**Research proposal**

|  |
| --- |
| **Proposal Title:** SmartVM: A Multi-Layer Microservice-Based Platform for Deploying SaaS |
| **Keywords**: SaaS deployment, microservice, monitor and auto scale |
| **Research abstract and goals**  The project aims to achieve resource-optimized,on-demand dynamic scaling across multiple tenants and reduce costs in SaaS deployment. The project proposes a new platform named SmartVM providing microservice-oriented deployment kits to enable SaaS developer to create customize and deploy SaaS solutions in a multi-tier microservice-based manner, which can optimize the resources, scale in/out the underlying resources not only based on resource utilization but also the non-functional requirement such as timing constraint. |
| **Research content and technical approach**  The project proposes a new platform that can monitor resource usage and business time consuming to auto scale in/out microservice. In the project we create an experiment environment and the SmartVM prototype.  The prototype contains five components.  Business Microservice(BMS) implements the business logics.  API Microservice(AMS) implements the resource-aware library functions. Both BMS and AMS are deployed in docker containers.  Load Balancer is a gateway between users and the underlying applications. Global Monitor aims to connecting each distributed monitor to keep track of the status of AMS and BMS containers.  Distributed Monitor is deployed on each VM to monitor BMS and AMS in docker containers on that VM.  Scale Out Manager is connected to a GM to generate requested BMS and AMS containers.  A workflow is that the load balancer receives user requests and piles them into queues. Then requests are forwarded to the BMSs in different VMs through the dispatcher. BMS deliver the resource-aware tasks to AMS. Both BMS and AMS are monitored by DM. When resource usage or business time consuming exceed limits, DM will scale out relative AMS and BMS and report details to GM.  The difficulties are   * The difficulty of monitoring resource usage accurately and instantly. * The difficulty of auto scaling containers instantly. * The difficulty of connection between BMS, AMS, DM,GM across containers and distributed VMs. * The difficulty of load balancing between BMS and AMS |
| **Research Plan and Milestones**   1. Investigate microservice architecture and deploying microservice in docker 2. Create experiment AMS and BMS. 3. Develop load blancer 4. Develop distributed monitor and global monitor 5. Test the prototype on servers. |
| **Excepted outcomes**  The project will create a new platform for microservice-based SaaS deployment, which can be efficient scalability and minimizing deployment cost. |