Mathematical Modeling of Cluster Dynamics

Alexandra De Cecco, Guillaume Dufour, David Sanchez

Mathematical Modeling of Cluster Dynamics

Alexandra De Cecco[a], Guillaume Dufour[b], David Sanchez[c,*]

[a] *ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France*
[b] *ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France*
[c] *Institut de Mathématiques de Toulouse , UMR5219 Université de Toulouse, CNRS, INSA, F-31077 Toulouse, France*

Abstract

This paper deals with the modeling and the numerical simulation of computational cluster networks (grid) dynamics and is more precisely focused on the management of the computational load offered by transferring jobs from one cluster to another. In order to tackle the complexity arising from dealing with the sheer number of processes existing in the system, we develop a comprehensive model of the global population of jobs using the kinetic theory. It takes into account the main characteristics of the clusters, their interactions and allows for a thorough description of the jobs. We then study the resulting system of macroscopic conservation laws and illustrate numerically its ability to capture some interesting behaviors of the grid. This model allows for real-time insight of the performance of a given policy for load management in the grid.

Keywords: Fluid models, Kinetic models, Computational Grid, Numerical simulation, Cluster network
2000 MSC: 35Q68, 35Q94, 68M10, 68M20, 35F50

1. Introduction

According to [1], the electric consumption of the information technology raised to 270 TWh in 2012 which is roughly equivalent to 1.4% of the worldwide electrical consumption while the complete Information and Communication Technology sector (excluding manufacturing) accounts for 4.7% of it. Moreover, the data center power needs increased annually by 5% between 2006 and 2012 (see also [2]). For the year 2030, the data centers alone may use between 3% (best case) and 13% (worst scenario) of the global electricity production [3]. We are interested here in the management of clusters running in several data centers and in the improvement of their energetic consumption while maintaining a certain quality of service. There are currently many ways to limit their energetic impact [4]. One way consists in enhancing the use of the clusters particularly thanks to a cooperation between different clusters. There exist different policies:

*Corresponding author
URL: alexandradececco@gmail.com (Alexandra De Cecco), Guillaume.Dufour@onera.fr (Guillaume Dufour), david.sanchez@insa-toulouse.fr (David Sanchez)

some are based on an *a priori* study of the state of the cluster network before launching any job on it [5]; others are based on a dynamic management of the clusters *via* the displacement of services or jobs between computers or clusters. These movements are facilitated by the use of virtual machines in which the jobs are encapsulated [6]. However, the evaluation of the quality of these policies is particularly difficult through the use of direct methods [7]. Indeed, the complexity increases greatly as we are reaching exa-scale systems and other methods are required in order to not treat individually each of the millions or billions of processes [8]. We propose here an evaluation method based on a macroscopic point of view where the value of interest is the local density of jobs rather than the individual jobs themselves.

These macroscopic representations are already widely used in other application domains where the sheer number of individuals makes it difficult to run a complete simulation (e.g. dynamics of animal swarms, crowd dynamics, etc.). Dealing with these representations involve a rather large panel of mathematical methodology which are the subject of several papers [9, 10, 11]. A comprehensive survey of these methods can for example be found in [12]. In the field of computer networks, this kind of approach was first referenced as fluid methods and is based on a set of coupled Ordinary Differential Equations (ODE) [13, 14, 15], each one describing the load on a given network node based on only two parameters (the input and the output rate). An extension of this kind of models, using Partial Differential Equations (PDE) has also been introduced for a group of processes with heterogeneous parameters, such as Mice and Elephant processes in TCP/IP protocol [16]. However, these models are provided thanks to conservation considerations and are phenomenological. Thus, their generalization or the inclusion of more parameters require a complete rethinking of the model.

In this article, we propose a set of conservation laws describing the evolution of local density of jobs on a network derived from a proper kinetic model of the cluster dynamics. The idea is to use the microscopic / mesoscopic / macroscopic formalism rather than an *ad hoc* model of the cluster load. Taking as a basis the individual based model of the jobs and of their movements between the clusters, we define a density distribution function for the jobs, the main part of the work being devoted to the derivation of the corresponding kinetic equation (or system of equations) solved by this distribution function. We are then finally able to obtain a macroscopic model that describes the evolution of the cluster network. The foundations of this model have been laid in [17] and developed in the thesis of De Cecco [18]. One main difference with the fluid models derived in the context of crowds or swarms of animals where the amount of interactions with the other participants (be it collisions or repulsion forces) lead to a global dynamics of the swarms, is that in the context of computer networks, the dynamics of one element is uniquely decided by the environment. The interactions between these elements occur through their own impact on the environment so that no collision operator has to be considered.

We will detail in Section 2 a microscopic description of the clusters and the jobs running on them and highlight the asymptotics allowing us to derive the fluid model in Section 3. In Section 4 we mathematically justify the models we have obtained and finally, we present in Section 5 some numerical illustrations on simple cases to highlight the potential benefits of different job management policies.

2. A comprehensive cluster description

2.1. Cluster

We consider in this study a network of several data centers which will be modeled as a network of C clusters (denoted by C_j, $j \in \{1, \dots, C\}$). This network is assumed to be managed

for each cluster by a centralized middle-ware which decides whether a given job running on a cluster will still be executed on site or will be sent on another cluster in order to improve a cost (computational time, energetic, financial,... or any combination of them). To evaluate the possible gain arising from moving a job from one cluster to another, each cluster C_j will be characterized by a small number of parameters such as its performance index $v_j \in \mathbb{R}^+$ (its computational speed), its energetic power $Z_j(t,q) \in \mathbb{R}^+$, depending on the time t and the load of the cluster q (it could depend on more parameters such as the number of processors used), its working cost $C_j(t) \in \mathbb{R}^+$, (rental cost for example, possibly depending on the number of processors/cores used by a job), its total number of processors $\pi_j \in \mathbb{N}^*$, its maximal number of simultaneous jobs $T_j \in \mathbb{N}^*$.

In our case, we assume each clusters to be homogeneous and to be defined only by its number of processors. Moreover, the transfer time between each interconnected cluster of the network will be defined thanks to a matrix of transfer time $\tau = (\tau_{jk})_{j,k\in\{1,..,C\}} \in M_C(\mathbb{R}^+)$ where τ_{jk} is the time needed to transfer a job from the cluster C_j onto the cluster C_k. Given that there is no transfer between a cluster and itself and that all connections are bidirectional, we assume that for each j, $\tau_{jj} = 0$ and that τ is symmetric.

2.2. Jobs

We consider that N jobs, denoted by J_i, $i \in \{1,\dots,N\}$ are executed on the network and that there is no deadline constraint. In the current model, interactive applications or services are not included. Each one is described by its size $s_i \in \mathbb{R}_+^*$ (the memory size occupied by the job on a cluster), the remaining load of computations required to complete the job $q_i(t) \in \mathbb{R}_+^*$, its position $P_i(t) \in \{C_1,..,C_C\}$ in the network, *i.e.* the cluster on which the job is executed at time t, the minimum number of processors required to perform the job $p_i \in \mathbb{N}^*$, the age of the job $a_i(t) \in \mathbb{R}^+$ and its waiting time $\theta_i(t) \in \mathbb{R}$. We introduce this time to take into account the transfer time of a job. When the job is moved from a cluster to another, the resources on the arrival cluster are reserved for the incoming job and will thus not be available for new jobs. We assume that the job is transferred immediately but that it arrives on the new cluster with the waiting time θ_i equal to the transfer time needed to transfer its data through the network.

In this study we do not use the parameters s_i and p_i and a_i. The transfer time between clusters are given and do not depend on the memory size of the job.

The decision to move a job from a cluster to another at a time t depends on a cost function. It can for example be the remaining execution time for a job actually located on the cluster C_j if moved to the cluster C_k,

$$t^{exe}(q_i(t), P_i(t) = C_j, C_k) = \frac{q_i(t)}{v_k} + \tau_{jk}\,,\ \forall (j,k) \in \{1,..,C\}^2,$$

sum of the transfer time from C_j to C_k and the required time to execute the job on the cluster C_k, the associated energetic consumption,

$$K^{exe}(q_i(t), C_k) = \int_0^{\frac{q_i(t)}{v_k}} Z_k(t,q)\,dt\,,\ \forall k \in \{1,..,C\},$$

the working cost of the material...

The functional to minimize will be a weighted combination of the remaining execution time and of the energy consumption:

$$K(q_i(t), C_k) = c_t\, t^{exe}(q_i(t), C_j, C_k) + c_e\, K^{exe}(q_i(t), C_k)\,,$$

with the weights c_t and $c_e \in \mathbb{R}^+$ such that $c_t + c_e = 1$.

A job is transferred to a cluster C_{k^*} (arbitrarily chosen if there is not uniqueness) if it minimizes its cost function. This cluster may eventually be the one on which the job is currently executed ($P_i(t) = C_{k^*}$). A decision function Dec is associated to the cost function such that

$$\mathrm{Dec}(t, q_i(t), P_i(t), P) = \begin{cases} 1 & \text{if } P_i(t) \neq C_{k^*} \text{ and } P = C_{k^*}\,, \\ 0 & \text{else.} \end{cases}$$

The middle-ware acts like a black box that sends the result 1 if the current job has to be moved from a cluster to another one and 0 otherwise. We introduce τ_{pr} the time taken to compute the decision function for a job. It is supposed to be identical for all jobs. We assume that the middle-ware tests a job after another continuously in an infinite loop. We give a number $\alpha_i \in \{1, .., N\}$ to each job that determines the order in which each job will be tested so that the time needed to test the job J_i is $\alpha_i \tau_{pr}$. We moreover assume that if the remaining load of a job is small enough, it will never be moved so the function Dec will be equal to zero for such a job.

We then obtain the following equations for each job:

$$q_i'(t) = -v_{P_i(t)}\ \mathbf{1}_{q_i(t)>0}\ \mathbf{1}_{\theta_i(t)\leq 0}\,, \tag{1}$$

i.e. the remaining workload q_i of the job decreases linearly until it finishes ($q_i = 0$) if it is not being transferred from a cluster to another ($\theta_i = 0$);

$$\theta_i'(t) = -\mathbf{1}_{\theta_i(t)>0} + \sum_{\substack{k=1 \\ k\neq j}}^{C} \tau_{jk}\ \mathrm{Dec}(t, q_i(t), C_j, C_k)\ \delta_{\theta_i(t)=0}\,, \tag{2}$$

i.e. the transfer time θ_i decreases linearly during the transfer until the execution begins ($\theta_i = 0$) or it jumps from 0 to τ_{jk} if the job is transferred from C_j to C_k.

