AGH University of Science and Technology, Kraków
Department of Computer Science

Installation of complex e-Science applications on heterogeneous cloud infrastructures

MSc Thesis

Bartosz Wilk

Supervisor:
Marian Bubak

Consultancy:
Marek Kasztelnik, ACC Cyfronet AGH
Adam Beloum, University of Amsterdam

Reviewer:
Marek Wieczorek, Google Krakow

Outline

Background

- Motivation
- Objectives

State of the Art

- Assessment of Available Solutions
- New Solution Concept
- Modeling of Configuration Domain

Implementation

- Design of the Tool
- Overview of Cloudberries Implementation
- Cloduberries Interface
- Tool Validation

Conclusions

- From Feature Model
  to Automatic Product Line Generation
- Summary

Motivation

Cloud computing in e-Science

- Attracts more and more researchers
- Cheap (pay for use not for hardware)
- Powerful and scalable
- Highly available

e-Science application deployment

- Distributed infrastructure
- Heterogeneous environment
- Complex configuration
- Various software and data

Scientists

- Focused on research
- Should not pay attention to technology
- May lack administration knowledge
- Want to be more productive

Consequence

- Need for a tool to build an experiment environment

Objectives

The aim of this work

- Provide a solution increasing the productivity of scientists
- Reduce the complexity of experiment environment configuration

Approach

- Employ automation whenever it is possible
- Adapt modern system administration solutions
- Introduce an interface facilitating the use

Assessment of Available Solutions

System administration software

- Distributed shell – simultaneous operation on multiple shells
- Unattended installation – automated operating system installation
- Provisioning tools

Provisioning – what is it about?

- Client-server architecture – client is installed on the target node and operates in the command of server
- Installation packages / scripts– deployment procedures are organized in packages and managed by repositories
- Declarative configuration – deployment is managed declaratively to abstract platform-specific behavior

Provisioning tools evaluation

- Bcfg2, CFEngine, Chef, Puppet
- Similar possibilities, different interfaces
- Too inconvenient to attract scientists

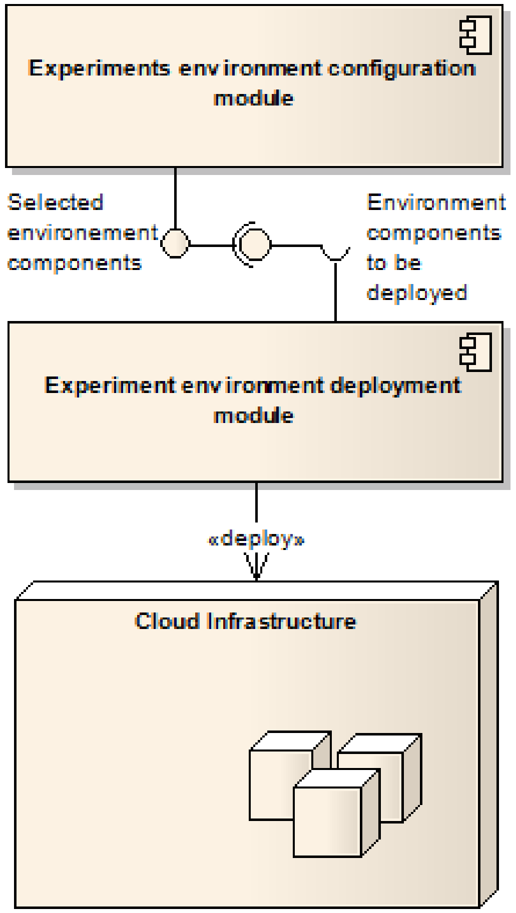
Why to use Chef?

