

FastFlow + Docker

We are investigating the possibility to execute FastFlow programs within the Docker platform.

The goal is to test:

- **elasticity**: the resources (CPU , Memory) needed by FF program executed in a Docker container can be dynamically allocated (e.g. increasing the cores needed by a FF program, changing the graph topology- increasing the workers in a farm)
- **multi-tenant**: two programs running on the same host should not interfere with each others.

Thread-affinity in Docker

Docker is able to guarantee *thread-affinity* in order to authorize a given application to access only some CPUs (via `--cpuset-cpus` command)

The test uses the image `agileek/cpuset-test` that it is configured to run the `cpuburn` script for load the CPUs of a machine.

Starts the container with name `test` assigning only the CPU 0.

```
docker run -ti --cpuset-cpus=0 --name test agileek/cpuset-test
```

With the `update` command it is possible to change the number of CPUs at runtime.

```
docker update --cpuset-cpus=0,2 test
```

IN order to test if a container detects the CPUs also with the `--cpuset-cpus` flag.

```
docker run -ti --cpuset-cpus=2 ubuntu cat /proc/cpuinfo |  
grep processor
```

The output prints the number of processors:

```
processor      : 0  
processor      : 1  
processor      : 2  
processor      : 3
```

Conclusion

1. *elasticity*: (resolved) with the `update` command and the `--cpuset-cpus` option the resources (e.g. number of CPUs) of a FF program executing in a Docker container can be dynamically allocated.
2. *multi-tenant*: (resolved) two FF programs can be executed in two Docker containers mapping only onto subset of CPUs.

Update options

Name, shorthand	Default	Description
<code>--blkio-weight</code>	0	Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)
<code>--cpu-period</code>	0	Limit CPU CFS (Completely Fair Scheduler) period
<code>--cpu-quota</code>	0	Limit CPU CFS (Completely Fair Scheduler) quota
<code>--cpu-rt-period</code>	0	Limit the CPU real-time period in microseconds
<code>--cpu-rt-</code>		Limit the CPU real-time runtime in

<code>runtime</code>	<code>0</code>	microseconds
<code>--cpu-shares, -c</code>	<code>0</code>	CPU shares (relative weight)
<code>--cpuset-cpus</code>		CPUs in which to allow execution (0-3, 0,1)
<code>--cpuset-mems</code>		MEMs in which to allow execution (0-3, 0,1)
<code>--kernel-memory</code>		Kernel memory limit
<code>--memory, -m</code>		Memory limit
<code>--memory-reservation</code>		Memory soft limit
<code>--memory-swap</code>		Swap limit equal to memory plus swap: '-1' to enable unlimited swap
<code>--restart</code>		Restart policy to apply when a container exits