Plan for Experimentation

Scenario 1: Preliminary for identifying research questions.

Purpose:

1. Test the efficiency of the algorithms, in terms of total power consumption, accumulated SLA violation, and average resource utilization (CPU, mem and net)
   1. Collect raw data from the First-fit in 100 runs
   2. Collect raw data from the balanced-fit in 100 runs
   3. Collect raw data from the min-power in 100 runs
   4. Collect raw data from the max-utilization in 100 runs
2. Compare the performance of the above algorithms. We might find that at different stages of the cloud operations, one algorithm might be better than the others. This finding will be served as the entry point for us to conduct the rest of the experiments meaningfully.
3. It is hard to measure the average resource utilization as some servers are idle and other are running. To give a reasonable and comparable measure of resource utilization between experimental runs, let Aui denotes the consumed resource of server i, and Api be the configured physical resources on the server i. Then the average resource utilization of the datacenter at time tj is expressed as where n is the number of active servers at time tj, and N is the total number of servers in the datacenter.

Configurations for scenario 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Rack-space: 8 | Total-racks: 6 | Simulation-time-unit: 5 | Total-services: 300 |
| Service-lifetime: [100, 300] | Random-seed: each run of the experiment must have a unique random seed so that t the results from each run will be different from others. Otherwise, the repeated runs will produce almost identical results. | | |
| CPU/Mem/Net-usage-distribution: beta[2, 4] (to be close to a Gaussian distribution) | Memory-access-ratio: beta[2, 4] | Service-generation-speed: 300 | Service-submission-strategy: ‘closest’ |
| Service-placement-algorithm:  All of them, one-by-one, except the ‘Random’ algorithm, which was made for demonstration only. | | Evaluation-stage: 2 | Scheduler-queue-capacity: 50 (This doesn’t affect anything as the implementation of this part hasn’t been completed.) |
| Scheduler-history-length: 5 (making this parameter a bigger value may results in a better result, but traded from large amounts of memory usage on your laptop.) | | Server-standby-strategy: all-off | Server-cpu/mem/net-utilization threshold: [20, 90] |
| Server-model: [2, 3, 4, 5] | Datacenter-level-heterogeneity? ‘OFF’ | Consolidation? ‘YES’ | Rack-level-heterogeneity? ‘YES’ |
| Consolidation-interval: 12 | Server-consolidation-strategy: ‘within-datacentre’ | Power-estimation-method: mean | Display-migration-movement? ‘OFF’ |
| Auto-migration? ‘YES’ | Auto-migration-strategy: ‘least-migration-number’ | Power-model-method: ‘Stepwise-simple-linear-regression’ | Show-trace: ‘OFF’ |

Scenario 2: Requested resources are less than the capacity of the datacenter.

Purpose: based on the findings discovered from the scenario 1, use the alternating approach for resource management. The results should be compared with the ones produced from scenario 1.

1. Test the efficiency of the alternating algorithm, in terms of total power consumption, accumulated SLA violation, and average resource utilization (CPU, mem and net)
   1. Collect raw data from the alternating algorithm, with time interval: {6, 12, 18, 24, 30, 36}. The numbers should correspond to the “simulation-time-unit”. Since we have set the value of the parameter to 5, i.e., 5 minutes per tick, the interval 6 corresponds to half an hour. We should exercise the number to a certain degree where the alternating algorithm starts to make no senses.
   2. Try to perform 100 runs for each interval.
   3. Use the same configurations used in Scenario 1.
   4. Compare the performance of the alternating algorithm with the intervals listed above. We might find that at certain stage of the operation, the algorithm with different intervals don’t not make much differences (in terms of power, resource, and sla), but has significant impact on the number of migrations. If that was observed (it should be), we will conduct further experiments as described below (point e).
   5. Starts with a number of parallel threads of simulation instance with different intervals (as used above). For example, from Scenario 1, we might have identified that the first-fit, balanced-fit and min-power were the candidates in the alternating algorithm; and from the Scenario 2 (point a) we might have identified that interval of 6, 12, and 24 were to be exercised based on the conclusion drawn from (point d), them the test here should have, at least 6 threads representation of the combination of the algorithm+interval. Again, use the best simulated results from the threaded simulation, and collect the raw data, finally compare it with the results collected from Point d.

Scenario 2: Requested resources are more than the capacity of the datacenter.

Purpose: this part test the robustness of the algorithm when the environment becomes hazardous.

Perform the same test as per described in Scenario 2 Point e.

NOTE: we assume the priority of selection has already defined: Power > SLA > Resource Util > Number of Migration. In the future work, we will model their relationships as a directed, weighted graph model.