Environment

1. Java8 + lambda
2. Play framework

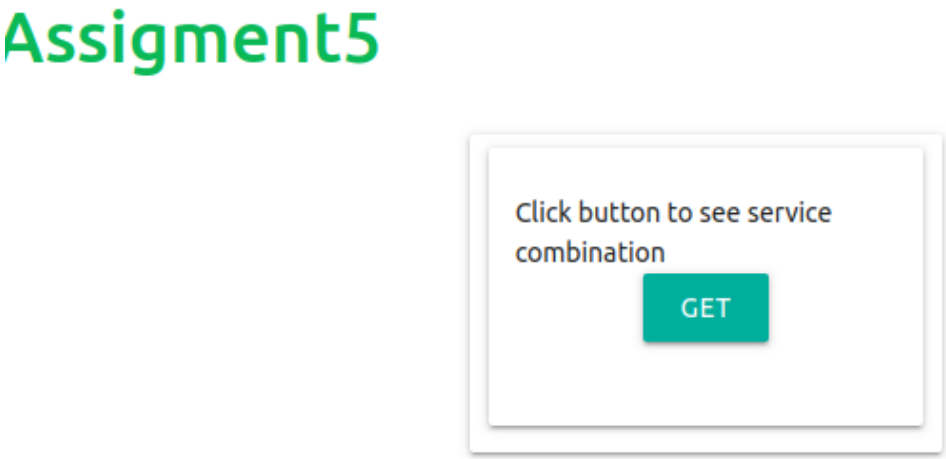
Step by Step setup

Set up backend server

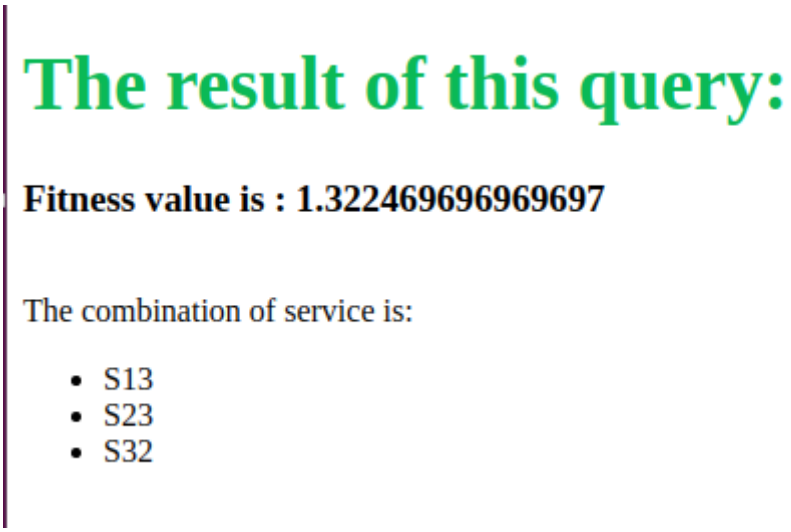
1. Find the `sc.xlsx` in `Backend` project, and move it to your own soc path, such as `/tmp/soc`

   If you decide to use other path, please change the variable `file` in `ReadServiceFeatures.java`

2. Start backend server on port 9000
3. Start frontend server on any point
4. Click the `Get Service` button to get the service combination



5. The query result is:



Requirements

1. You can either run the frontend and backend server to get the result or simply run `CalculateServiceCluster.java`
2. Based on my calculation, the fitness value is 1.322469696969697 and the service combination is S13, S23, S32 in the three clusters respectively.
3. Before doing the calculating, first normalized `time` and `cost` attributes. I calculate the value by applying this formula: `(currentCost-minCostOfAllClusters)/(maxCostOfAllClusters-minCostOfAllClusters)`, for the detail explanation, please go to `Implementation` part.
4. - Define an array with three element, every element represents a service cluster, and the value of each element will represent the server index in each cluster;
   - By evolving, the genes will have different values, I will get different combinations, and for each combination, I will calculate a fitness value;
   - The `Implementation` part explains the process in a detail way.

Implementation

1. Define the gene array that represent three clusters. The value of each element will represent service index in each cluster.

```
Gene[] sampleGenes = new Gene[3];
sampleGenes[0] = new IntegerGene(conf, 0, 4); // service cluster1
sampleGenes[1] = new IntegerGene(conf, 0, 2); // service cluster 2
sampleGenes[2] = new IntegerGene(conf, 0, 7); // service cluster 3
```

2. Calculate the service feature to get the output of S3 (`w3`):
   - Since S1 and S2 are sequential, I get the output(`w1`) of S2 by the following formula: `w1 = seq (s1, s2)`
   - Since the output of S2 and S1 are parallel, I get the input of S3, which is `w2`, by the following formula: `W2 = join (W1, s1)`
   - Since S3 and `w2` are sequential, I can get the output of S3 by doing `W3= seq (W2, s3)`
   - The major code is as following:

```
private static ServiceFeature getFeaturesByIndex(int sc1Index, int
sc2Index, int sc3Index) {
  List<ServiceFeature> cluster1Features = serviceFeatures.get(0);
  ServiceFeature serviceFeature1 = cluster1Features.get(sc1Index);
  List<ServiceFeature> cluster2Features = serviceFeatures.get(1);
  ServiceFeature serviceFeature2 = cluster2Features.get(sc2Index);
  //get the output of s2,  name W1, so W1 = seq (s1, s2)
  ServiceFeature w1 = seq(serviceFeature1, serviceFeature2);
  //get the input of s3,    name W2, so W2 = join (W1, s1)
  ServiceFeature w2 = join(w1, serviceFeature1);
  List<ServiceFeature> cluster3Features = serviceFeatures.get(2);
  ServiceFeature serviceFeature3 = cluster3Features.get(sc3Index);
  //get the output of s3,  name W3, so W3= seq (W2, s3)
  ServiceFeature w3 = seq(w2, serviceFeature3);
  return w3;
}
private static ServiceFeature seq(ServiceFeature serviceFeature1,
ServiceFeature serviceFeature2) {
```

```
  ServiceFeature serviceFeature = new ServiceFeature();
  Double reliability = Math.min(serviceFeature1.getReliability(),
serviceFeature2.getReliability());
  Double availability = Math.min(serviceFeature1.getAvailability(),
serviceFeature2.getAvailability());
  Double cost = serviceFeature1.getCost() + serviceFeature2.getCost();
  Double performance = serviceFeature1.getTime() +
serviceFeature2.getTime();
  ...
}
private static ServiceFeature join(ServiceFeature serviceFeature1,
ServiceFeature serviceFeature2) {
  ServiceFeature serviceFeature = new ServiceFeature();
  Double reliability = serviceFeature1.getReliability() *
serviceFeature2.getReliability();
  Double availability = serviceFeature1.getAvailability() *
serviceFeature2.getAvailability();
  Double cost = serviceFeature1.getCost() + serviceFeature2.getCost();
  Double performance = Math.max(serviceFeature1.getTime(),
serviceFeature2.getTime());
  ...
}
```

3. Calculate fitness value by applying the following formula to `w3`

```
fitness = COST_WEIGHT * serviceFeature.getCost() + RELIABLE_WEIGHT *
serviceFeature.getReliability() +
        TIME_WEIGHT * serviceFeature.getTime() + AVAILABLE_WEIGHT *
serviceFeature.getAvailability();
```