Review Open Call 7 Kubernetes Container Scheduling Strategy (KCSS)

Tarek MENOUER

Umanis Research & Innovation (France)

FEC7 & Experiment Review

01/04/2020

Outline

- Experiment description
- Project results
- Business impact
- Feedback

Kubernetes Container Scheduling Strategy

KCSS

Experiment description

Experiment description (1/4)

CLOUD COMPUTING & CONTAINERS:

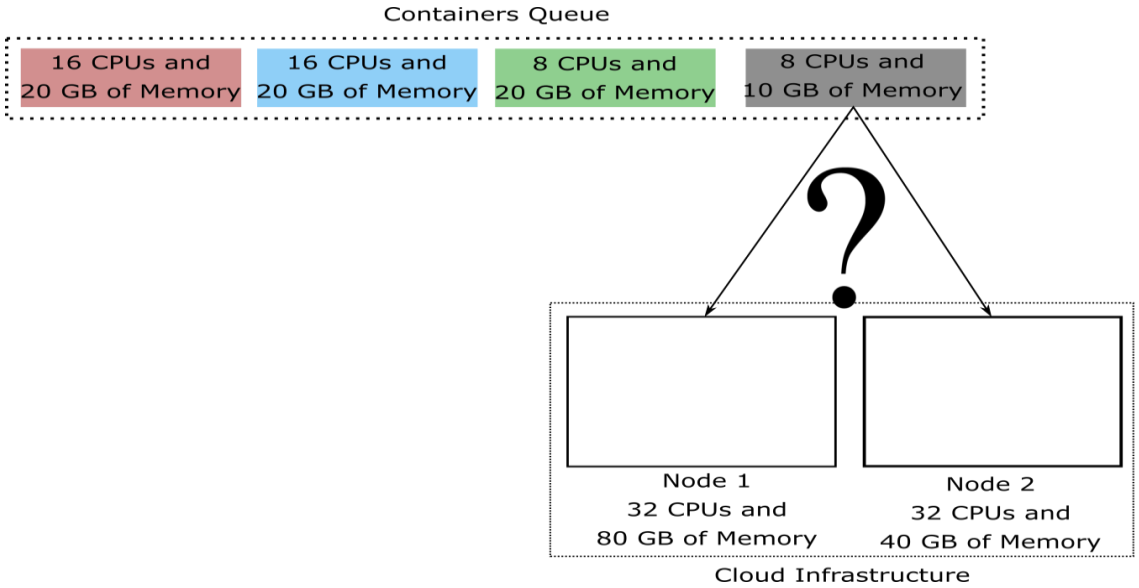
- Cloud Computing aims to offer distributed, virtualized, and elastic resources as utilities to end users
- Different forms of resources exists: VMs, Containers, Bare-metal, …
- Containers technology is widely used in the industrial field
  - It has a much faster launching and termination time
- Several containers scheduling framework exist:
  - Docker SwarmKit, Kubernetes, MESOS, …
  - Each framework uses a particular scheduling strategy
  - ✗ The majority of scheduling strategy are based only **on one criterion**

SOLUTION

✓ We propose a new **Kubernetes Container Scheduling Strategy (KCSS)** which select for each container node basing on a **multicriteria.**

Experiment description (2/4)

GOAL OF OUR PROJECT



SELECT THE BEST NODE THAT MUST EXECUTE A CONTAINER AND WHICH OPTIMIZE THE GLOBAL SCHEDULING IN THE CLOUD INFRASTRUCTURE

Experiment description (3/4)

PRINCIPAL

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

Experiment description (4/4)

DESCRIPTION OF TOPSIS ALGORITHM

- Build an outranking between different nodes
- Comparison pair by pair of possible nodes along the different *n* criteria

IMPLEMENTATION

- The code of our strategy is implemented in GO language inside kubernetes with a minimum of change → can be used directly with the next version of kubernetes

BENEFITS OF OUR PROJECT

- Reduce the makespan (computing time).
- Reduce the power consumption.
- Improve the quality of services of the applications in the containers as much as possible.

