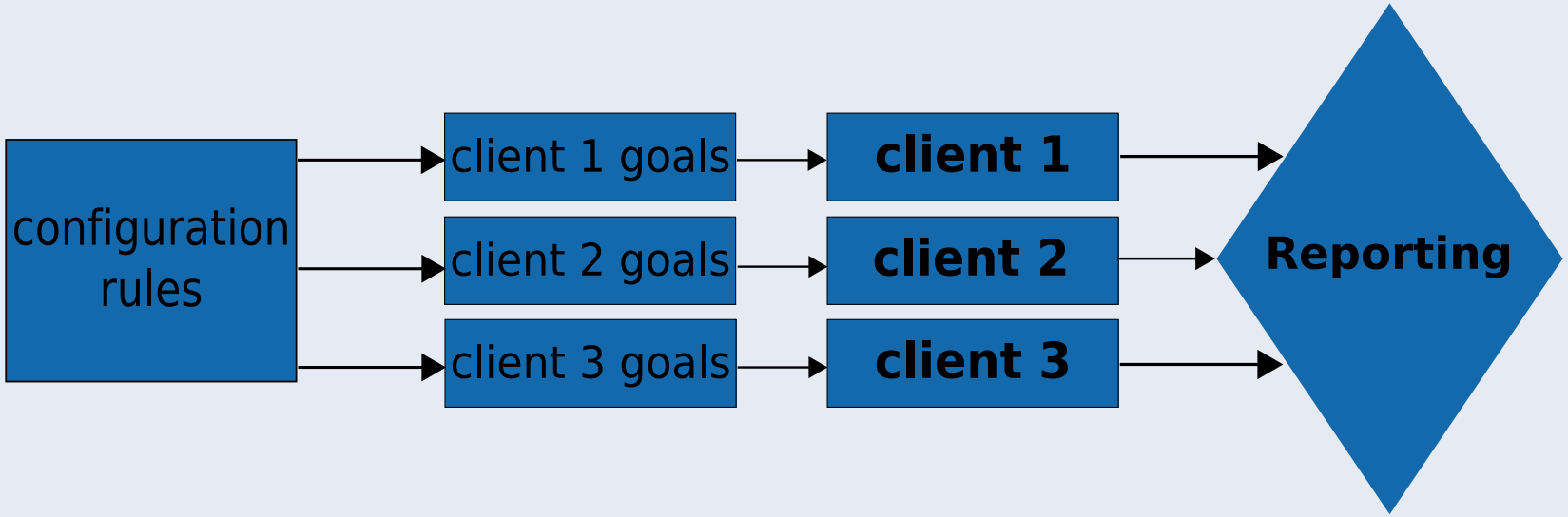
The Bcfg2 Architecture

Architectural goals

- Built on verification
- Configuration is discoverable
- Rules used to build per-client configuration goals
- Clients compare goals with current state and reconcile

The Bcfg2 Architecture

Architectural goals



The Bcfg2 Architecture

Configuration goals

- Goals comprised of collections of entries
  - Entries correspond to familiar types (Path, Package, Service, etc)
  - Entries are Verifiable, Idempotent, Installable
- Comprehensive
- Complete
- Literal
- Typical client goals contain 200-2500 entries

BCFG2

The Bcfg2 Architecture

Bcfg2 Client Functions

1. Executes local state probes
2. Retrieves goals from the server
3. Compares current local state to goals
4. Determines which goals are unmet
5. Attempts to meet those goals
6. Report on local state back to the server

BCFG2

The Bcfg2 Architecture

Client Execution Triggers

- Init script (boot time)
- Cron jobs (hourly or daily)
- Job prologue/epilogue (in HPC environments)
- Agent mode (via SSH)

BCFG2

The Bcfg2 Architecture

Action Determination

- Normal mode (all pending changes made)
- Dry run mode (no changes made)
- Interactive mode (user prompted for each change)
- Extra entry removal (remove unmanaged entries)
- Centralized decision mode (whitelist/blacklist)

BCFG2

The Bcfg2 Architecture – Reporting System

Client Reporting Data

- Entry counts (total/good/bad)
- Detailed entry information
  - Good Entries
  - Bad Entries
  - Modified Entries
  - Extra Entries
- Detailed Activity information
- Performance data

BCFG2

Argonne NATIONAL LABORATORY

Server Architecture

The Bcfg2 Server

- Serves data using XML-RPC over HTTPS
- Two main tasks
  - Rendering configuration rules into per-client goals
  - Routing client statistics to the reporting system
- Like all sufficiently mature software, has developed a plugin infrastructure for extension

BCFG2

Bcfg2 Server Plugins

Plugin Capabilities

- Configuration goal construction
- Metadata functionality
- Statistics handling
- Configuration validation
- Specialized/customized configuration rule representations

Bcfg2 Metadata

Client Metadata

- Describes aspects of the client
  - Group memberships
  - Arbitrary specificity
  - Inheritance
- Includes
  - Client Identifier (hostname or uuid)
  - Probed client information
  - Information from external sources (via Plugins)

Bcfg2 Server Plugins

Examples of Stock Plugins

- Cfg
  - Manages configuration files
- Packages
  - Manages package deployment (dependency handling)
- SSHbase
  - Manages ssh keys and known_hosts files
- Templating Plugins
  - Provides templating interface for files and Bundles
- Ldap
  - Experimental plugin for adding Metadata from a directory server

BCFG2

DevOps and Bcfg2

What can you do with Bcfg2?

- Reporting allows fine-grained rollback mechanism (especially when combined with version control)
- Deploy custom in-house applications
- Pull in manual configuration from clients
- Automate service deployment
  - Services can be configured with templates to automate complex processes

BCFG2

Conclusion

Source code, documentation, papers, and much more available at: http://bcfg2.org http://docs.bcfg2.org