Multi-Layered Cloud Applications Auto-scaling Performance Estimation

**Goal of the guided research:** **Development of a tool which estimates and analyzes the different configurations of existing cloud auto-scaling solutions in respect to performance and costs, resulting in the best suited configuration for the deployment of application along with the pros and cons of other configurations.**

**Structure of the guided research (deadline is in square brackets – DD.MM):**

1. **[16.05]** Identify all major market-level autoscaling solutions on all levels of virtualization (AWS Autoscale, Kubernetes, etc.).
2. **[16.05]** Derive the parameters for different instances types of AWS and recreate these instances types as templates in LRZ Cloud. Identify cost model of AWS for instances types that we will use in the experiments.
3. **[16.05]** Develop model of autoscaling solutions. Each model should include:
   1. Metrics used as markers for autoscaling decisions (e.g. CPU load).
   2. Tuning parameters that are used by the autoscaling solution.
   3. Autoscaling solutions’ impact on cost.
4. **[23.05]** Identify metrics of quality for autoscaling solutions, e.g. how long does it take to scale in/out the application under variable load. You could also use the paper that you have found, for example.
5. **[23.05]** Find an example application that could be scaled on different levels (VMs, pods, containers, etc.). The application should be deployable in the cloud and should have stream data processing interface. Identify its requirements on the data provided in stream, derive corresponding data layout for your load generator. It may also be necessary to adapt an application for the tool (e.g. to add some incoming data interface).
6. **[27.06]** Develop a tool to estimate the quality of the multilayered application.

**The input for the tool:**

* 1. an application to estimate (see 5);
  2. a configuration of autoscaling solutions to estimate (e.g. use specific autoscalers, use specific autoscaling rules, etc.);
  3. an application data to test the autoscaling;
  4. a dynamic load generation law (should be bound to different scenarios, e.g. scale in\scale out);
  5. metrics of quality for autoscaling solutions to estimate.

**The desired output:**Performance and cost of autoscaling versus configurations of autoscaling solutions with the current applications. For each configuration of autoscalers:

1. performance values for desired metrics of quality for current atoscaling configuration;
2. costs of autoscaling for current atoscaling configuration.

**The processing inside the tool:**

1. configuration of application using the autoscaling tools specified in the file;
2. deployment of the configured application in the cloud;
3. generation of the load according to the specified load generation law for different scenarios (at least: scale in, scale out, rapidly changing load);
4. measuring the performance on each scenario;
5. calculation the costs based on the VM types used;
6. creation of the performance graphs for different metrics and different autoscalers configurations;
7. offloading the collected performance and cost data to files.
8. **[11.07]** Estimate the quality of existing market-level autoscaling solutions based on identified metrics using the developed tool.  
   General testing approach: there are given QoS requirements for the application; the load is changed in such a way that QoS are not met and autoscaling is triggered; measure how long does it take to meet QoS requirements under new load (scale out); repeat experiment with low load (scale in).  
   **The testing should be conducted on a widespread configurations of layers and autoscalers**. E.g. pure VMs with cloud service provider’s own autoscaling solution, or add Kubernetes.
9. **[11.07]** Based on the results of the conducted experiments, compare individual autoscaling solutions and their widespread combinations. Which configuration works better? Which configuration is the worst? What are the main drawbacks of these autoscaling solutions and their combinations for the developed cloud application?
10. **[23.07]** Identify a model of an ideal autoscaling solution (e.g. with more tuning parameters and less impact on cost model), try to get rid of identified drawbacks.
11. **[23.07]** Write a paper with the results of the evaluation.

Development of a tool which estimates and analyzes the different configurations of existing cloud auto-scaling solutions in respect to performance and costs, resulting in the best suited configuration for the deployment of application along with the pros and cons of other configurations.

cost and performance of existing cloud auto-scaling solutions for multilayered cloud applications and analyzing different configurations of auto-scaling solutions