Remark 1. *We remark that θ_i (related to the transfer) and q_i (related to the execution) do not evolve simultaneously. We also allow $\theta_i(t)$ to be negative. It will never happen in practice but this technical assumption will facilitate the derivation of the kinetic model in the following. Finally, let us note that a job waiting on a cluster is not tested until its waiting time is* 0 *which is relevant since the job is currently transferred.*

3. Toward a fluid model

3.1. Defining a distribution function

To obtain the global behavior of the jobs on the network without studying them individually, we introduce the distribution function of jobs f, valued in $\mathbb{R}^C$, whose each component f^j is the local density of jobs at load q and waiting time θ on the cluster C_j:

$$f(t, P, q, \theta) = \left(f^j(t, q, \theta)\ \delta_{P=C_j}\right)_{j\in\{1,..,C\}}$$

with

$$f^j(t, q, \theta) = \frac{1}{T_j} \sum_{\substack{i=1 \\ P_i(t)=C_j}}^{N_j} \delta_{q=q_i(t)}\ \delta_{\theta=\theta_i(t)}$$

and N_j the number of jobs on the cluster C_j. Then $\int_{q,\theta} f^j\, dqd\theta = N_j/T_j$ is the filling rate of the cluster C_j.

We now study the evolution of f^j between the times t and $t+\delta t$ (with $\delta t > 0$) in the distributional sense. Let $\Phi = (\phi_j)_{1\le j\le C} \in \left(\mathcal{D}(\mathbb{R}_+^*)_q \otimes \mathcal{D}(\mathbb{R})_\theta\right)^C$ be a test function associated to the whole network:

$$\Phi(P,q,\theta) = \sum_{j=1}^{C} \phi^j(q,\theta)\, \mathbf{1}_{P=C_j} \qquad \text{then} \qquad \left\langle f^j(t)\,,\, \phi^j \right\rangle = \frac{1}{T_j} \sum_{\substack{i=1 \\ P_i(t)=C_j}}^{N_j} \phi^j\left(q_i(t),\theta_i(t)\right)$$

Along the evolution between t and $t+\delta t$, either a job is moved from one cluster to another (case B) or it stays in place (case A).

$$\begin{aligned} &\left\langle f^j(t+\delta t) - f^j(t), \phi^j \right\rangle = A + B \\ =&\frac{1}{T_j}\left\{ \sum_{\substack{i=1 \\ P_i(t+\delta t)=P_i(t)=C_j}}^{N_j} + \sum_{\substack{i=1 \\ P_i(t+\delta t)\neq P_i(t)}}^{N_j} \right\} \left(\phi^j\left(q_i(t+\delta t),\theta_i(t+\delta t)\right) - \phi^j\left(q_i(t),\theta_i(t)\right)\right) \end{aligned} \tag{3}$$

By distinguishing whether the job is waiting or executing between t and $t+\delta t$ (or partially waiting then executing, executing and finished or waiting, executing and finished) we obtain

$$\begin{aligned} A = \quad & -v_j\, \delta t\, \langle f^j(t)\,,\, \partial_q\, \phi^j\, \mathbf{1}_{q>v_j\delta t}\, \mathbf{1}_{\theta\le 0}\rangle && \text{[jobs in execution state]} \\ & -\delta t\, \langle f^j(t)\,,\, \partial_\theta\, \phi^j\, \mathbf{1}_{q>0}\, \mathbf{1}_{\theta>\delta t}\rangle && \text{[jobs in waiting state]} \\ & -\langle \theta\, f^j(t)\,,\, \partial_\theta\, \phi^j\, \mathbf{1}_{0<\theta\le\delta t}\, \mathbf{1}_{q_i(t)>v_j(\delta t+\theta)}\rangle && \text{[jobs which waited during } \theta\ldots \\ & -v_j\, \langle(\delta t-\theta)\, f^j(t)\,,\, \partial_q\, \phi^j(q,0)\, \mathbf{1}_{0\le\theta\le\delta t}\, \mathbf{1}_{q>v_j(\delta t+\theta)}\rangle && \text{and were executed during } \delta t-\theta] \\ & -\langle f^j(t)\,,\, \phi^j\, \mathbf{1}_{q\le v_j\delta t}\, \mathbf{1}_{\theta\le 0}\rangle && \text{[jobs ended during } \delta t\ldots \\ & -\langle f^j(t)\,,\, \phi^j\, \mathbf{1}_{q\le v_j(\delta t-\theta)}\, \mathbf{1}_{0<\theta\le\delta t}\rangle && \text{or waited during } \theta \text{ before ending]} \\ & + O(\delta t^2). \end{aligned} \tag{4}$$

To describe simply the term B and avoid any redundancy with the terms presented in A we assume that δt is small enough so that a job can not be displaced and executed during the same time interval. This leads to

$$\delta t < \min_{j,k\ j\neq k} \tau_{jk}\,.$$

This implies that we only have to look either at the jobs arriving at the cluster C_j (case B_1) or leaving it (case B_2) during the time interval δt. We also impose that

$$\alpha_i \tau_{pr} \le \delta t, \forall i \in \{1,..,N\}\,,$$

so that the job has been tested between t and $t+\delta t$ by the middle-ware. We obtain

$$\begin{aligned} B = B_1 + B_2 = \quad \frac{1}{T_j}\Bigg[& \sum_{\substack{i=1 \\ P_i(t+\delta t)=C_j\neq P_i(t)=C_k}}^{N_j} \phi^j\left(q_i(t+\delta t),\theta_i(t+\delta t)\right) \\ & - \sum_{\substack{i=1 \\ P_i(t)=C_j\neq P_i(t+\delta t)=C_k}}^{N_j} \phi^j\left(q_i(t),\theta_i(t)\right) \Bigg], \end{aligned}$$

where C_k is the cluster from which the job is either arriving or leaving. The term B_1 is the density of jobs arriving on the cluster C_j at time $t + \alpha_i \tau_{pr}$ from the clusters C_k with $k \neq j$. They arrive with a waiting time equals to τ_{kj}:

$$
\begin{aligned}
B_1 = & \sum_{\substack{k=1 \\ k\neq j \\ P_i(t+\delta t)=C_j}}^{C} \frac{1}{T_j} \sum_{\substack{i=1 \\ P_i(t)=C_k \\ \alpha_i\tau_{pr}\leq\delta t}}^{N_j} \phi^j\left(q_i(t) - v_k\,\alpha_i\tau_{pr}, \tau_{kj} - (\delta t - \alpha_i\tau_{pr})\right)\, \mathbf{1}_{\theta_i(t)\leq 0} \\
& \qquad\qquad \mathbf{1}_{q_i(t)>v_k\delta t} \mathbf{1}_{\mathrm{Dec}(t+\alpha_i\tau_{pr}, q_i(t)-v_k\alpha_i\tau_{pr}, k, j)=1} \\
= & \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j}\,\frac{1}{T_k} \sum_{\substack{i=1 \\ \alpha_i\tau_{pr}\leq\delta t}}^{N_k} \Big[\phi^j\left(q_i(t), \tau_{kj}\right) - v_k\,\alpha_i\tau_{pr}\,\partial_q\,\phi^j\left(q_i(t), \tau_{kj}\right) \\
& \qquad -(\delta t - \alpha_i\tau_{pr})\,\partial_\theta\,\phi^j\left(q_i(t), \tau_{kj}\right)\Big]\, \mathbf{1}_{\theta_i(t)\leq 0} \\
& \qquad \mathbf{1}_{q_i(t)>v_k\delta t}\, \mathbf{1}_{\mathrm{Dec}(t+\alpha_i\tau_{pr}, q_i(t)-v_k\alpha_i\tau_{pr}, k, j)=1} + O(\delta t^2).
\end{aligned}
$$

The term B_2 describes the jobs leaving the cluster C_j:

$$
\begin{aligned}
B_2 = - \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{1}{T_j} \sum_{\alpha_i\tau_{pr}\leq\delta t}^{N_j} & \phi^j\left(q_i(t), \theta_i(t)\right)\, \mathbf{1}_{\theta_i(t)\leq 0}\, \mathbf{1}_{q_i(t)>v_j\delta t} \\
& \mathbf{1}_{\mathrm{Dec}(t+\alpha_i\tau_{pr}, q_i(t)-v_j\alpha_i\tau_{pr}, j, k)=1}\,.
\end{aligned}
\tag{5}
$$

To take the limit as $\delta t \to 0$ we first need to scale the characteristic times of the problem with δt so that the behavior of the jobs is correctly taken into account. Since we are looking at the transfer of jobs between clusters, many jobs have to be tested during the time interval δt to obtain a pertinent model. This implies that the time to test a job τ_{pr} is much smaller than δt, $\tau_{pr} \ll \delta t$. However, if all the jobs are tested during the time interval δt, it corresponds to the case of a middle-ware taking instantaneous decisions. In our model the time to test all the jobs $\mathcal{T} = N\,\tau_{pr}$ should remain constant in the limit $\delta t \to 0$. We then assume in the following that

$$
\tau_{pr} = \delta t^2 \ll \delta t \ll \mathcal{T} = O(1). \tag{6}
$$

Remark 2. *This scaling implies that the number of jobs on the network fulfills $N = O\left(1/\delta t^2\right)$. Note however that the number C of clusters is constant. The number of jobs tested on the network during δt is defined by*

$$
N_{\delta t} = \frac{\delta t}{\tau_{pr}} = O\left(\frac{1}{\delta t}\right) \gg 1\,.
$$

Moreover, under the assumption that the distribution of the tested jobs is uniform on $[1, N]$ (it depends neither on the cluster number nor on its state), the number of jobs tested on the cluster C_j during the time interval δt is

$$
\frac{N_j}{N}\,N_{\delta t} = N_j \frac{\delta t}{\mathcal{T}} = O\left(\frac{1}{\delta t}\right) \gg 1.
$$

