PDE FLUID MODELING OF MULTITHREAD SYSTEMS AND OPTIMAL CONTROL

Florian De Vuyst
Ecole Centrale Paris, laboratoire de Mathématiques Appliquées aux Systèmes
Grande Voie des Vignes, 92 295 Châtenay-Malabry cedex FRANCE
devuyst@mas.ecp.fr

Pascal Jaisson
Ecole Centrale Paris, laboratoire de Mathématiques Appliquées aux Systèmes
Grande Voie des Vignes, 92 295 Châtenay-Malabry cedex FRANCE
jaisson@mas.ecp.fr

Abstract We here propose fluid models of a multithread/multitask system. This can be straighforwardly extended to a network of such components. The leading equations form a system of hyperbolic equations coupled by a nonlocal term of current total load. PDEs are also used for the computation of the service times. A numerical stable and efficient method of discretisation is then proposed. To illustrate the usefulness of such models, we numerically solve a problem of optimal control of quality of service (QoS) management and demonstrate the efficiency of the method.

Keywords: System modeling, fluid modeling, buffer system, multithread system, partial differential equation (PDE), optimal control, quality of service, genetic algorithms

Introduction

It is of interest to try to mimic principles of Fluid Mechanics to other fields and disciplines, especially for process and flux management. About pioneer works in this direction, let us mention Whitham's book [8]; the author proposes conservation laws for modeling roadway traffic flows. Those PDEs are shown to be able to capture traffic bottlenecks with formation of nonlinear shock waves.
The idea to find high level mean flow equations is particularly interesting

for information system (IS) management where billions of events commonly travel the system, what makes discrete event simulation (DEVS) irrelevant for large scale and heavy traffic systems. Liu *et al* [7], [6] did a comparison of efficiency of network simulation between using fluid model and packet-level models.
In Fluid Dynamics, if one looks at the mean flow at the macroscopic level, modeling often gives conservation laws of the form

$$\partial_t U \ + \ \nabla \cdot \mathbf{F}(U) \ = \ 0.$$

If one neglects the viscous effects of the fluid, these equations are called the Euler equations. For certain simple fluids, we know that the solutions of these equations are (at least in a formal way) limits of solutions of equations that describe the microscopic behaviour of the fluid (collision between particles, mean free transport). These microscopic equations belong to the family of particle transport equations, like Boltzmann's equation. Sometimes, we do not have clear connection between macroscopic models and microscopic ones. This is due to the fact that fluids are too complex so that we cannot express complex mechanisms of particle interaction into limit equations of state. In this case at the macroscopic level, we only express conservation of certain quantities and model the fluid finely enough to reproduce the most features of its macroscopic behaviour.

In the Computer Simulation community, the most usual models are based on microscopic equations that finely model event scheduling. At this scale, simulation need stochastic models of sources (those are called injectors) with adapted laws of probability.
When networks systems are of small size, such models give results in reasonable computational times. But when networks become larger or when too many events have to be scheduled, such microscopic models become irrelevant because of the prohibitive computational times.
Moreover, for decision making purposes and especially for system performance enhancement, we do not generally need so rich models. We here adopt the fluid macroscopic approach, trying to find convenient conservation laws for particular high-level architectures or policies.
About related works of mean flow equations applied to information systems, let us for example mention the works from Baccelli, McDonald and Reynier [2] for modeling multiple internet TCP connections. Basing their arguments from classical stochastic models, they find some mean flow evolution equations by passing to the fluid limit.
The original feature of the present work is to use PDE models rather than usual ODE ones for buffer system modelling. The introduction of a

space-like variable allows us to take into account the memory and delay features using a transport process at a particular velocity of propagation of information. We also here extend our previous works of queuing and buffer system modeling (see De Vuyst [3], [4]) to multithread/multitask systems, and demonstrate that this framework can be utilized for performance analysis and optimisation purposes.
Our article is organized as follows. In section 1, we first explain how to find the fluid equations of a multithread system; we comment the domain of validity of such models; we also comment the extension to a systems with multiple arrival sources and the extension to a network of multithread servers. In section 2, we present our formalism to compute mean service times; this is an essential task for performance analysis. In section 3, we propose adapted numerical discretizations of such coupled systems of ODE-PDE. Finally, sections 4 and 5 are numerical experiments and validate the numerical method on two different problems of optimal control of Quality of Service (QoS).