- Windows support – essential for VPH-Share project
- Community – installation package repository (*cookbooks*)
- API – REST and third party Java support
- May be expanded to facilitate the use

Overview of a New Concept

Proposed solution

- Provides additional application layer built on top of provisioning system.
- Powerful mechanism of deployment based on modular configuration units *(cookbooks)* allows to split the process of installation from the selection of components
- Graphical user interface minimizes the complexity of configuration. Intellectual load of a scientist boils down to the selection of configuration components and attributes customization
- Administration tasks are delegated to the *Chef* repository administrator
- Deployment can be executed again thanks to the storage of configuration
- Deployment process can be monitored in the real-time via the GUI


Experiments environment configuration module
Selected environement components
Environment components to be deployed
Experiment environment deployment module
«deploy»
Cloud Infrastructure

A challenge to be addressed

How to choose an appropriate representation for the domain of configuration components? The selected representation has to be complex enough to comprise their hierarchy and dependencies as well as to be easily understandable for scientists.

Modeling of Configuration Domain

How do the others solve similar problems?

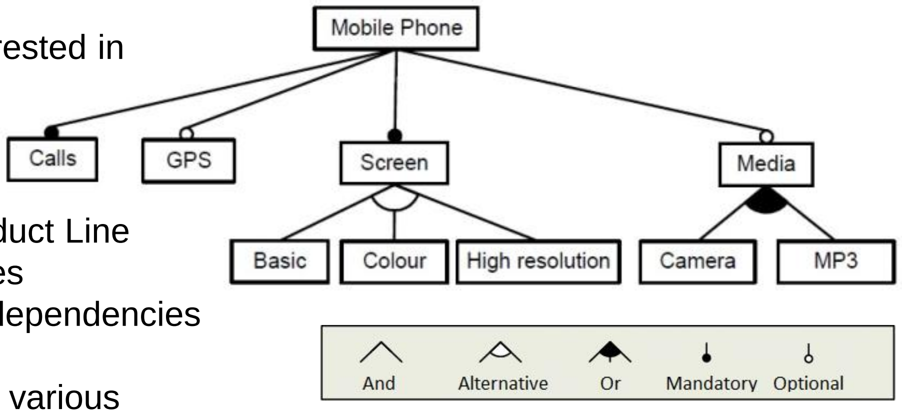
Software Product Line

- Paradigm and engineering methods for software creation
- Reusable assets (software components, source code, data)
- Repeatedly applicable means of production
- Client decides which features he or she is interested in

The same case? - maybe…

Feature Modeling

- Modeling method widely used in Software Product Line
- Each product is represented as a set of features
- Model comprises a hierarchical structure and dependencies
- Easy to read and understand
- Generic semantics allows for representation of various configuration domains



Feature Model automated reasoning State of the Art

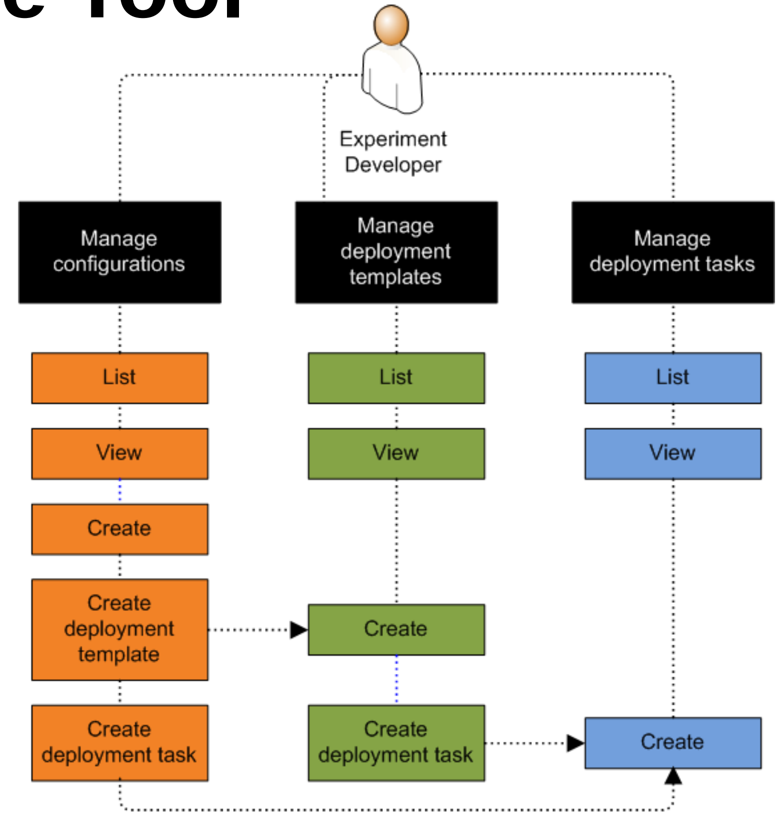
- Operations (product validation, conflict detection, autocompletion)
- Problem mappings, algorithms and data structures: Boolean Satisfiability Problem (SAT), Binary Decision Diagrams (BDD)
- Software Libraries:
  - Low-level solvers: SAT4j, JavaBDD
  - Feature Model reasoners: AHEAD, Choco, FaMa, SPLAR