Finally, aside from this choice of asymptotic, we impose that the filling rate of each cluster remains bounded (and consequently not null):

$$
\int_{q,\theta} f^j\, dq\, d\theta = \frac{N_j}{T_j} = O(1)\,.
$$

Lemma 1. Formal limit as $\delta t \to 0$
The kinetic equation governing the evolution of the distribution function in each cluster C_j, $j = \{1, .., C\}$, for all time $t > 0$ is given in $(\mathcal{D}'(\mathbb{R}_+^*))_q \otimes (\mathcal{D}'(\mathbb{R}))_\theta$ by:

$$\begin{aligned}\partial_t f^j(t) - \mathbf{1}_{\theta\leq 0} v_j \partial_q f^j(t) - \mathbf{1}_{\theta>0}\partial_\theta f^j(t) = -\frac{1}{\mathcal{T}} \sum_{\substack{k=1\\k\neq j}}^{C} f^j(t)\, \mathbf{1}_{\theta\leq 0}\, Dec(j,k) \\ + \frac{1}{\mathcal{T}} \sum_{\substack{k=1\\k\neq j}}^{C} \frac{T_k}{T_j} \left\langle f^k(t)\, Dec(k,j)\,,\, \mathbf{1}_{\theta\leq 0}\right\rangle_\theta \delta_{\theta=\tau_{kj}}.\end{aligned} \tag{7}$$

Proof. We assume that $f_N^j \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow} f^j$ weak-$*$. Taking the limit of $\frac{A}{\delta t}$ we obtain, using a first order development :

$$\frac{1}{\delta t} A \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow} -v_j \left\langle f^j(t)\,,\, \partial_q \phi^j\, \mathbf{1}_{q\geq 0}\, \mathbf{1}_{\theta\leq 0}\right\rangle - \left\langle f^j, \partial_\theta \phi^j\, \mathbf{1}_{q\geq 0}\, \mathbf{1}_{\theta>0}\right\rangle.$$

Since $\alpha_i\, \tau_{pr} \leq \delta t$ then $\alpha_i \tau_{pr} \underset{\delta t\to 0}{\longrightarrow} 0$ and the term B_1 reduces to

$$\begin{aligned}B_1 = \sum_{\substack{k=1\\k\neq j}}^{C} \frac{T_k}{T_j} \frac{1}{T_k} \sum_{\substack{i=1\\ \alpha_i\tau_{pr}\leq\delta t}}^{N_k} \phi^j\left(q_i(t), \tau_{kj}\right)\, \mathbf{1}_{\theta_i(t)\leq 0}\, \mathbf{1}_{q_i(t)>v_k\delta t} \\ \mathbf{1}_{Dec(t+\alpha_i\tau_{pr}, q_i(t)-v_k\alpha_i\tau_{pr}, k, j)=1} + \mathcal{O}(\delta t^2).\end{aligned}$$

The main difficulty in the formal limit of B lies in determining on a given cluster the number of jobs moved since the Dec function depends on the characteristics of each job on the cluster. We then get around this problem thanks to the following assumption:

Assumption 1. *Let us assume that $Dec^{-1}\{1\}$ is an open set.*

This assumption is valid as soon as the cost function is continuous and represents the gain obtained through the minimization process. Then $q \mapsto \mathrm{Dec}(t, q, k, j)$ is piece-wisely constant. Let $I_{tjk} = \mathrm{Dec}^{-1}(t, \cdot, j, k)(\{1\})$. It also implies that Dec only depends on the jobs characteristics and not the jobs themselves. Since the number of tested jobs during δt is equal to $\dfrac{\delta t}{\mathcal{T}}$ we obtain the following weak-$\star$ limit:

$$\frac{1}{\delta t}\frac{1}{T_j} \sum_{\substack{i=1\\ \alpha_i\tau_{pr}\leq\delta t\\ P_i(t)=C_j}}^{N_j} \delta_{q=q_i(t)}\, \delta_{\theta=\theta_i(t)} \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow} \frac{1}{\mathcal{T}} f^j. \tag{8}$$

If $\mathrm{supp}\, \phi^j \subset I_{tjk} \otimes \mathbb{R}$, then:

$$\frac{1}{\delta t} B_2 \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow} -\sum_{\substack{k=1\\k\neq j}}^{C} \frac{1}{\mathcal{T}} \left\langle f^j(t), \phi^j\, \mathbf{1}_{\theta\leq 0}\, \mathbf{1}_{q>0}\right\rangle \mathbf{1}_{Dec(t,q,j,k)=1}. \tag{9}$$

Since the variable q and θ do not vary simultaneously we let

$$\phi^j(q,\theta) = \left(\phi_q^j \otimes \phi_\theta^j\right)(q,\theta) \in \left(\mathcal{D}(\mathbb{R}_+^*)\right)_q \otimes \left(\mathcal{D}(\mathbb{R})\right)_\theta \ , \ \forall j = \{1,..,C\}\,. \tag{10}$$

and denote by $\langle\,.\,,\,.\,\rangle_\theta$ the dual product on $\mathbb{R}_\theta$. Then if supp $\phi_q \subset I_{tjk}$,

$$\begin{aligned}
\frac{1}{\delta t}B_1 = \frac{1}{\delta t}\sum_{\substack{k=1\\k\neq j}}^{C}\frac{T_k}{T_j}\Bigg(\frac{1}{T_k}\sum_{\substack{i=1\\ \alpha_i\tau_{pr}\leq\delta t}}^{N_k}\phi_q^j\left(q_i(t)\right)\ \mathbf{1}_{\theta_i(t)\leq 0}\ \mathbf{1}_{q_i(t)>v_k\delta t}\\
\mathbf{1}_{Dec(t+\alpha_i\tau_{pr},q_i(t)-v_k\alpha_i\tau_{pr},k,j)=1}\Bigg)\phi_\theta^j\left(\tau_{kj}\right) + \mathcal{O}(\delta t)\\
\underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow}\sum_{\substack{k=1\\k\neq j}}^{C}\frac{T_k}{T_j}\frac{1}{\mathcal{T}}\langle f^k(t)\,\mathbf{1}_{q>0}\,\mathbf{1}_{\mathrm{Dec}(t,q,k,j)=1}\,,\,\phi_q^j\otimes\mathbf{1}_{\theta\leq 0}\rangle\ \phi_\theta^j\left(\tau_{kj}\right)\\
\underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow}\sum_{\substack{k=1\\k\neq j}}^{C}\frac{T_k}{T_j}\ \frac{1}{\mathcal{T}}\langle\langle f^k(t)\,\mathbf{1}_{q>0}\,\mathbf{1}_{\mathrm{Dec}(t,q,k,j)=1}\,,\,\mathbf{1}_{\theta\leq 0}\rangle_\theta,\phi_q^j\otimes\phi_\theta^j(\tau_{kj})\rangle\,.
\end{aligned}$$

We now let $Dec(j,k) = \mathbf{1}_{Dec(t,q,j,k)=1}$, and we get:

$$\frac{1}{\delta t}\,B_1 \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow}\frac{1}{\mathcal{T}}\sum_{\substack{k=1\\k\neq j}}^{C}\frac{T_k}{T_j}\,\langle\,\langle f^k(t)\,Dec(k,j),\mathbf{1}_{\theta\leq 0}\,\mathbf{1}_{q>0}\rangle_\theta\ \delta_{\theta=\tau_{kj}},\phi^j\rangle\,, \tag{11}$$

and

$$\frac{1}{\delta t}\,B_2 \underset{\substack{N\to+\infty\\ \delta t\to 0}}{\longrightarrow} -\frac{1}{\mathcal{T}}\sum_{\substack{k=1\\k\neq j}}^{C}\langle f^j(t),\phi^j\,\mathbf{1}_{\theta\leq 0}\,\mathbf{1}_{q>0}\rangle\ Dec(j,k)\,, \tag{12}$$

which gives the kinetic equation (7). □

Since the evolution on q and θ are not simultaneous we split the distribution function f^j as

$$f^j(t,q,\theta) = \rho_j^{exe}(t,q)\ \delta_{\theta=0} + \rho_j^{wait}(t,q,\theta)\ \mathbf{1}_{\theta>0} + \rho_j^{garb}(t,q,\theta)\ \mathbf{1}_{\theta<0}\,, \tag{13}$$

where the functions ρ_j^{exe} (for the working jobs), ρ_j^{wait} (for the waiting jobs) and ρ_j^{garb} are defined by:

$$\begin{gathered}
\rho_j^{exe}(t,q) = f^j(t,q,\theta)\ \mathbf{1}_{\theta=0},\quad \rho_j^{wait}(t,q,\theta) = f^j(t,q,\theta)\ \mathbf{1}_{\theta>0},\\
\rho_j^{garb}(t,q,\theta) = f^j(t,q,\theta)\ \mathbf{1}_{\theta<0}.
\end{gathered} \tag{14}$$

Remark 3. *As the evolution in θ stops as soon as $\theta \leq 0$, ρ^{garb} is taken into account for the sake of generality and more importantly so that the definition domain in θ is an open set. This formulation is coherent with the solutions obtained for transport equations with discontinuous coefficients [19].*

Lemma 2. ***Characterization of the solutions of the kinetic equation***
We assume that $f^j(t=0,q,\theta)$ fulfills the decomposition (13) *with $\rho_j^{exe}(t=0,q) \in \mathbb{W}^{1,\infty}(\mathbb{R}_+^*)$, $\rho_j^{wait}(t=0,q,\theta) \in \mathbb{W}^{1,\infty}(\mathbb{R}_+^*\times\mathbb{R}_+^*)$ and $\rho_j^{garb}(t=0,q,\theta) \in \mathbb{W}^{1,\infty}(\mathbb{R}_+^*\times\mathbb{R}_-^*)$. Then the solutions in $\left(\mathcal{D}'(\mathbb{R}_+^*)\right)_q \otimes \left(\mathcal{D}'(\mathbb{R})\right)_\theta$ of* (7) *for $t > 0$ fulfill the decomposition* (13)*, with $\rho_j^{wait}(t,q,\theta) \in \mathbb{L}^\infty(\mathbb{R}^+,\mathbb{W}^{1,\infty}(\mathbb{R}_+^*\times\mathbb{R}_+^*))$ and $\rho_j^{garb}(t,q,\theta) \in \mathbb{L}^\infty(\mathbb{R}^+,\mathbb{W}^{1,\infty}(\mathbb{R}_+^*\times\mathbb{R}_-^*))$.*

