ŠIAULIAI UNIVERSITY
EVERYTHING HAPPENS HERE

Grid- and Cloud-based Model for Evaluation of Simulated Annealing Method for Distributed Software Engineering

Lithuanian National Grid Initiative is well known by its number of the grid applications for various domains including software engineering.

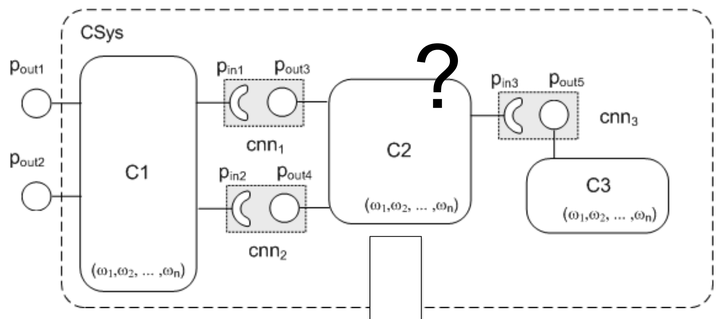
In distributed software engineering process it is important to answer to 2 main questions:

- Can we compose required software from existing components? This can be done using automated component selection tools (e.g. *SoCoSys*).
- **Is this compositional solution the best?**
  The evaluation of huge set of the components-candidates with the same functionality but with different non-functional attributes having relatively short time for evaluation of such attributes.

We have proposed the method which is capable to solve second problem. However the algorithm for the evaluation and the improvement of the component set presents the class of algorithms. It is possible to change the possibility of global optimum detection of our optimization algorithm using different values of parameters α, β, T and g.

The parameters are real numbers, consequently isn't possible to test all the values. However in order to investigate the impact of the values of parameters it is necessary to iterate over large sets of discrete values.

Moreover in order to ensure that result is independent from compositional problem, it is necessary to repeat calculations on set of the problems.



$$f(x) = \sum_i \sum_j x_i.\omega_j \bullet s_j$$

$$f(x) \to \max$$

Step 1

initial component set is x_0,
the list of initial components' indexes is $v = (v_1, \ldots, v_m)$,
the quality of the CBS system is $f_0 = f(x_0)$,
the "temperature" is T_0

Similarity Matrix

$$R^i = \left[R^i_{s,u}\right]_1^{M_i}$$

$$R^i_{s,u} = 1 - \frac{\sum_{j=1}^{n} \left|a^i_s.\omega_j - a^i_u.\omega_j\right|}{n}$$

$$0 \le R^i_{s,u} \le 1$$

The number of already performed iterations $k = 0$
particular values of the coefficients α_0, β_0 and g_0 are selected

Step 2

The list of indexes $r = (r_1, r_2, \ldots, r_m)$ is generated. Each value of r_i is selected such as

$$\frac{\sum_{l=1}^{r_i-1} (R^i_{v_i,l})^{\alpha_k}}{\sum_{l=1}^{M_i} (R^i_{v_i,l})^{\alpha_k}} < \eta_i \le \frac{\sum_{l=1}^{r_i} (R^i_{v_i,l})^{\alpha_k}}{\sum_{l=1}^{M_i} (R^i_{v_i,l})^{\alpha_k}}$$

The candidates to the r_i are iteratively chosen from the range $(1 \ldots M_i)$ until the inequality is true.

Step 3

New set of similar components $y_{k+1} = (y_1, y_2, \ldots, y_m)$ is taken
New list of components' indexes is fixed: $v = r$.

Step 4

The value of the function $f_{k+1} = f(y_{k+1})$ is calculated and the values of T_{k+1} and α_{k+1} recalculated:

$$T_{k+1} = g_k \cdot T_k, \qquad \alpha_{k+1} = \beta_k \cdot \alpha_k$$

The parameter g_k is classical parameter of the Simulated Annealing method,
α is used to calculate probability that credentials of neighborhood component are better than current component.
To avoid so dynamic "jumps" the value of α is constantly changed by coefficient β

Step 5

If inequality $e^{\frac{f_{k+1} - f_k}{T_{k+1}}} > \mu$ is true then

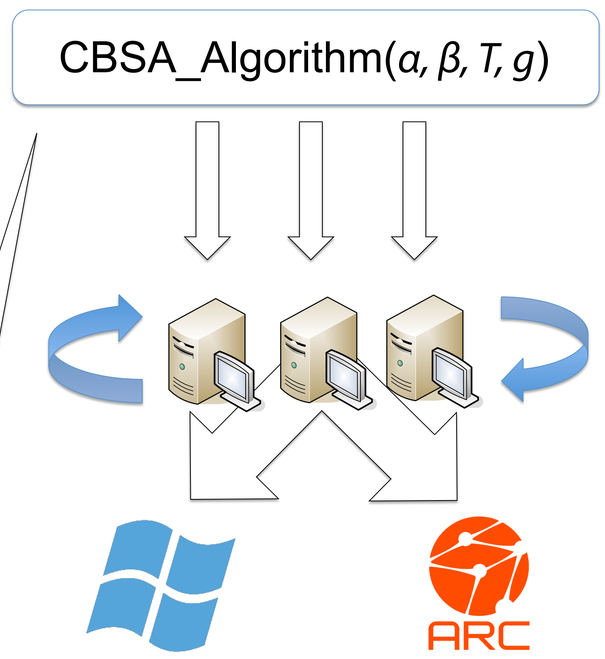
$x_{k+1} = y_{k+1}$ (i.e. new selected set of components is approved as the possible solution)

Else

$x_{k+1} = x_k$ (i.e. new selected set of components is rejected);
$k = k + 1$ and go to Step 2.

Step 6

The algorithm stops when
fixed number of iterations is already performed OR
the criteria have not improved after several steps.



We are using massive computations using two distributed computing infrastructures:

- **Windows Azure.** It's legacy choice because initial program for our experiment was developed using C# (for one PC). For distribution of tasks to different VMs Power Shell script is used.
- **ARC-based grid.** Advance Resource Connector is used as a main middle-ware by Lithuanian NGI. To make grid-based solution, our program is re-implemented using C++. The tasks are submitted, monitored and the results are collected using command-line tools. This kind of interface is very useful and perfectly fits the generative software development methods, we are using.

We are looking for the international partners for the research on grid and cloud-based development of distributed software. Any contribution is welcome.

Dr. Vaidas Giedrimas,
Prof. Hab. Dr. Leonidas Sakalauskas,

+370 41 595723
http://www.grid.su.lt
grid@su.lt