Kubernetes Container Scheduling Strategy (KCSS)

Tarek MENOUER and Patrice DARMON

Umanis Research & Innovation

GOALS

- ❑ The KCSS is proposed to optimize the scheduling of several containers submitted online by users. The goal is to improve the performance in terms of computing time, power consumption and the quality of services of the applications in the containers as much as possible.

NOVELTY

- ❑ The novelty of the KCSS consist to select for each newly submitted container the node that has a good compromise between hybrid criteria related to the user needs and the state of the cloud infrastructure.

PRINCIPLE

- ❑Apply the multicriteria Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) algorithm
- ❑Choose the node that has a good compromise between six criteria:
  - ➢ Duration of transmitting the image selected by the user on the container;
  - ➢ Number of available CPUs in each node;
  - ➢ Size of available memory in each node;
  - ➢ Size of the available storage disk in each node;
  - ➢ Power consumption of each node;
  - ➢ Number of running containers in each node.

TOPSIS ALGORITHM

- ❑ Fill a Decision Matrix (*DM*) of *n* lines (number of nodes) and c columns (number of criteria). Each value f_{ij} in *DM* represents the value of the node n_i in criterion j.
- ❑Calculate the Normalized Decision Matrix (*NDM*). The normalized value r_{ij} is determined as following:

$$r_{ij}=f_{ij}/\sqrt{\sum_{i=1}^{n} f_{ij}^2}, \text{ for i = 1, ..., n and j = 1, ..., c.}$$

- ❑Calculate the Weighted Normalized Decision Matrix (*WNDM*). The weighted normalized value v_{ij} is determined as following:

$$v_{ij} = w_i \times r_{ij} \text{ for i = 1, ..., n and j = 1, ..., c.}$$

w_i is the weight of the j^{th} criterion, and the $\sum_{j=1}^{c} w_j = 1$.

- ❑Determine the best (A^+) and the worst (A^-) solutions.

$A^+ = \{v_1^+, \dots, v_c^+\}$ ={(max(v_{ij} | i=1, ..., n) | $j \in I'$), (min(v_{ij} | i=1, ..., n) | $j \in I''$)}

$A^- = \{v_1^-, \dots, v_c^-\}$ ={(min(v_{ij} | i=1, ..., n) | $j \in I'$), (max(v_{ij} | i=1, ..., n) | $j \in I''$)}

I' is associated to the criteria having the positive impact, and I'' is associated to the criteria having the negative impact.

- ❑Calculate the Separation Measures (SM), using the n-dimensional Euclidean distance. The separation of each node from the best solution is given by the SM^+ formula. The separation of each node from the worst solution is given by the following SM^- formula.

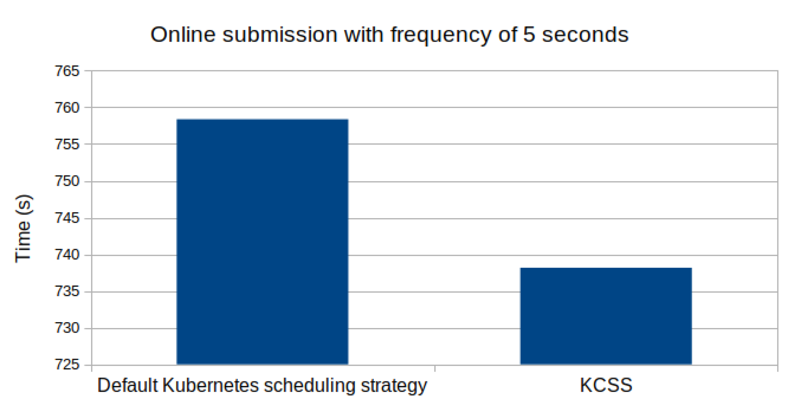
$$SM_i^- = \sqrt{\sum_{j=1}^{c}(v_{ij} - v_j^-)^2}, \text{ for } i = 1, \dots, n$$

$$SM_i^+ = \sqrt{\sum_{j=1}^{c}(v_{ij} - v_j^+)^2}, \text{ for } i = 1, \dots, n$$