Proof. We show that no singularity in θ arises along the time evolution of ρ_j^{wait} and ρ_j^{garb}. We test the kinetic equation (7) with a test function ϕ^j such that supp $\phi^j \subset (\mathbb{R}_+^*)_q \otimes (\mathbb{R}_-^*)_\theta$ and get

$$\langle \partial_t f^j(t)\,,\, \phi^j \rangle = \langle \partial_t \rho^{garb}(t)\,,\, \phi^j \rangle \tag{15}$$

with

$$\begin{aligned} \rho_j^{garb}(t,q,\theta) = \quad & \rho_j^{garb}(0, q + v_j t, \theta) - \int_0^t \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \rho_j^{garb}(s, q + v_j(t-s), \theta) \\ & \mathrm{Dec}(s, q + v_j(t-s), j, k)\, ds\,. \end{aligned} \tag{16}$$

By testing (7) with a test function ϕ^j such that supp $\phi^j \subset (\mathbb{R}_+^*)_q \otimes (\mathbb{R}_+^*)_\theta$, we get

$$\langle \partial_t f^j(t)\,,\, \phi^j \rangle = \langle \partial_t \rho_j^{wait}(t)\,,\, \phi^j \rangle \tag{17}$$

with

$$\begin{aligned} \rho_j^{wait}(t,q,\theta) = \quad & \rho_j^{wait}(0, q, \theta + t) + \int_0^t \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j} \\ & \int_{\theta \leq 0} f^k(s, q, \theta + t - s) \mathrm{Dec}(s, q, k, j) d\theta\, \delta_{\theta + t - s = \tau_{kj}}\, ds\,. \end{aligned} \tag{18}$$

Since the initial data are regular enough, ρ_j^{exe}, ρ_j^{wait} and ρ_j^{garb} are regular for $t > 0$. □

Let us note that if $\rho_j^{garb}(0, \cdot, \cdot) = 0$ in (16) (which is the case in our model), then $\rho_j^{garb}(t, \cdot, \cdot) = 0$ for all $t > 0$ and if the solution exists, it writes

$$f^j(t,q,\theta) = \rho_j^{exe}(t,q)\, \delta_{\theta=0} + \rho_j^{wait}(t,q,\theta)\, \mathbf{1}_{\theta>0}\,. \tag{19}$$

with the functions ρ_j^{exe} and ρ_j^{wait} defined by

$$\rho_j^{exe}(t,q) = \langle f^j(t,q,\theta)\,,\, \mathbf{1}_{\theta\leq 0} \rangle \;\text{ and }\; \rho_j^{wait}(t,q,\theta) = f^j(t,q,\theta)\, \mathbf{1}_{\theta>0}\,.$$

3.2. Asymptotics for conservation laws

We study here the fluid system fulfilled by ρ_j^{exe} and ρ_j^{wait} for all $j \in \{1, .., C\}$. We use the tensor test functions $\phi^j \in (\mathcal{D}'(\mathbb{R}_+^*))_q \otimes (\mathcal{D}'(\mathbb{R}))_\theta$ such that

$$\phi^j(q,\theta) = \phi^{exe}(q,\theta)\, \mathbf{1}_{\theta\leq 0} + \phi^{wait}(q,\theta)\, \mathbf{1}_{\theta>0} \tag{20}$$

Testing (7)) with ϕ^j,we get in the distribution sense $\forall\, j \in \{1, .., C\}$

$$\begin{cases} \partial_t \rho_j^{exe}(t,q) - v_j \partial_q \rho_j^{exe}(t,q) = S_j^{exe}(t,q), \\ \partial_t \rho_j^{wait}(t,q,\theta) - \partial_\theta \rho_j^{wait}(t,q,\theta) = S_j^{wait}(t,q,\theta), \end{cases} \tag{21}$$

with

$$S_j^{exe}(t,q) = \rho_j^{wait}(\theta = 0^+) - \frac{1}{\mathcal{T}} \sum_{k\neq j} \rho_j^{exe}(t,q)\; Dec(j,k), \tag{22}$$

and

$$S_j^{wait}(t,q,\theta) = \frac{1}{\mathcal{T}} \sum_{k\neq j} \frac{T_k}{T_j}\, \rho_k^{exe}(t,q)\; Dec(k,j)\; \delta_{\theta=\tau_{kj}}. \tag{23}$$

3.3. Model extension

Let us note that we could use more parameters to characterize the jobs but they were not used in this model. These parameters could be $s_i \in \mathbb{R}^+$ the size of a job, $p_i \in \mathbb{N}^*$ the number of processors required for the job J_i, $a_i \in \mathbb{R}_+^*$ the age of the job.

These parameters evolve following

$$a_i'(t) = 1\,,\ s_i'(t) = 0\,,\ p_i'(t) = 0\,. \tag{24}$$

By denoting $x = (a, p, s)$ we then get the following kinetic equation in the distribution sense

$$\begin{cases} (\partial_t \rho_j^{exe} - v_j \partial_q \rho_j^{exe} + \partial_a \rho_j^{exe})(t, q, x) = S_j^{exe}(t, q, x), \\ (\partial_t \rho_j^{wait} - \partial_\theta \rho_j^{wait} + \partial_a \rho_j^{wait})(t, q, \theta, x) = S_j^{wait}(t, q, \theta, x). \end{cases} \tag{25}$$

where S_j^{exe} and S_j^{wait} are defined as in (22)-(23).

To get the fluid equations on the momentum of the distribution functions ρ_j^{exe} and ρ_j^{wait}, we let:

$$n_j^{exe/wait} = \int_x \rho_j^{exe/wait}\, dx, \tag{26}$$

$$\bar{\alpha}_j n_j^{exe/wait} = \int_x \alpha\, \rho_j^{exe/wait},\ \text{for } \alpha = \{a, p, s\}, \tag{27}$$

with $n_j^{exe/wait}$ the job density, $(n_j \bar{a}_j)^{exe/wait}$ the mean age of the jobs, $(n_j \bar{s}_j)^{exe/wait}$ the mean size of the jobs, $(n_j \bar{p}_j)^{exe/wait}$ the mean number of needed processors for the jobs, whether they are executed or waiting.

4. Mathematical study of the model (21)-(22)-(23)

4.1. Links between mesoscopic and macroscopic models

To prove the existence of the fluid system we use a theorem from Jabin [20] that we adapt to our case.

We let the initial data:

$$f(0, \cdot, \cdot) = f_0\,, \quad \rho_j^{exe}(0, \cdot) = \rho_{j,0}^{exe} \ \text{ and } \rho_j^{wait}(0, \cdot, \cdot) = \rho_{j,0}^{wait}\,. \tag{28}$$

Lemma 3. *ρ_j^{exe} and ρ_j^{wait}, with the initial data given in* (28)*, are solutions of the fluid system* (21) *in $\mathcal{D}'(\mathbb{R}_+^*)_q$ and $(\mathcal{D}'(\mathbb{R}_+^*))_q \otimes (\mathcal{D}'(\mathbb{R}))_\theta$ respectively if and only if f^j written in the form* (19) *is solution of the kinetic equation* (7) *with an initial data fulfilling the closure assumption* (19)*.*

Proof. Let us first assume that ρ_j^{exe} and ρ_j^{wait}, with their initial data, are solutions if the fluid system (21) and that

$$f^j(0, q, \theta) = \rho_j^{exe}(0, q)\ \delta_{\theta=0} + \rho^{wait}(0, q, \theta)\ \mathbf{1}_{\theta>0}\,. \tag{29}$$

Since we have $\rho_j^{garb}(0, \cdot, \cdot) = 0$ and (29), we apply Lemma 2 to characterize the solutions of the kinetic equation and get for all $t > 0$ that the distribution function writes

$$f^j(t, q, \theta) = \rho_j^{exe}(t, q)\ \delta_{\theta=0} + \rho_j^{wait}(t, q, \theta)\ \mathbf{1}_{\theta>0}\,. \tag{30}$$

By writing the kinetic equation in the distribution sense with the test functions defined in (20), replacing f^j by the closure assumption and testing this equation with ϕ^j we get that ρ_j^{exe} and ρ_j^{wait} are solutions of the fluid system (21). □

4.2. Local theory

The fluid system (21) is nonlinear due to the decision function Dec. We introduce the natural distribution spaces such that $\rho_j^{exe} \in B^{exe}$ and $\rho_j^{wait} \in B^{wait}$, $\forall j \in \{1,..,C\}$, where the spaces are defined by

$$B^{exe} = \mathbb{L}^\infty\left([0,T], \mathbb{W}^{1,\infty}(\mathbb{R}_+^*)_q\right) \cap F_q \tag{31}$$

and

$$B^{wait} = \mathbb{L}^\infty\left([0,T], \mathbb{W}^{1,\infty}\left(\mathbb{R}_+^*\right)_q \times \mathbb{W}^{1,\infty}(\mathbb{R})_\theta\right) \cap F_\theta \tag{32}$$

where F_X is the set of function which admit a right-handed limit in the X-variable.

We introduce the vectors ρ^{exe} and ρ^{wait} such that $\rho^{exe} = (\rho_j^{exe})_{j\in\{1,..,C\}} \in (B^{exe})^C$ and $\rho^{wait} = (\rho_j^{wait})_{j\in\{1,..,C\}} \in \left(B^{wait}\right)^C$. We moreover define the norm of ρ^{exe} such that

$$\|\rho^{exe}\|_{exe} = \sum_{j=1}^{C} T_j \|\rho_j^{exe}\|_{B^{exe}} . \tag{33}$$

Proposition 1. ***Study of a model problem***
Let $a \in \mathbb{R}$ and $\rho \in B = \mathbb{L}^\infty([0,T], \mathbb{W}^{1,\infty}(\mathbb{R}))$. We consider the problem

$$\begin{cases} \partial_t\rho(t,x) - \partial_x\rho(t,x) = S(t)\,\delta_{x=a}\,, & \forall\,(t,x) \in [0,T]\times\mathbb{R}, \\ \rho(0,x) = \rho_0(x)\,, & \forall\, x \in \mathbb{R}. \end{cases} \tag{34}$$

There exists then a solution $\rho(t,x) \in B \cap F_x$ which is given by

$$\rho(t,x) = \begin{cases} \rho_0(x+t) & , \text{ if } x < a-t \text{ or if } x \geq a\,, \\ \rho_0(x+t) + S(x+t-a) & , \text{ if } a-t \leq x < a\,, \end{cases} \tag{35}$$

if $S(x+t-a)$ is right limited.