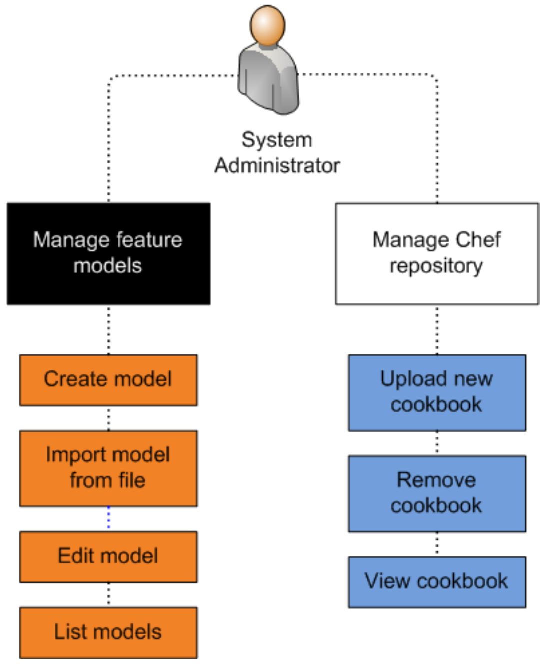
Design of the Tool

Feature Model was employed to represent the structure, hierarchy and dependencies of configuration components. Features are mapped to installation packages *(Chef cookbooks)* provided by the administrator.

Scientific user - Experiment Developer

- Configures e-Science application by selecting features and specifiying attribute values.
- Can store attiriibute defaults in a from of templates.
- Is able to run and monitor installation process via the graphical user interface.





System administrator

- Manages Feature Models
- Creates and tests installation packages (*cookbooks*)
- Extends the domain of configuration components by providing the system with new features and corresponding installation packages (*cookbooks*).