1. Continuous formulation of multithread processing systems

1.1 Continuous modeling

Let us first define the capacity ϕ_o (index o means *output*) of a system the rate of which it can deliver services sequentially. This is equivalent to consider a mean service time $D = (\phi_o)^{-1}$ for a given class of requests. Suppose that the system is able to split up its capacity into small pieces for multithread processing. Consider N levels of task completion. At level $1/N$ ($k = 1$), the just arrived request begins to be served by the system. At final level N/N ($k = N$), the request is completed and served. More generally, suppose that, at level k/N, the number of current requests is M_k. If the server divides its capacity homogeneously, then the optimal instantaneous flux of requests between levels k/N and $(k+1)/N$ is

$$\phi_{k,k+1} \;=\; \frac{M_k}{\sum_{l=1}^{N} M_l}\,\phi_o, \tag{1}$$

so that, considering all the partial fluxes, we retrieve the whole capacity:

$$\sum_{k=1}^{N-1} \phi_{k,k+1} \;=\; \phi_o.$$

Suppose now that the number of levels of completion and the total number of current requests are large. We would like to rewrite balance equations in the fluid limit when both N and $\sum_l M_l$ tend to infinity. We introduce a “mass density” $\rho(x,t)$ (we voluntarily use Fluid Mechanics terminology). Variables x and t respectively denote space and time. Let $\omega =]\omega_l, \omega_r[$ be a small interval in $]0,1[$. The quantity

$$\int_{\omega_l}^{\omega_r} \rho(x,t)\,dx$$

represents the quantity of current requests being processed between levels ω_l and ω_r. Because the system is conservative, we can write a mass balance between levels ω_l and ω_r. This can be expressed by

$$\frac{d}{dt}\int_{\omega_l}^{\omega_r} \rho(x,t)\,dx \;=\; (\text{flux at } \omega_l) \;-\; (\text{flux at } \omega_r). \tag{2}$$

The continuous equivalent of (1) is clearly

$$\frac{\rho(x,t)}{\int_0^1 \rho(y,t)\,dy}\,\phi_o$$

which defines current local flux between levels x and $x + dx$. Consequently, equation (2) is more precisely

$$\frac{d}{dt}\int_{\omega_l}^{\omega_r} \rho(x,t)\,dx \;=\; \frac{(\rho(\omega_l,t) - \rho(\omega_r,t))\,\phi_o}{\int_0^1 \rho(x,t)\,dx}. \tag{3}$$

By considering any interval ω and making tend $mes(\omega)$ to zero, we find the following Partial Differential Equation (PDE)

$$\partial_t \rho \;+\; \partial_x \left(\frac{\rho\phi_o}{\int_0^1 \rho(x,t)\,dx} \right) \;=\; 0,\;\; x \in]0,1[, t > 0, \tag{4}$$

where x is a space variable representing the level of completion of the current requests. This equation can be read as a transport equation

$$\partial_t \rho \;+\; \partial_x \rho v(t) \;=\; 0, \tag{5}$$

with time-varying velocity $v(t)$ equal to

$$v(t) \;=\; \frac{\phi_o}{\int_0^1 \rho(x,t)\,dx}. \tag{6}$$

It can also be read in a weak sense.