Proof. We are dealing with a transport equation with negative speed. We find that $\rho(t,x) = \rho_0(\xi) + \displaystyle\int_0^t S(s)\,\delta_{s=(x+t)-a}\,ds$, which gives the result as soon as $S(x+t-a)$ admits a limit on the right-hand side. □

4.3. Global existence and uniqueness theorem

Theorem 1. ***Existence and uniqueness of fluid solutions***
Let $\rho_{j,0}^{exe} \in B^{exe}$ and $\rho_{j,0}^{wait} \in B^{wait}$ for all $j \in \{1,..,C\}$. There exists then an unique solution ρ^{exe} and ρ^{wait} to (21) for all $t \in \mathbb{R}^+$ with the initial data $\rho_j^{exe}(0,q) = \rho_{j,0}^{exe}(q)$ and $\rho^{wait}(0,q,\theta) = \rho_{j,0}^{wait}(q,\theta)$..

Proof. We use a fixed point theorem. Let $\Psi : (B^{exe})^C \to (B^{exe})^C$ the application that maps ρ^{exe} to $\rho^{exe(1)}$ where $(\rho^{exe(1)}, \rho^{wait})$ is solution of

$$\begin{cases} \partial_t \rho_j^{exe(1)}(t,q) - v_j\,\partial_q \rho_j^{exe(1)}(t,q) = S_j^{exe(1)}(t,q), \\ \partial_t \rho_j^{wait}(t,q,\theta) - \partial_\theta \rho_j^{wait}(t,q,\theta) = S_j^{wait}(t,q,\theta), \end{cases} \tag{36}$$

with

$$\begin{aligned} S_j^{exe(1)}(t,q) &= \rho_j^{wait}(t,q,\theta=0^+) - \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \rho_j^{exe(1)}(t,q)\; Dec(j,k)\,, \\ S_j^{wait}(t,q,\theta) &= \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j}\,\rho_k^{exe}(t,q)\; Dec(k,j)\;\delta_{\theta=\tau_{kj}}\,, \end{aligned} \tag{37}$$

and the initial condition

$$\rho^{exe(1)}(0,q) = \rho_0^{exe}(q) = \left(\rho_{j,0}^{exe}(q)\right)_{j\in\{1,..,C\}}, \forall q \in \mathbb{R}_+^*,$$
$$\rho^{wait}(0,q,\theta) = \rho_0^{wait}(q,\theta) = \left(\rho_{j,0}^{wait}(q,\theta)\right)_{j\in\{1,..,C\}}, \forall (q,\theta) \in \mathbb{R}_+^* \times \mathbb{R}_+^*.$$

We define the sequence $\rho^{exe(l+1)} = \Psi(\rho^{exe(l)})$ starting from $\rho^{exe(0)} = \rho_0^{exe}$. In the first part of the proof we compute explicitly $\Psi(\rho^{exe})$.

Thanks to Prop. 1 we obtain that

$$\rho_j^{wait(l)}(t,q,\theta) = \begin{cases} \rho_{j,0}^{wait}(q,\theta+t) \text{ if } \theta < \tau_{kj} - t \text{ or } \theta \geq \tau_{kj}\,, \\ \rho_{j,0}^{wait}(q,\theta+t) + \dfrac{1}{\mathcal{T}} \displaystyle\sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j} \rho_k^{exe}(\theta+t-\tau_{kj},q)\mathbf{1}_{Dec(\theta+t-\tau_{kj},q,k,j)=1}\,, \\ \qquad \text{if } \tau_{kj} - t \leq \theta < \tau_{kj}\,, \end{cases}$$

if the second member of the second equation of (36) admits a limit on the right hand side in $\theta + t - \tau_{kj}$, *i.e.* if ρ_k^{exe} has a limit on the right hand side in time. This is satisfied since $\rho^{exe} \in B^{exe}$. Then we have

$$\rho_j^{wait}(t,q,0^+) = \begin{cases} \rho_{j,0}^{wait}(q,t)\,, \text{ if } t < \tau_{kj}\,, \\ \rho_{j,0}^{wait}(q,t) + \dfrac{1}{\mathcal{T}} \displaystyle\sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j} \rho_k^{exe}(t-\tau_{kj},q)\mathbf{1}_{Dec(t-\tau_{kj},q,k,j)=1}\,, \quad \text{if } t \geq \tau_{kj}\,. \end{cases}$$

Back to the first equation in (36) we get

$$\begin{aligned} \rho_j^{exe(1)}(t,q) = \rho_{j,0}^{exe}(q) & \\ + \int_0^t \frac{1}{\mathcal{T}} \sum_{k\neq j} \frac{T_k}{T_j} \rho_k^{exe}(s-\tau_{kj}, q+v_j(t-s))\, \mathbf{1}_{Dec(s-\tau_{kj},q+v_j(t-s),k,j)=1}\, ds & \\ + \int_0^t \rho_{j,0}^{wait}(q+v_j(t-s),s)\, ds & \\ - \int_0^t \frac{1}{\mathcal{T}} \sum_{k\neq j} \rho_j^{exe(1)}(s, q+v_j(t-s))\, \mathbf{1}_{Dec(s,q+v_j(t-s),j,k)=1}\, ds\,. & \end{aligned} \tag{38}$$

We now look for conditions so that Ψ is a contraction. Let $\mu_j^{exe} \in B^{exe}$, and $\mu_j^{exe(1)} \in B^{exe}$, $\mu_j^{wait} \in B^{wait}$ solutions of the fluid system (36) so that $\mu^{exe(1)} = \Psi(\mu^{exe})$. Then

$$\begin{aligned} |\rho_j^{exe+1)}(t,q) - \mu_j^{exe(1)}(t,q)| &= \chi(t,q) - \int_0^t \psi(s,q)\, \left(\rho_j^{exe(1)}(s,q) - \mu_j^{exe(1)}(s,q)\right) ds \\ &\leq |\chi(t,q)| + \int_0^t |\psi(s,q)|\, |\rho_j^{exe(l)}(s,q) - \mu_j^{exe(l)}(s,q)|\, ds \end{aligned}$$

with

$$\chi(t,q) = \int_0^t \frac{1}{\mathcal{T}} \sum_{k\neq j} \frac{T_k}{T_j} \left(\rho_k^{exe} - \mu_k^{exe}\right)(s-\tau_{kj}, q+v_j(t-s))\, \mathbf{1}_{Dec(s-\tau_{kj},q+v_j(t-s),k,j)=1}\, ds\,,$$

and

$$\psi(s,q) = \frac{1}{\mathcal{T}} \sum_{k \neq j} \mathbf{1}_{Dec(s,q+v_j(t-s),j,k)=1} \,.$$

Thanks to Gronwall inequality we get

$$|\rho_j^{exe(1)}(t,q) - \mu_j^{exe(1)}(t,q)| \leq |\chi(t,q)| + \int_0^t |\chi(s,q)|\ |\psi(s,q)|\ \exp\left(\int_s^t |\psi(u)|\ du\right)\ ds\,.$$

and then

$$\|\rho_j^{exe(1)} - \mu_j^{exe(1)}\|_{\mathbb{L}^\infty([0,T])_t,\mathbb{L}^\infty(\mathbb{R}_+^*)_q)} \leq \frac{1}{C-1} \sum_{k \neq j} \frac{T_k}{T_j} \|\rho_k^{exe} - \mu_k^{exe}\|_{\mathbb{L}^\infty([0,T])_t,\mathbb{L}^\infty(\mathbb{R}_+^*)_q)} \left(-1 + \exp\left(\frac{T\,(C-1)}{\mathcal{T}}\right)\right).$$

By taking the derivative in q of (38) we get in the same way

$$T_j\ \|\rho_j^{exe(1)} - \mu_j^{exe(1)}\|_{B^{exe}} \leq \frac{1}{C-1} \sum_{k \neq j} T_k\ \|\rho_k^{exe} - \mu_k^{exe}\|_{B^{exe}} \left(-1 + \exp\left(\frac{T\,(C-1)}{\mathcal{T}}\right)\right)$$

and finally

$$\|\Psi(\rho^{exe}) - \Psi(\mu^{exe})\|_{exe} \leq \frac{C}{C-1} \|\rho^{exe} - \mu^{exe}\|_{exe} \left(-1 + \exp\left(\frac{T\,(C-1)}{\mathcal{T}}\right)\right).$$

If

$$T \leq \frac{\mathcal{T}}{C-1}\ \ln\left(\frac{3C-1}{2C}\right) = T_{max}\,,$$

the application Ψ is $1/2$-Lipschitz. Since the time T_{max} only depends on the model we also have existence of solutions on the time interval $[T, 2T]$,... as long as $T \leq T_{max}$ and then the existence of an unique solution on $\mathbb{R}^+$. □

Theorem 2. ***Existence and uniqueness of the kinetic solution under closure assumption*** *Let ρ^{exe} and ρ^{wait} such that $\rho_j^{exe} \in B^{exe}$ and $\rho_j^{wait} \in B^{wait}$ satisfy* (21) *$\forall\, j \in \{1,..,C\}$,with the initial data $\rho_j^{exe}(0,q) = \rho_{j,0}(q)$ and $\rho_j^{wait}(0,q,\theta) = 0$. Let us assume that $f^j(t=0,q,\theta)$ splits according to the closure assumption* (13)*. There exists then an unique solution to the kinetic equation that splits according to the closure assumption for all $t \in \mathbb{R}_+^*$.*

Proof. The assumptions on the distribution function allow us to apply Lemma 2 to characterize the kinetic equations. For all time the solution f then splits according to the closure assumption. Thanks to Theorem 1, we moreover have uniqueness of the solutions ρ_j^{exe} and ρ_j^{wait}. Since the distribution function admits a unique decomposition according to the closure assumption, defined through its moments, we obtain the uniqueness of the kinetic equation and of its decomposition. □

5. Numerical results

5.1. Numerical method

We now numerically solve the system through the fluid model. We have on each cluster C_j:

$$\begin{cases} \partial_t \rho_j^{exe}(t,q) - v_j\, \partial_q \rho_j^{exe}(t,q) = S_j^{exe}(t,q)\,, \\ \partial_t \rho_j^{wait}(t,q,\theta) - \partial_\theta \rho_j^{wait}(t,q,\theta) = S_j^{wait}(t,q,\theta)\,, \end{cases} \tag{39}$$

with the source terms S_j^{exe} and S_j^{wait} defined by (22)-(23)

In order to simulate this system of conservation equations, a Finite Volume approach has been used, along with a first order Explicit Euler method for time integration. We discretized the respective phase spaces for these two equations. We then impose $q \in [0, q_{max}]$ and $\theta \in [0, \theta_{max}]$, where q_{max} is obtained by taking the worst case scenario (maximal execution time and flow rate on the grid) and $\theta_{max} = \max_{j,k\in\{1,..,C\}} \tau_{jk}$.