Project results

Project results (1/3)

TEST ENVIRONMENT

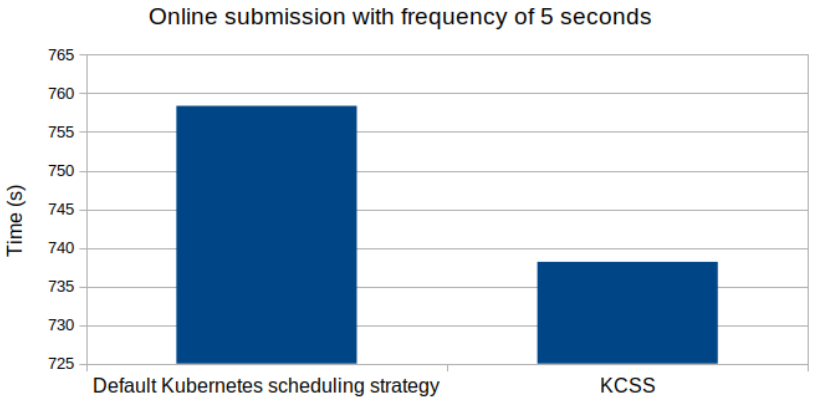
- Our project is realized inside the **Virtual Wall** testbed

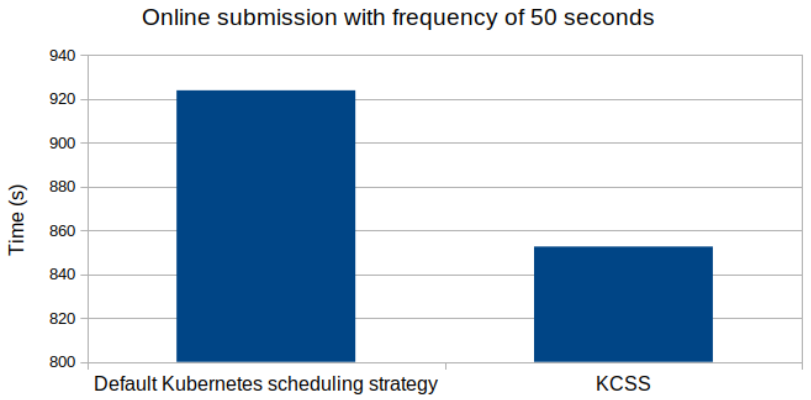
USE CASE

- 45 containers submitted with execution time = 120s for each container
- We suppose that we have three categories of containers:
  - **Small container** which need 4 CPUs, 5 GB of memory and 2 GB of storage disk,
  - **Average container** which need 8 CPUs, 10 GB of memory and 4 GB of storage disk,
  - **Large container** which need 16 CPUs, 20 GB of memory and 8 GB of storage disk.
- 2 ways to submit on line containers:
  - With frequency of 5 seconds → each 5 seconds 3 containers are submitted.
  - With frequency of 50 seconds → each 50 seconds 3 containers are submitted.

Project results (2/3)

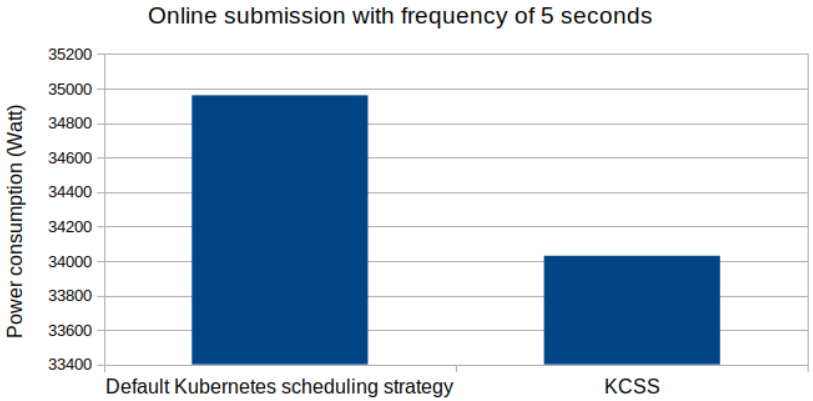
COMPARISON BETWEEN THE MAKESPAN OBTAINED BY THE KCSS AND THE DEFAULT KUBERNETES SCHEDULING STRATEGY

