The CHAMPS System: A Schedule-optimized Change Manager

Alexander Keller (alexk@us.ibm.com)
with: Joseph L Hellerstein, Joel Leonard Wolf,
Kun-Lung Wu, Vijaya Krishnan, Aaron Brown
Goal: Installation Schedule of an Internet Storefront

- Wouldn’t it be nice if:
  - one could automatically generate such workflows and compute an optimal Schedule?
  - let the Administrator extend/modify/combine them by means of COTS tools, if needed?
  - automatically (re-)assign the best available resources from pools, based on SLAs and Policies?
  - estimate individual task durations (or, at least, reuse data from previous deployments)?
  - automatically detect which tasks must be applied sequentially, and which can be done in parallel?
Change Management Process and CHAMPS Scope

RFC

Filter Request

 Assess Change Impact for “x”

Consolidate Impact (cost/timing)

Create Change Plan for “x”

Verifying Change Plan

Test Change Plan

Final Approval (cost/time)

Implement Change

Verify Change

Success

Close RFC

Reject

No Go

Valid

Go

Inconsistent

Issues

Go

Not Valid

Go

Source: ITIL
Task Graph Builder: Install TPC-W `bestsell` Servlet (1)

TPC-W Application

Servlet Container

Web Application Server

Database

Dependency Strengths between Systems:

25%, 50%, 75%, 100%
Task Graph Builder Example: Install *bestsell Servlet* (2)

Goal: Maximize Parallelism for time-consuming Tasks, gather Information from Deployment Descriptors
Task Graph Builder Example: Install *bestsell* Servlet (3)

- Using Workflows for Task Graphs and Change Plans:
  - COTS Tools
  - Offload state-checking of Activities to Workflow Engine
- BPWS4J is a BPEL4WS Workflow Environment
  - Workflow Editor
  - Workflow Engine
  - Available on IBM Alphaworks
- Per-Host Structure defined by outermost Sequences
- Cross-System Dependencies inserted as links between Activities
Planner & Scheduler

- Determines sequencing of RFCs
- Binds tasks to servers/resources
- Determines when tasks execute

Time

Server 1
- Install Linux OS
- Install IP Service
- Install Application Server

Server 2
- Install AIX OS
- Install IP Service
- Install Database

Server 4

Server 5

Task

RFC
Optimization Problem Formulation

Maximize Revenue

\[
\sum_{j} V_j x_j - \sum_{j} \sum_{k} \sum_{i} \sum_{p} a_{\beta_{j,k},j,p} \int_{s_{\alpha_{j,k}}}^{c_{\beta_{j,k},j,p}} \sum_{j} C_{j,k}(t)dt
\]
Parameterization according to Objective Functions

- Maximize profits
  * if $C_{j,k}(t) \equiv 0$
- Maximize number of jobs done
  * if $C_{j,k}(t) \equiv 0$ and $V_j \equiv 1$
- Minimize costs
  * $V_j \equiv 0$
- Minimize downtime due to jobs
  * if $V_j \equiv 0, \mathcal{K}_j = 1, \alpha_{j,1} = 1, \beta_{j,1} = I_j, C_{j,1}(t) = 1$
- Minimize total execution time
  * if $V_j \equiv 0, \mathcal{K}_j = I_j, \alpha_{j,k} = \beta_{j,k} = k, C_{j,k}(t) = 1$

Job = RFC

Objective Functions represent Policies
Planner & Scheduler: Scheduling 7 RFCs

- Performs Assignment of Tasks to Resources from Pools (Late Binding)
- Supports very general Objective Functions:
  - Minimize penalties from SLA step functions with multiple deadlines
    - On an RFC and/or task basis
  - Minimize sums of RFC makespans
    - Or weighted sums of RFC makespans
  - Minimize average response (flow) times
    - Or weighted average response times
- Any combination of these and other objective functions
Status, Lessons learned & Research Challenges

- CHAMPS Components architected to support Late Binding of Change Plans
  - Task Graph Builder (dependency acquisition from deployment descriptors, automated workflow creation in BPEL4WS)
  - Planner & Scheduler (resource & date/time assignment, build optimized Schedule)

Lessons learned
- BPEL4WS is a viable alternative for expressing change management workflows
  - Concurrent execution, Deadlines, Links across different workflows
- Availability of Dependency Information is crucial
  - Facilitates creation of Provisioning Workflows based on Change Mgmt. Ops. & Impact Analysis
  - Tooling for automated creation of Deployment Descriptors at Development Time needed

Research Challenges
- Reusable Change Plans: Task Graph Library
- On-line Plan Adjustment
  - In case Provisioning Activities are running behind Schedule
  - Based on SLAs and Policies
- Passing Configuration Information along with Workflows
- Workflow-driven Provisioning with BPWS4J Workflow Engine
  - Integration of BPWS4J Workflow Engine & Tivoli Intelligent Orchestrator
Example: DB2/WAS Provisioning for SPECJAppServer2004

- Install and configure DB2 (Linux) and WAS (WinXP) on two machines to run untuned SPECjAppServer2004 benchmark
- Procedure: 64 steps, 3 context switches
- Knowledge: 26 distinct parameters used 58 times, 7 outside of source context
- Today: Manual setting of parameters by Admin, default settings, response files
- Approach: group parameters in configuration templates, annotate workflows

**DB2:**
- create user
- install wiz. (db2)
- select features (2)
- specify params (7, 5 def.)
- reboot
- app cmd (cli)
- env var (3)
- cfg file (1 line)
- app cmd (cli)
- start svc (db2)

**WAS:**
- install wiz (db2)
- select features (4)
- app cmd (cli)
- app cmd (cli)
- install wiz (was)

Knowledge:
- user/pass
- db2host
- db_name

Knowledge = workflow
= cfg data flow

node alias
Demo: Concurrent Provisioning of 2 Websites with BPEL

- Start:
  - 2 Linux Systems without http Server
  - 1 IHS 2.0 Package; 2 zip-Files w/ Web Content
  - 1 Change Plan for provisioning them
- Load the BPEL Change Plan in BPWS4J:
  - Sequences/Flows of Actions
  - Partner Identifications
- Execute the BPEL Change Plan
  - Done by BPWS4J Engine
  - Trigger TIO Logical Operations
  - Monitor Execution Status:
    - From BPWS4J
    - From TIO
  - When 1 Task completes, start next
- Result:
  - 2 different Websites serving different Web Content