The numerical domain for the workload is divided into N_q intervals of uniform length $\Delta q = q_{max}/N_q$. The $N_q + 1$ points of discretization are noted $q_{i-1/2} = (i-1)\Delta q$ where $i \in \{1, .., N_q + 1\}$. By definition, $0 = q_{1/2} < q_{3/2} < ... < q_{Nq-1/2} < q_{N_q+1/2} = q_{max}$. Q_i denotes the interval $[q_{i-1/2}, q_{i+1/2}[$, so that $\bigcup_{i=1}^{Nq} Q_i = [0, q_{max}]$.

We use the same approach for the θ variable, introducing N_θ cells of uniform length, noted $\Theta_m = [\theta_{m-1/2}, \theta_{m+1/2}]$ with $\theta_{m-1/2} = (m-1)\Delta\theta$, for $m \in \{1, .., N_\theta + 1\}$.

T will denote the time horizon and the interval $[0, T]$ is split within N_t subintervals of length $\Delta t = T/N_t$. We will note $t^n = n \times \Delta t$, for $n \in \{0, .., N_t\}$.

Finally, with these notations, we define :

$$\rho_{j,i}^{exe}(t) \simeq \frac{1}{\Delta q} \int_{q_{i-1/2}}^{q_{i+1/2}} \rho_j^{exe}(t,q)\, dq$$

and

$$\rho_{j,i,m}^{wait}(t) \simeq \frac{1}{\Delta q\, \Delta\theta} \int_{q_{i-1/2}}^{q_{i+1/2}} \int_{\theta_{m-1/2}}^{\theta_{m-1/2}} \rho_j^{wait}(t,q,\theta)\, dq\, d\theta.$$

We are then able to define the fluxes of jobs with the "running" status on a cluster C_j. The equation (39) being a 1D-transport equation, a first-order upwind scheme is used. In this particular case, the velocity v_j is known to be positive so that we have :

$$\mathcal{F}_{j,i+1/2}^{exe}(t) = -v_j\, \rho_{j,i+1}^{exe}(t)$$

Similarly, we use a first-order upwind scheme for the fluxes of "waiting" jobs :

$$\mathcal{F}_{j,i,m+1/2}^{wait}(t) = -\rho_{j,i,m+1}^{wait}(t)\,.$$

which yields the following semi-discretized scheme :

$$\begin{aligned} \frac{d}{dt}\rho_{j,i}^{exe}(t) + \frac{1}{\Delta q}\left[\mathcal{F}_{j,i+1/2}^{exe}(t) - \mathcal{F}_{j,i-1/2}^{exe}(t)\right] &= S_{j,i}^{exe}(t)\,, \\ \frac{d}{dt}\rho_{j,i,m}^{wait}(t) + \frac{1}{\Delta\theta}\left[\mathcal{F}_{j,i,m+1/2}^{wait}(t) - \mathcal{F}_{j,i,m-1/2}^{wait}(t)\right] &= S_{j,i,m}^{wait}(t)\,, \end{aligned}$$

where $S^{exe}_{j,i}(t)$ and $S^{wait}_{j,i,m}(t)$ are the source term values, assuming that Dec is constant on any $]q_{i-1/2}, q_{i+1/2}[$, given by :

$$S^{exe}_{j,i}(t) \simeq -\mathcal{F}^{wait}_{j,i,1/2}(t) - \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \rho^{exe}_{j,i}(t)\ \mathbf{1}_{Dec(t,q_i,j,k)=1}\,,$$

and

$$S^{wait}_{j,i,m}(t) \simeq \frac{1}{\mathcal{T}\ \Delta\theta} \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j}\, \rho^{exe}_{k,i}(t)\ \mathbf{1}_{Dec(t,q_i,k,j)=1}\ \delta_{\theta_m=\tau_{kj}}\,.$$

To take into account the transport term and the source term we use a first order Lie splitting. We first solve on a time step the sourceless transport equation:

$$\begin{cases} \dfrac{d}{dt}\rho^{exe}_{j,i}(t) + \dfrac{1}{\Delta q}\left[\mathcal{F}^{exe}_{j,i+1/2}(t) - \mathcal{F}^{exe}_{j,i-1/2}(t)\right] = 0\,, \\ \dfrac{d}{dt}\rho^{wait}_{j,i,m}(t) + \dfrac{1}{\Delta\theta}\left[\mathcal{F}^{wait}_{j,i,m+1/2}(t) - \mathcal{F}^{wait}_{j,i,m-1/2}(t)\right] = 0\,, \end{cases} \tag{40}$$

We then inject the solution of the previous system in the ordinary differential equation with only the source term that we solve on a time step

$$\frac{d}{dt}\rho^{exe}_{j,i}(t) = \tilde{S}^{exe}_{j,i}(t)\frac{d}{dt}\rho^{wait}_{j,i,m}(t) = \tilde{S}^{wait}_{j,i,m}(t)\,, \tag{41}$$

with $\tilde{S}^{exe}_{j,i}(t)$ and $\tilde{S}^{wait}_{j,i,m}(t)$ defined by

$$\tilde{S}^{exe}_{j,i}(t) = \tilde{\rho}^{wait}_{j,i,1/2}(t) - \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \tilde{\rho}^{exe}_{j,i}(t)\ \mathbf{1}_{Dec(t,q_i,j,k)=1}\,, \tag{42}$$

and

$$\tilde{S}^{wait}_{j,i,m}(t) = \frac{1}{\mathcal{T}} \sum_{\substack{k=1 \\ k\neq j}}^{C} \frac{T_k}{T_j}\, \tilde{\rho}^{exe}_{k,i}(t)\ \mathbf{1}_{Dec(t,q_i,k,j)=1}\ \delta_{\theta_m=\tau_{kj}}\,. \tag{43}$$

In both cases we use a first order explicit Euler scheme whose CFL condition for the $\mathbb{L}^\infty$-stability is

$$\Delta t \leq \min\left\{\frac{\Delta q}{\max_{j\in\{1,\dots,J\}} v_j}, \Delta\theta, \mathcal{T}\right\}\,. \tag{44}$$

The simulations in the following sections were performed using the numerical parameters described in table 1, on a laptop equipped with an Intel® Core™ i5-2540M CPU @ 2.60GHz and 4 GB of RAM. The mesh sizes have been chosen so that their influence on the final results are quite unnoticeable.

Parameter	q_{max}	θ_{max}	N_q	N_θ	$\mathcal{T}$	T
Value	50	20 s	100	100	1 s	100 s

Table 1: Numerical simulation parameters

In order to accurately capture the dynamics of all clusters, a complete save of the simulation state is performed each 0.2 second of physical time simulated, yielding to a total of 500 save points for each simulation. In these conditions, each simulation required roughly 89.9 s to go to completion, the saves accounting for the vast majority of it (67.4 s). It has to be noted that such a high amount of save is unnecessary if the only purpose is to evaluate a cluster policy efficiency, allowing for faster than real-time simulations on a laptop.

5.2. A 3-cluster dynamics example

It is quite easy to find a 2-cluster configuration with an analytical solution in order to validate the numerical solver [18]. We will here present a 3-cluster configuration, exhibiting a more complex behavior. To this extent, we consider a cost function depending on both the remaining execution time for a job t^{exe} and the electrical consumption K_e of the target *cluster* C_k. These two items read :

$$K(q_i, C_k) = \alpha_t t^{exe}(q_i, C_j, C_k) + \alpha_e K_e(q_i, C_k)\,, \text{ with } \forall\, j,\, k \in \{1, .., C\}:$$

$$K_e(q_i, C_k) = \int_0^{\frac{q_i}{v_k}} Z_k(t)\, dt \text{ and } t^{exe}(q_i(t), C_j, C_k) = \frac{q_i(t)}{v_k} + \tau_{jk}\,,$$

Using a dimensionless parameter $\lambda \in [0, 1]$, the coefficients α_e and α_t are respectively taken as λJ^{-1} and $(1-\lambda)s^{-1}$ so that K is dimensionless and that we can define a cluster policy by changing the value of λ. For example, taking $\lambda = 0$ means that we are only interested in optimizing the time of execution regardless of the energy needed. Conversely, $\lambda = 1$ means that the only objective is to execute the jobs using the less energy possible with no constraint on the time needed to complete them.

We choose to model the part of the electrical consumption $Z_k(t)$ due to the computers as a linear function of the total load $\tau(t)$ on the cluster :

$$Z_k^1(t) = c_k \left(\int_q \rho_k^{exe}(t, q)\, dq + \int_{q,\theta} \rho_k^{wait}(t, q, \theta)\, dq\, d\theta \right) = c_k.\tau(t)\,,$$

where $c_k \in \mathcal{R}_+^*$ is a coefficient accounting for the energetic efficiency of the cluster. This kind of formula is in agreement with the estimates for energy consumption used in real computational clusters [21].

In order to obtain a non-trivial dynamic, we are interested in a configuration for which a cluster seems at first attractive performance-wise and becomes less and less interesting energetic-wise as its workload rises, until the point where some of the jobs are moved towards another cluster. To take this effect of over-consumption into account (due for example to the use of cooling systems), we define a cost function with a significant rise of the slope for an occupation rate$\tau(t)$ above 50%.

$$Z_k(t) = c_k * \tau(t) + c_k * \max(0, \tau(t) - 0.5) \tag{45}$$

The parameters at disposal to describe such a set-up are the performance indexes and energetic consumption of the clusters but also the transfer rates between them. Let us begin with the hierarchy of performances for the clusters. We fix the cluster C_0 as the less interesting one setting reference values $v_0 = 1.0$ and $c_0 = 1.0$. The two other clusters will both be more attractive than C_0, the second cluster C_1 being the best choice energetic-wise and the third C_2 being the best choice for the performance but also the most energy consuming (see table 2).

Cluster	Capacity	Performance	Consumption
0	1700	1.0	1.0
1	3000	2.0	1.0
2	600	10.0	11.0

Table 2: Grid configuration

Cluster	Rate (Nb.s^{-1})	Waiting time (s)	Duration (s)
0	100	2.0	20.0
1	65	4.0	20.0
2	20	10.0	10.0

Table 3: Jobs submission scenario for each cluster

5.2.1. A reference case with no transfer

In order to describe the test-case used in terms of workload scenario, we first perform a reference simulation where all transfers are forbidden. In our model, this is strictly equivalent to choose a cost function only based on performance ($\lambda = 0$) and insanely high transfer times between clusters.