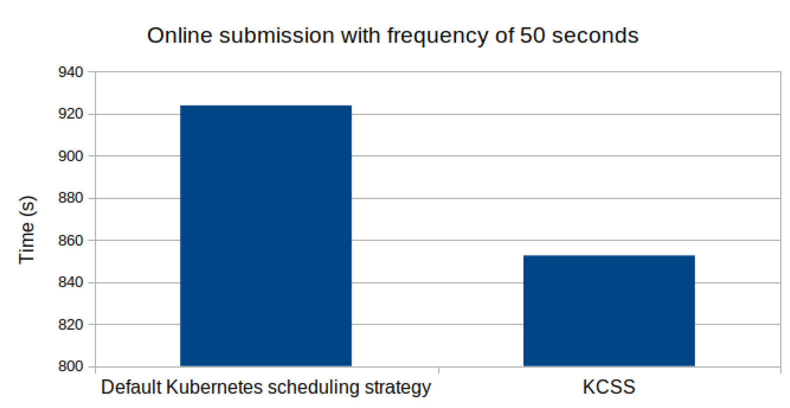
- ❑Calculate the Relative Closeness (RC) to the best solution. For each node n_i, the RC_i is defined by the following RC_i formula.

$$RC_i = SM_i^- / (SM_i^+ + SM_i^-), \text{ for } i = 1, \dots, n.$$

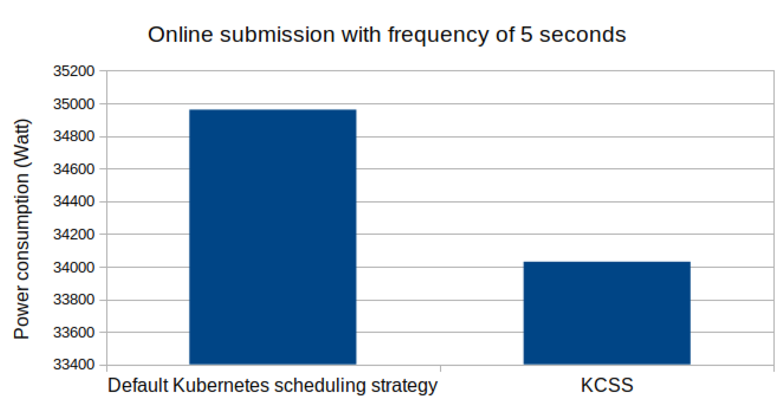
EXPERIMENTAL EVALUATION



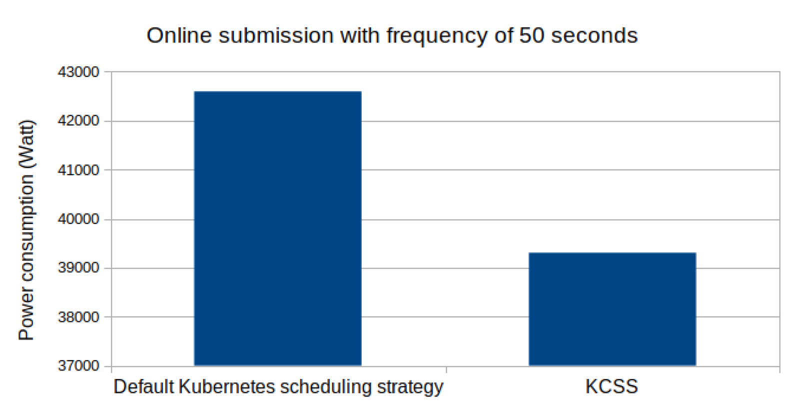
Comparison between the computing time (makespan) obtained by the KCSS and the default kubernetes strategy to schedule 45 containers submitted online with frequency of 5 seconds.



Comparison between the computing time (makespan) obtained by the KCSS and the default kubernetes strategy to schedule 45 containers submitted online with frequency of 50 seconds.



Comparison between the power consumption obtained by the KCSS and the default kubernetes strategy to schedule 45 containers submitted online with frequency of 5 seconds.



Comparison between the power consumption obtained by the KCSS and the default kubernetes strategy to schedule 45 containers submitted online with frequency of 50 seconds.

RESULT

- ❑ Experiments with KCSS are performed in Virtual Wall testbed.
- ❑ In our experimental environment, a speedup varied between 2% and 8% is obtained with KCSS in terms of:
  - ➢Computing time (makespan);
  - ➢Power consumption.

CONCLUSION

- ❑ KCSS select for each container the node with a good compromise between several criteria.
- ❑ KCSS is implemented in GO language inside Kubernetes with a minimum of change.