Neat子问题算法

├─overload

│ ├─trivial

│ │ ├─never\_overloaded\_factory: never considers the host overload

│ │ ├─threshold\_factory: current CPU utilization with static threshold algorithm

│ │ └─last\_n\_average\_threshold\_factory: averaging CPU utilization with static threshold algorithm(THR)

│ ├─statistics

│ │ ├─loess\_factory: loess(局部加权回归) based overload detection algorithm

│ │ ├─loess\_robust\_factory: robust loess(局部加权回归) based overload detection algorithm

│ │ ├─mad\_threshold\_factory: MAD based utilization threshold algorithm

│ │ └─iqr\_threshold\_factory: IQR based utilization threshold algorithm

│ ├─otf: OTF algorithm with limiting and migration time

│ ├─mhod

│ │ ├─mhod\_factory(core.py): create MHOD algorithm

│ │ ├─bruteforce.py: functions for solving NLP problems using brute-force method

│ │ ├─l\_2\_states.py: L functions for the 2 state configuration of the MHOD algorithm

│ │ ├─multisize\_estimation.py: multisize sliding window workload estimation functions

│ │ └─nlp.py: functions for defining the NLP problem of the MHOD algorithm

│ └─slr

├─underload

│ ├─trivial

│ │ ├─always\_underloaded\_factory: always considers the host underload

│ │ ├─threshold\_factory: current CPU utilization with static threshold algorithm

│ │ └─last\_n\_average\_threshold\_factory: averaging CPU utilization with static threshold algorithm(THR)

│ ├─statistics

│ │ ├─loess\_factory: loess(局部加权回归) based overload detection algorithm

│ │ ├─loess\_robust\_factory: robust loess(局部加权回归) based overload detection algorithm

│ │ ├─mad\_threshold\_factory: MAD based utilization threshold algorithm

│ │ └─iqr\_threshold\_factory: IQR based utilization threshold algorithm

│ └─slr

├─vm\_selection

│ └─algorithms

│ ├─random\_factory: select a random VM

│ ├─minimum\_utilization\_factory: select VM with minimum CPU utilization

│ ├─minimum\_migration\_time\_factory: select the VM with the minimum RAM usage

│ ├─minimum\_migration\_time\_max\_cpu\_factory: select the VM with minimum RAM

│ │ and maximum CPU utilization

│ └─minimum\_migration\_time\_min\_cpu\_min\_ram\_factory: select the VM with minimum migration,

│ maximum CPU utilization and minimum RAM

└─vm\_placement

├─best\_fit\_decreasing\_factory: bin packing

├─power\_aware\_best\_fit\_decreasing\_factory

└─improved\_power\_aware\_best\_fit\_decreasing\_factory

