FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

SILCA - Self-Initiative Load Clustering Agents

Masterstudium:
Software Engineering & Internet Computing

Michael Vögler

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzer
Betreuerin: Ao.Univ.-Prof. Dr. Dipl.-Ing. eva Kühn

Introduction:

- Clustering is a method of unsupervised learning.
- Clustering groups a collection of loads into smaller subsets.
- Each subset represents a so called **cluster**.
- **Figures 1** and **2** show the result of load clustering in a network.

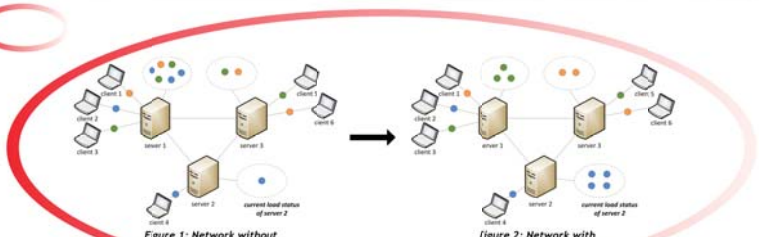

Figure 1: Network without load clustering

Figure 2: Network with load clustering

Motivation:

- The load a network has to handle dramatically rises, which has an effect on the performance of the network.
- To improve performance, load clustering can be used to group load into smaller clusters and to process a cluster of loads in one step.
- To find a suitable clustering algorithm is a complex and challenging task.
- SILCA eases this choice by providing a pattern and framework that allows the *plugging* and benchmarking of different clustering *algorithms* on different problem scenarios.

SILCA - Pattern:

- The SILCA pattern consists of several sub-patterns: **local-node**, **arbiter** and **clustering** pattern.
- These sub-patterns can be composed to form either the **basic** SILCA or **extended** SILCA pattern (**Figure 3**).
- The basic principle of SILCA are autonomous agents that communicate in a p2p-manner.
- The agents handle a dynamically changing amount of load and decide on their own if they process a load or not.

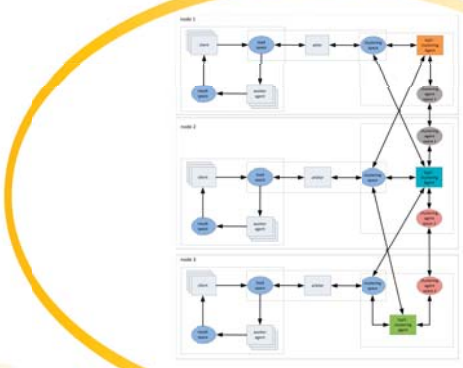

Figure 3: Composition of the basic and extended SILCA patterns

Implementation:

- For the implementation of SILCA we used **MozartSpaces 2.0** the Java based open source implementation of a space-based architecture (XVSM) and **Java 6.0**.
- Each SILCA space is realized as an addressable container, where we assume that each container is reachable in the internet.
- All containers provide the built-in *XVSM query-language for sorting* and counting operations.
- For the robustness of the framework we used the built-in XVSM transactions.

Benchmarks and Results

- For the benchmarks we used several clustering and classification algorithms.
- Benchmarked different test-settings (**Figures 4** and **5**).
- Benchmarks demonstrate the agility of SILCA by showing that algorithms can be easily plugged in and benchmarked.
- Results show that some algorithms outperform others (**Figure 6**) and that the combination of them can lead to a faster and more distinct clustering (**Figure 7**).

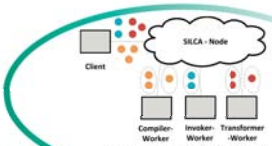

Figure 4: Test-Setting for basic benchmarks

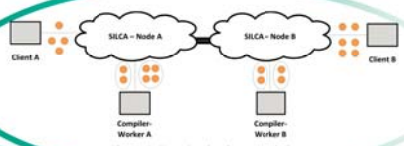

Figure 5: Test-Setting for extended benchmarks

Figure 6: Results for basic benchmarks

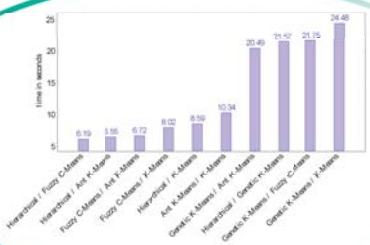

Figure 7: Results for extended benchmarks

Conclusion:

- The intention for SILCA was to develop a self-responsible load clustering architecture.
- SILCA should be extensible towards arbitrary load clustering strategies and any kind of problem scenario.
- Objectives could be solved by implementing this pattern with a **space-based architectural style**.
- For the evaluation, benchmarks were carried out that proved that an approach using active agents can be adapted to any underlying algorithm.

Kontakt: michael.voegler@chello.at