As for the initial condition, all clusters are considered completely empty with no job waiting. At $t = 0$, jobs are submitted to each cluster. Submission rate (total number of jobs submitted per second), duration of submission and associated waiting time varies with the cluster but remains constant over time. These parameters are listed in the table 3. All jobs submitted are assumed to have a constant distribution along the computational load variable (q).

The dynamics of ρ^{exe} and ρ^{wait} for the the cluster 0 are respectively represented on figures 1 and 2. We can easily distinguish four phases. At first, all jobs submitted are waiting and no one is in execution ($t = 1s$). Then in a second phase, the jobs submitted have partly entered execution state and as the job submission rate is big enough, ρ^{exe} is growing over time. In a third step, the job submission ends and all jobs are leaving the waiting state, ρ^{exe} reaching its maximum value. In the last phase, all remaining jobs are being executed. We can observe that the behavior of ρ^{exe} and ρ^{wait} are consistent with the exact solutions of equations 21.

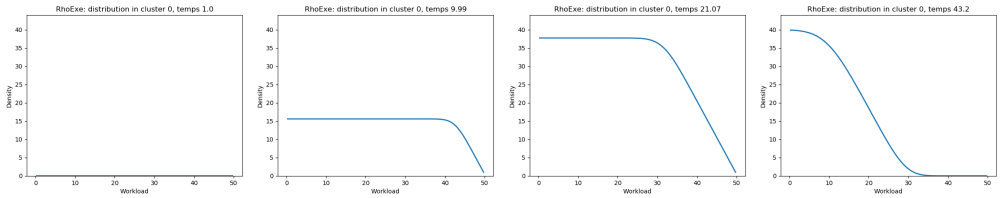

Figure 1: ρ^{exe} distribution for cluster 0 at times $t = 1s$, $9.99s$, $21.07s$ and $43.2s$

As for the occupation rates, as expected, the results first show a rise of the occupation rate (as new jobs are submitted) and then a decrease due to the execution of the jobs on the cluster (see figure 3). All values for this test-case have been fixed in order to keep the occupation rate under 100%.

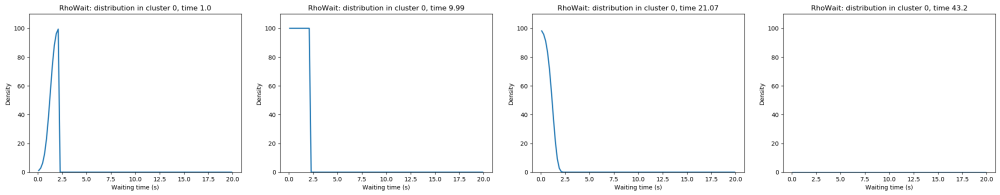

Figure 2: ρ^{wait} distribution for cluster 0 at times $t = 1s$, $9.99s$, $21.07s$ and $43.2s$

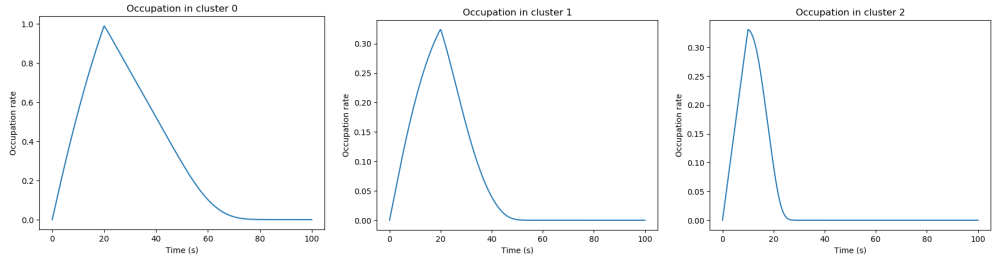

Figure 3: Occupation rate over time for clusters 0 to 2

5.2.2. Application to protocol monitoring and performance assessment

In this section, we propose to illustrate the impact of the cluster policy on the performance of the grid with respect to its energy consumption and time needed to execute all the jobs submitted. To this extent, we have performed some numerical simulations with various values for the parameter λ. These cases will be labeled with the relative importance λ of the energy in the cost function K. For example, a simulation labelled "80% energy" corresponds to $\lambda = 0.8$.

We propose a total of 5 test-cases, with $\lambda \in \{0, 0.01, 0.2, 0.8, 1\}$. The reference case with no transfer will also be displayed. It has to be noted that the reference case differs from the test-case with $\lambda = 0$ since for the reference case, the transfer times had been changed in order to prevent any transfer.

As for the transfer times used for these simulations, these are summarized in table 4. For simplicity sake, these times are considered to be symmetric but this aspect is not mandatory. These values have been chosen so that they have a non-negligible impact on the time needed to execute a job with a high workload but also do not render the transfer cost prohibitive.

As represented on the figure 4, enabling the transfers between clusters does have an impact on both the energy consumption and the time needed to complete all the jobs submitted. On the left side is represented the energy consumption with respect to time for all the test-cases performed. On the middle figure is indicated the number of jobs remaining in the system while

	Cluster 0	**Cluster 1**	**Cluster 2**
Cluster 0	0.0	3.0	8.0
Cluster 1	3.0	0.0	7.0
Cluster 2	8.0	7.0	0.0

Table 4: Transfer times (s) between clusters

on the right side is represented the remaining computational load (since all the jobs do not have the same computational load at submission). All these values are calculated for the whole grid.

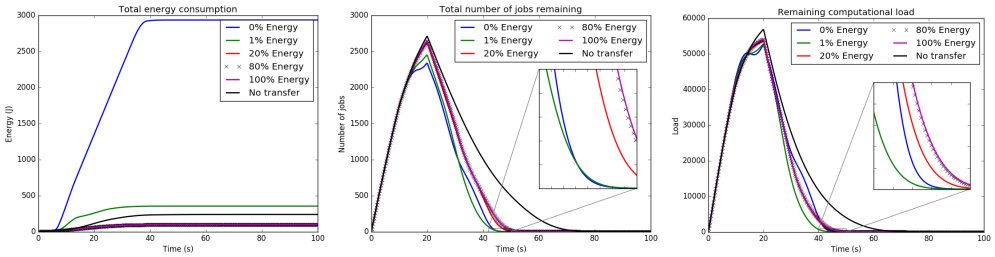

Figure 4: policies efficiency with respect with the energy consumption (left), number of jobs remaining (middle) and the total workload remaining (right)

Without any surprise, enabling transfers allows for smaller computational times in any scenario, even for the one focused only on energy consumption, since a lot of jobs are submitted in Cluster 0 whereas the grid contains better clusters in every aspect. Conversely, focusing only on the time aspect leads to a much higher energy consumption than the reference case, which is easily explained by the fact that the fastest cluster has a very high energy consumption compared to the two others.

On the middle and right figures, all curves seem to be well-ordered with respect to the importance of execution time, except for two of them which correspond to the scenarios with 0% and 1% energy. However, when zooming near the completion time for the two of them, it can be seen that the case with 0% energy does manage to finish before the other, even if barely. This can be explained by a transfer of jobs with high computational load to the fastest cluster delaying the completion of some of them (due to the transfer time) but in the end allowing to gain some seconds (or even milliseconds) to complete all of them. These curves also show that taking into account the consumption, even for a very small amount, can save a lot of energy while having a quite negligible impact on the computational time.

It has to be noted that the transfer decision is based only on the state of the grid at the time of the transfer, without having any knowledge of what will be submitted in the future, and that in our model no job can be transferred back so that it makes room for another one. Hence there is no guarantee that, for a given scenario, lowering the focus on energy will lead to a better global computational time. That makes the existence of a quick-time simulator even more attractive.

Dynamics split for every cluster. The results of energy consumption, occupation rate (normalized) and number of jobs ended for every cluster are displayed on figure 5. Each line corresponds to a different cluster while each column focus on one aspect (energy, occupation, jobs).

The results shown on figure 5 are in accordance with the global ones (figure 4) and what can be expected from the grid dynamics. For example, the occupation rate shows that the Cluster 0 is not the preferred one regardless of the policy used (as soon as transfers are enabled), which is in accordance with its characteristics. Both the occupation rate and the number of jobs ended grow on the cluster 2 as the focus on the energy decreases. The effect of the occupation rate on the consumption of the cluster (cooling system (45)) can be seen with energy-focused policies as some waves appear on the occupation rate as well as on the number of jobs ended.

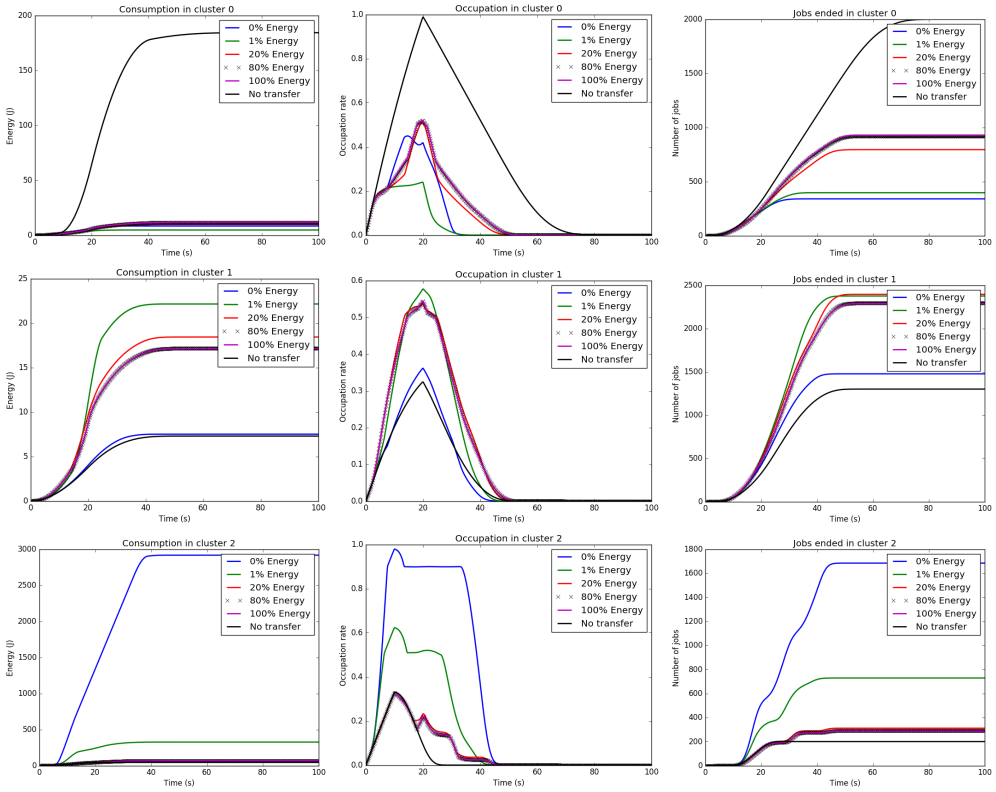

Figure 5: Impact of the grid policies on the energy consumption (left column), occupation rate (middle column) and number of jobs ended (right column) for cluster 0 (top line), cluster 1 (middle line) and cluster 2 (bottom line)

Transfers. The amounts of transfers performed between the clusters are all represented on figure 6.