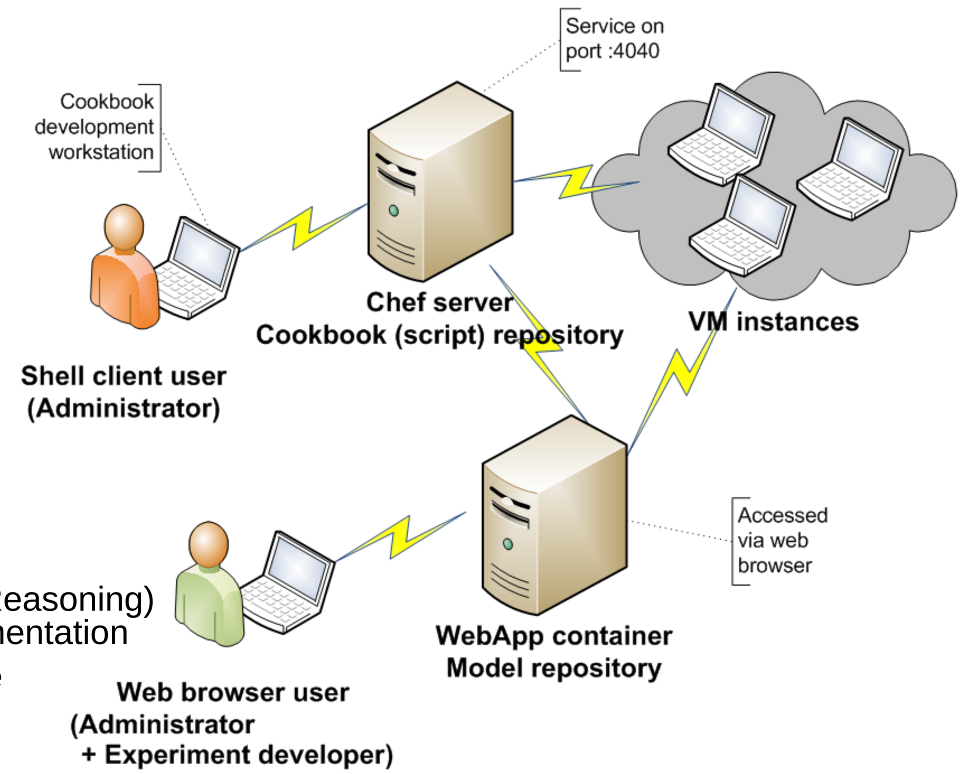
Overview of Cloudberries Implementation

Cloudberries – the tool architecture

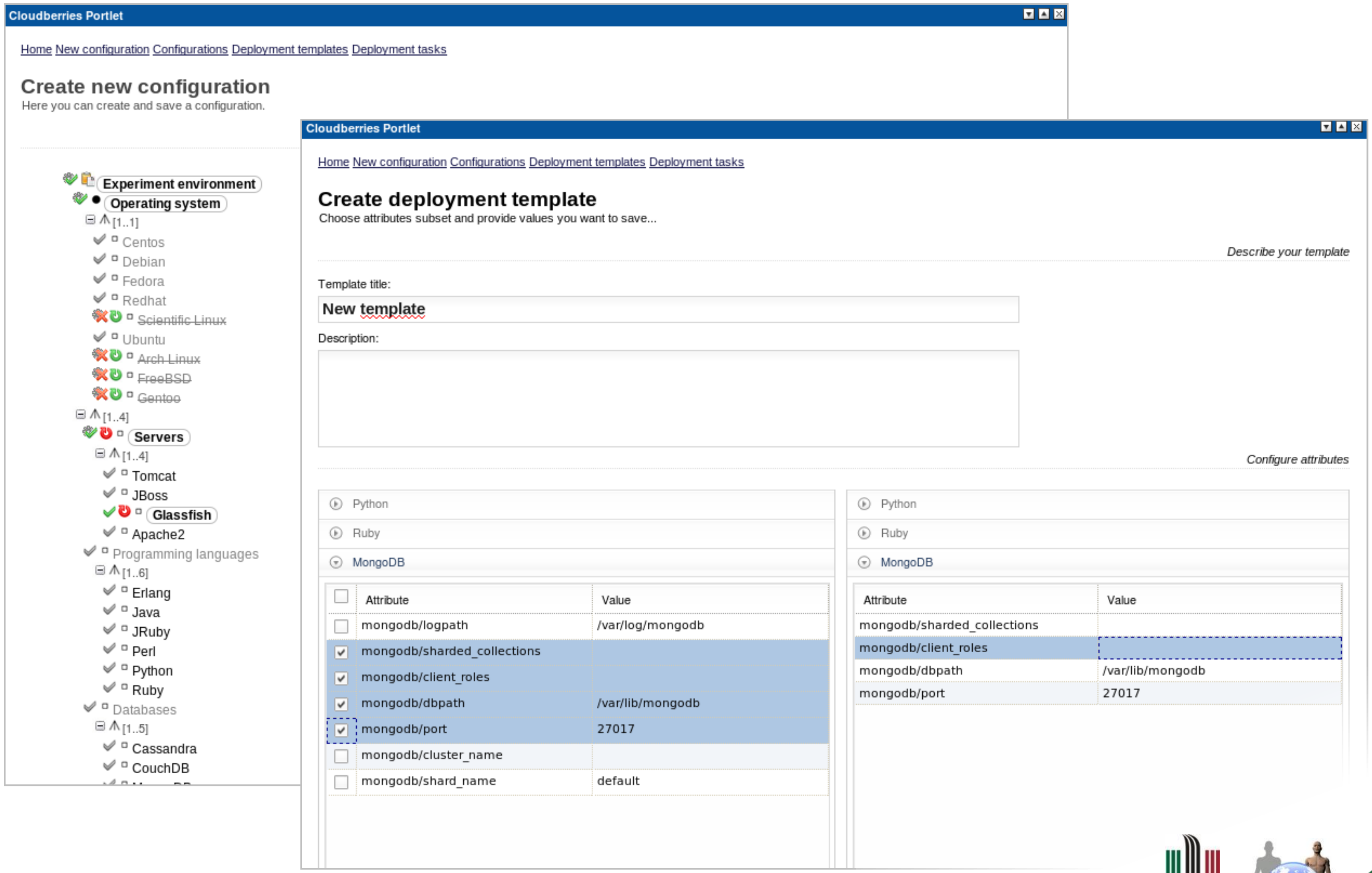
- Java Enterprise Edition Web application
- Integrated with Jetspeed Portal in VPH-Share production environment
- Relational database backend
- Chef server may be remote
- Chef bootstrap via SSH on each node
- Virtual Machines – internet access