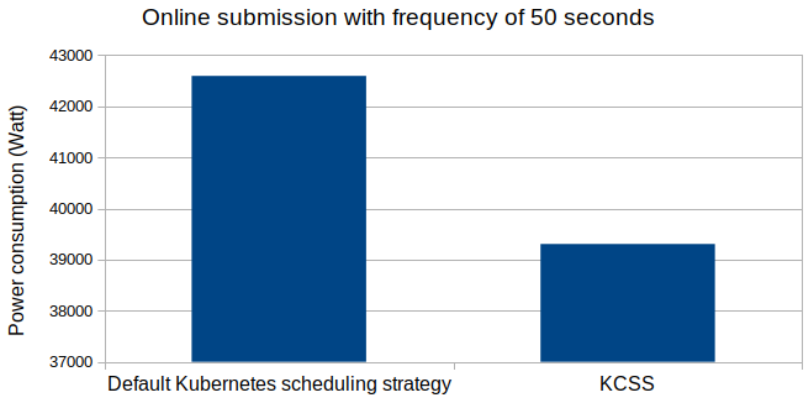




Project results (2/3)

COMPARISON BETWEEN THE POWER CONSUMPTION OBTAINED BY THE KCSS AND THE DEFAULT KUBERNETES SCHEDULING STRATEGY





Business impact

Business impact (1/3)

IMPACT ON OUR BUSINESS

- The Kubernetes Container Scheduling Strategy (KCSS) is proposed to improve the performance of each company that use Kubernetes as a container scheduling framework.
- For our company, the KCSS will be used very soon as a default container scheduling strategy in the Teambuilder project.
    - TeamBuilder is a tool proposed by Umanis company and which automatically matches the profiles of employees with the available projects basing on some services encapsulated in containers.
- Thanks to the result obtained by our project, recently, we started to discuss with some customers who is very interested in use our KCSS:
    - Offer our solution in the form of a paid service.
    - Adapt our solution according to the need of customers in form of a paid service.
    - Offer to customers our expertise in the field of containers scheduling.

Business impact (2/3)

HELP FROM FED4FIRE

Fed4FIRE allowed us to give several helps:

- Have access to a real cloud infrastructure at no cost,
- Experiment our approach in a real cloud infrastructure,
- Compute the cost we have gained with our approach,
- Receive funds to carry out our research and our project,
- Have new ideas and perspectives basing on the results obtained with KCSS.

Business impact (3/3)

WHY DID YOU COME TO FED4FIRE ?

To perform our experiments, we chose Fed4FIRE:

- Funding our research,
- The availability of resources,
- Easy setup of experiments,
- Tools offered,
- Combining infrastructures.

Feedback

Feedback (1/4)

USED RESOURCES

In our experiments:

- Virtual Wall (imec) testbed was the ideal solution.
  - Pcgen3 nodes and Pcgen4 nodes.
- We have booked, several time, an infrastructure composed from 4 nodes to implement and test our approach.
  - Each node has 32 or 24 CPUs with 48 or 24 GB of memory.
- Each time we have booked resources we have used it.

Feedback (2/4)

USED TOOLS

In our experiments,

- JFed tool
  - Easy to install
  - Documentation is well written
- JFed command Line (CLI)
  - Easy to use
- Kubernetes framework
  - To schedule containers
- CloudSim Plus tool
  - To simulate the power consumption of nodes

kubernetes

CloudSim Plus

Feedback (3/4)

THE BENEFIT OF THE FED4FIRE

According to our experience, we find that :

- Fed4FIRE is a success at all levels,
- The support is excellent and the documentation is well writing,
- I really appreciated the very fast reactivity of email exchanges about all subjects,
- The administrative part went very well,
- Diversity and the availability of resources which is a good point to run the experiments,
- Infrastructure is very easy to use,
- Possibility to combine several infrastructures.

Feedback (4/4)

COMMENTS

- Propose to organize each month an online meetup.
  - Discuss online between the Fed4FIRE experimenters about the progress of their projects and the problem encountered
    - Collaborate between experimenters if it is possible …
- Add in the Virtual WALL testbed a Wattmeter to compute the power consumption.

SCIENTIFIC RESULT OF OUR PROJECT

- The approach proposed in stage 1 is under review in an international conference.


Co-funded by the European Union


Co-funded by the Swiss Confederation

This project has received funding from the European Union's Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.

THANK YOU FOR YOUR ATTENTION

WWW.FED4FIRE.EU