As prescribed in our model, no transfer is allowed from one cluster to itself (flat lines on the diagonal of figure 6). As expected, the number of transfers towards Cluster 2 rises when the focus on energy decreases. It can be noted that some transfers exist from cluster 2 to cluster 1 when energy is the main focus. In these cases, even transfers from cluster 2 to cluster 0 exist, suggesting that the occupation rate of the cluster 1 yields a rise of its consumption. Yet, this amount remains negligible.

It can also be noted that the timing for the transfers from cluster 1 to cluster 2 are very different for the scenarios 0%- and 1%- energy, the 0% case having a bigger transfer occurring later in the simulation. This explains the crossing lines for both the number of jobs and computational load remaining observed on figure 4.

6. Conclusion and prospects

Starting from a microscopic model based on the realistic behavior of jobs submitted in clusters, and from a generic description of the job transfer policy between different clusters of the

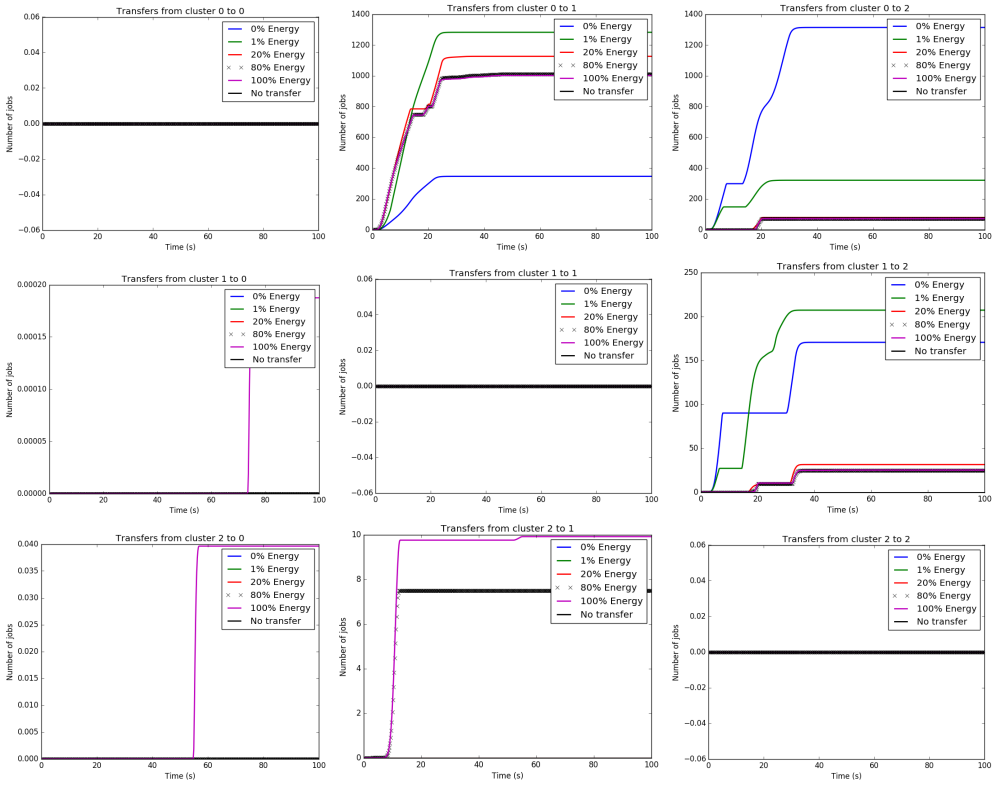

Figure 6: Impact of the grid policies on the number of transfers between clusters, following the structure of table 4

same computing grid, we proposed a derivation of a macroscopic model able to represent the grids dynamics.

Existence and uniqueness of a global solution to the resulting fluid system can then be proven under minimal assumptions of regularity of the decision function used for the transfer of jobs. This guarantees that the model covers a wide range of decision functions used for the job transfers, where the associated cost-functions can be seen as a "black-box". Hence, this kind of model enables the evaluation of the impact of a transfer policy on the performance of the cluster.

Using a finite volume discretization of the fluid equations and a simple cost function mixing both the aspects of computing time and energy consumption, we have illustrated a simple test case exhibiting a non trivial dynamics. It shows that our model has a consistent behavior with the one which can be expected from a computing grid. The ability of the numerical model to provide real-time estimates of the performance of a given set of policies for one job submission scenario, has also been highlighted.

This kind of model can easily be extended to a more complex set of parameters. For example, on the cost function side, it would be quite straightforward to include the mean time of execution for each job or the different kinds of fees paid for the use of the cluster. An interesting generalization of the model would be to take into account, at the microscopic level, the memory size of each job (as a new variable) which would impact its transfer time. This would add one

more conservation equation (on the mean memory size of the jobs) in the system (21). Implementing this aspect and using a known submission history would allow to simulate a real cluster dynamics, like the Grid'5000 Cluster in France [22].

One important assumption in our model is the fact that N remains constant through time, implying that all jobs are always tested, even the finished ones. If we allow the middle-ware to only focus on the jobs in execution state, it would introduce a variable characteristic time $\mathcal{T}(t)$, yielding to a more complex analysis. This aspect will be tackled in future studies. Finally, another extension of our model may focus on the effect of *exceptional* (oversized or prioritized for example) jobs on the cluster dynamics. Such jobs should be numerically treated with a specific particular solver which would need to be coupled with our fluid model through source terms and/or dynamic values for the clusters characteristics. Such an approach may allow to tackle some practical problems such as cluster shutdowns.

Acknowledgements

The authors have to thank G. Da Costa, from IRIT, for many fruitful discussions about the grids architecture and for providing some insight on realistic behaviours and values.

References

[1] W. V. Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, P. Demeester, Trends in worldwide ict electricity consumption from 2007 to 2012, Computer Communications 50 (2014) 64 – 76. Green Networking.

[2] G. Cook, J. Lee, T. Tsai, A. Kong, J. Deans, B. Johnson, E. Jardim, Clicking clean: Who is winning the race to build a green internet?, Greenpeace International, Amsterdam, The Netherlands (2017).

[3] A. S. Andrae, T. Edler, On global electricity usage of communication technology: trends to 2030, Challenges 6 (2015) 117–157.

[4] M. Zakarya, L. Gillam, Energy efficient computing, clusters, grids and clouds: A taxonomy and survey, Sustainable Computing: Informatics and Systems 14 (2017) 13–33.

[5] R. Ranjan, A. Harwood, R. Buyya, et al., Grid federation: An economy based, scalable distributed resource management system for large-scale resource coupling, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia (2004).

[6] A. Varasteh, M. Goudarzi, Server consolidation techniques in virtualized data centers: A survey, IEEE Systems Journal 11 (2017) 772–783.

[7] W. Pikatek, A. Oleksiak, G. Da Costa, Energy and thermal models for simulation of workload and resource management in computing systems, Simulation Modelling Practice and Theory 58 (2015) 40–54.

[8] G. Da Costa, T. Fahringer, J.-A. Rico-Gallego, I. Grasso, A. Hristov, H. D. Karatza, A. Lastovetsky, F. Marozzo, D. Petcu, G. L. Stavrinides, D. Talia, P. Trufio, H. Astsatryan, Exascale machines require new programming paradigms and runtimes, Supercomputing Frontiers and Innovations, Sustainability in ultrascale computing systems Hors-série (2015) (on line). Nesus.

[9] N. Bellomo, C. Dogbe, On the modeling of traffic and crowds: A survey of models, speculations, and perspectives, SIAM review 53 (2011) 409–463.

[10] P. Degond, Mathematical models of collective dynamics and self-organization, arXiv preprint arXiv:1809.02808 (2018).

[11] N. Bellomo, J. Soler, On the mathematical theory of the dynamics of swarms viewed as complex systems, Mathematical Models and Methods in Applied Sciences 22 (2012) 1140006.

[12] G. Albi, N. Bellomo, L. Fermo, S.-Y. Ha, J. Kim, L. Pareschi, D. Poyato, J. Soler, Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives, Mathematical Models and Methods in Applied Sciences 29 (2019) 1901–2005.

[13] D. Ros, R. Marie, Loss characterization in high-speed networks through simulation of fluid models, Telecommunication Systems 16 (2001) 73–101.

[14] Y. Liu, F. Lo Presti, V. Misra, D. Towsley, Y. Gu, Fluid models and solutions for large-scale ip networks, in: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 91–101.

[15] J. Incera, R. Marie, D. Ros, G. Rubino, Fluidsim: a tool to simulate fluid models of high-speed networks, Performance Evaluation 44 (2001) 25–49.
[16] M. A. Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, A. Tarello, Using partial differential equations to model tcp mice and elephants in large ip networks, IEEE/ACM Transactions on Networking 13 (2005) 1289–1301.
[17] G. Da Costa, G. Dufour, D. Sanchez, Modèles fluides pour l'économie d'énergie dans les grilles par migration: une première approche, in: Renpar'19, pp. 1–8.
[18] A. De Cecco, Fluid Modeling for Network Dynamics, Theses, Universite de Toulouse, 2016.
[19] F. Bouchut, F. James, One-dimensional transport equations with discontinuous coefficients, Nonlinear Analysis 32 (1998) 891.
[20] P.-E. Jabin, Various levels of models for aerosols, Mathematical Models and Methods in Applied Sciences 12 (2002) 903–919.
[21] L. Grange, G. Da Costa, P. Stolf, Green it scheduling for data center powered with renewable energy, Future Generation Computer Systems 86 (2018) 99–120.
[22] Grid'5000, Grid'5000 homepage, `https://www.grid5000.fr/w/Grid5000:Home`, 2020.