Choice of technology

- Chef – cloud provisioning tool
- jclouds-Chef – Chef programmatic interface
- SPLAR (Software Product Lines Automated Reasoning) – Feature Model reasoning algorithms implementation
- Java Portlet API – integration with VPH-Share project portal (JetSpeed)
- Spring MVC – portlet support, IoC container
- FreeMarker – template engine for MVC View generation
- Dojo Toolkit – DOM manipulation, JavaScript, AJAX
- Hibernate – Java Persistence API implemenetation - object/relational mapping


Service on port :4040
Cookbook development workstation
Chef server
Cookbook (script) repository
VM instances
Shell client user (Administrator)
Accessed via web browser
WebApp container
Model repository
Web browser user (Administrator + Experiment developer)

Cloduberries Interface


Cloudberries Portlet

Home New configuration Configurations Deployment templates Deployment tasks

Create new configuration

Here you can create and save a configuration.

- Experiment environment
- Operating system
  - [1..1]
    - Centos
    - Debian
    - Fedora
    - Redhat
    - ~~Scientific Linux~~
    - Ubuntu
    - ~~Arch Linux~~
    - ~~FreeBSD~~
    - ~~Gentoo~~
  - [1..4]
- Servers
  - [1..4]
    - Tomcat
    - JBoss
    - Glassfish
    - Apache2
- Programming languages
  - [1..6]
    - Erlang
    - Java
    - JRuby
    - Perl
    - Python
    - Ruby
- Databases
  - [1..5]
    - Cassandra
    - CouchDB

Cloudberries Portlet

Home New configuration Configurations Deployment templates Deployment tasks

Create deployment template

Choose attributes subset and provide values you want to save...

Describe your template

Template title: New template

Description:

Configure attributes

Python

Ruby

MongoDB

	Attribute	Value
☐	mongodb/logpath	/var/log/mongodb
✓	mongodb/sharded_collections	
✓	mongodb/client_roles	
✓	mongodb/dbpath	/var/lib/mongodb
✓	mongodb/port	27017
☐	mongodb/cluster_name	
☐	mongodb/shard_name	default

Python

Ruby

MongoDB

Attribute	Value
mongodb/sharded_collections	
mongodb/client_roles	
mongodb/dbpath	/var/lib/mongodb
mongodb/port	27017

Tool Validation

Cloudberries were integrated with the VPH-Share portal and successfully deployed in the project production environment; the tool can be accessed via a web browser at http://vph.cyfronet.pl/puff

As a case study for Cloudberries evaluation euHeart e-Science application was chosen to be installed

- Installation consisted of 12 configuration steps including software installation, creation of system user, granting system privileges, copying files
- The process of configuration was entirely automated and boiled down to a single feature selection

System usability evaluation

- User interfaces
  - Minimized intellectual load - configuration is simple and intuitive
  - Facilitated redeployment - configuration can be stored at two different levels (selection of components and deployment template)
  - Deployment trace - collected in a persistent entity; can be accessed either at the time of installation process or recalled later
- Tool installation - trivial procedure

System security – adequate

- User space is secured by the JetSpeed portal
- Communication with Chef is secured by SSL

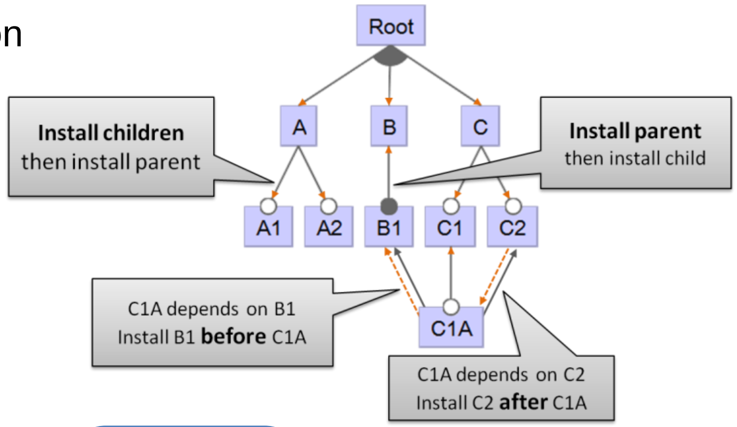
System scalability

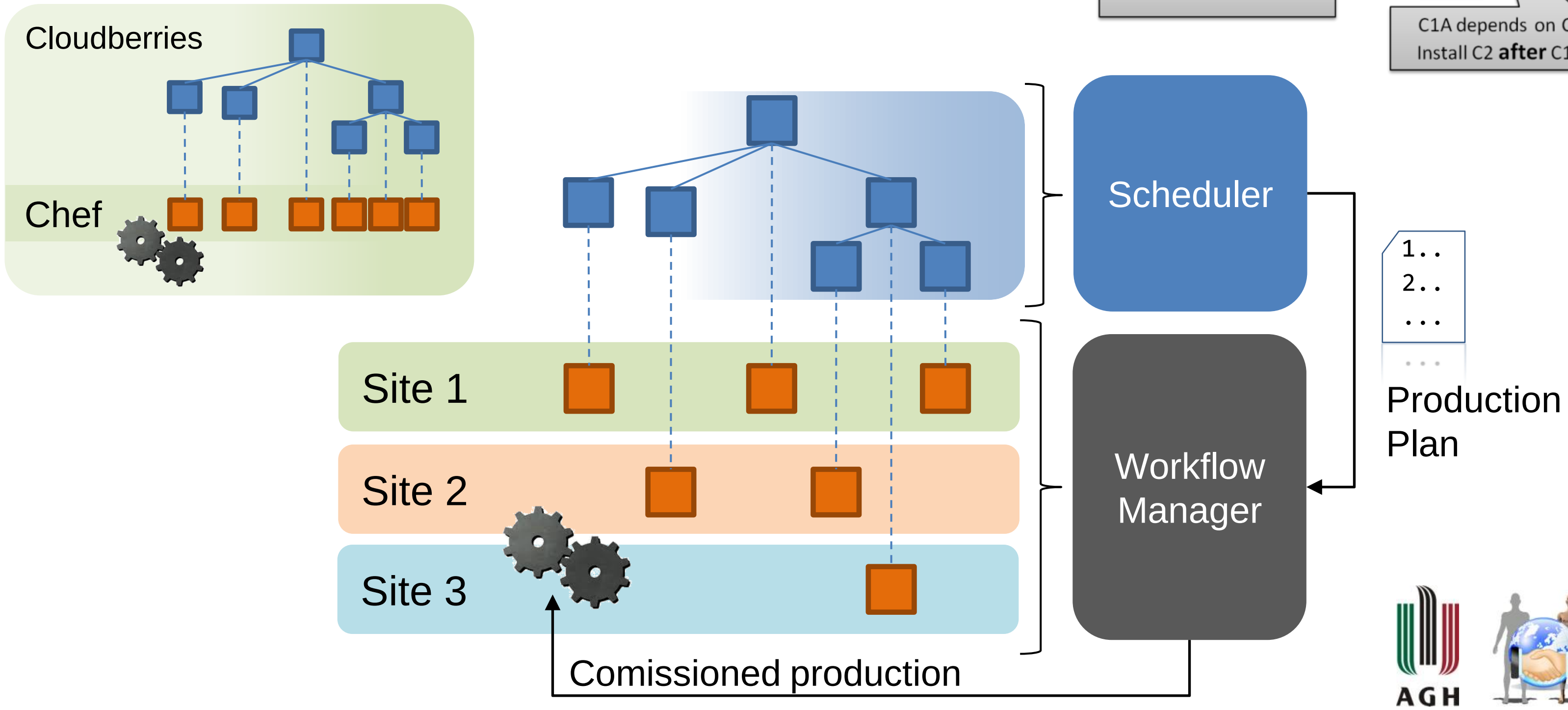
- System loaded by user request – the tool may be easily scaled by frontend / backend components replication
- Domain of configurations – easily extensible. Scalability limited by model complexity (model has to be understandable for administrator)
- Scalability of deployment – multiple deployment planned for future work

From Feature Model to Automatic Product Line Generation

Concepts:

- Generic framework for Software Product Line creation
- Extended Feature Model (additional relationships)
- Production workflow mechanism automatically derived from the model
- Extensible modular architecture (Production Sites)
- Production parameter exchange layer
- Feature Model - based configuration interface


Root
A
B
C
A1
A2
B1
C1
C2
C1A
Install children then install parent
Install parent then install child
C1A depends on B1 Install B1 before C1A
C1A depends on C2 Install C2 after C1A



Summary

Results

Research:

- Feasibility study of Feature Model adaptation to modeling of an e-Science application composition
- An architectural concept of a framework for Feature Model – based automatic production line generation

Cloubderries tool:

- Reduces the complexity of experiment environment configuration
- Minimizes e-Science application installation effort
- Automates redeployment
- Memorizes trace of deployment
- Increases the productivity of scientists

Future work

Tool:

- Implementation of multiple node deployment
- Implementation of filtering mechanism in the user interface
- Implementation of user interface allowing for feature model edition
- Implementation of on-demand Virtual Machine instance creation

Research:

- Production scheduling algorithm
- Implementation of a service-oriented framework
  for production line generation

More at http://dice.cyfrone.pl/VPH-Share

The result of this thesis is the paper
Bartosz Wilk, Marek Kasztelnik, Marian Bubak
„Installation of complex e-Science applications on heterogeneous cloud infrastructures„
for *Software Practice and Experience*

This thesis was realized partially in the framework of the following projects:

Virtual Physiological Human: Sharing for Healthcare (VPH-Share) – partially funded by the European Commission under the Information Communication Technologies Programme (contract number 269978).

Project UDA-POKL.04.01-01-00-367/08-00 "Improvement of didactic potential of computer science specialization at AGH", at the Department of Computer Science, AGH University of Science and Technology, Al. A. Mickiewicza 30, 30-059 Kraków