1.2 Domain of validity and stochastic limit

Of course, this continuous model is justified if there is enough request material to process, otherwise the current total mass of requests

$$m(t) \;=\; \int_0^1 \rho(x,t)\,dx$$

could tend to zero and thus the velocity of propagation of information $v(t)$ would tend to infinity, which does not agree with expected finite propagation velocity given by ϕ_o. The domain of validity of this continuous model is precisely the domain in which stochastic effects are not prevailing. However, it is natural to think that if the arrival rate is noisy, stochastic effects no more negligeable when m is of the order of 1. We would like our model be able to explore (at least coarsely) the stochastic-deterministic *transition* domain because this configuration occurs frequently in practice and is of large interest for capacity planning and risk analysis. For that, we can define a velocity which is more general than (6), depending on a continuous increasing function φ such that $\varphi(x) \geq 1$, $\varphi(-\infty) = 1$, $\varphi(x) \sim x$ for large x:

$$v(t) \;=\; \frac{\phi_o}{\varphi\left(\int_0^1 \rho(x,t)\,dx\right)}. \tag{7}$$

Thus, if we use for example the limiter function $\varphi(x) = \max(1,x)$, in the case where m becomes lower than 1, then the velocity v is exactly ϕ_o . In this case, if $\phi_i(t)$ denotes the arrival rate (index i means *input*), then the departure rate becomes at equilibrium

$$\rho(1,t)\phi_o \;=\; \rho(0,t-D)\phi_o \;=\; \frac{\phi_i(t-D)}{\phi_o}\,\phi_o \;=\; \phi_i(t-D).$$

Consequently, the departure rate is limited at the correct expected value (pure transport process).
About choices of functions φ, the following one-parameter family

$$\varphi^\epsilon(x) \;=\; \frac{(x-1)+\sqrt{\epsilon^2+(x-1)^2}}{2},\ \epsilon \in \mathbb{R}^+,\ x>0, \tag{8}$$

for rather "small" ϵ provides good candidates that have the benefit to be smooth with $\mathcal{C}^\infty$ regularity. The parameter ϵ should be calibrated and optimized from experimental real data. It is expected that we can find a law that links ϵ with some invariants of the stochastic process (variance of the Gaussian noise part of the arrival rate for example), but it is not at the aim of the paper and it is not discussed here. In what follows, we consider the following equation

$$\partial_t \rho \; + \; \partial_x \left(\frac{\rho \phi_o}{\max(1, \int_0^1 \rho(x,t)\, dx)} \right) \; = \; 0, \; x \in]0,1[, t > 0, \qquad (9)$$

as model equation.

Let us go back to the total mass $m(t)$ of current fluid in the system. We are looking for a differential balance equation on this variable. By integrating equation (5) on the interval $]0,1[$, we get the differential equation on m

$$\frac{dm}{dt} \; = \phi_i(t) \; - \; \rho(1,t)v(t), \qquad (10)$$

which corresponds to a balance between incoming and outgoing fluxes. Then, we can also formulate the equations as follows:

$$v(t) \; = \; \frac{\phi_o}{\max(1, m(t))}, \qquad (11)$$

$$\partial_t \rho \; + \; \partial_x \rho v(t) \; = \; 0, \; x \in]0,1[, \; t > 0, \qquad (12)$$

$$\frac{dm}{dt} \; = \phi_i(t) \; - \; \rho(1,t)v(t). \qquad (13)$$

For this initial value problem, we add some initial conditions: $m(0) = m^0 \geq 0$, and $\rho(x,0) = \rho^0(x)$ given such that $\int_0^1 \rho^0(x)\, dx = m^0$, $\rho^0 \in BV(0,1)$. Because of the hyperbolic nature of the equation, we also add upstream boundary conditions which correspond to a flux compatibility between the arrival rate $\phi_i(t)$ and the flux of equation (12). This is equivalent from defining upstream Dirichlet conditions on ρ according to

$$\rho(0,t)v(t) \; = \; \phi_i(t). \qquad (14)$$

1.3 Extension to a multiple source system

That formalism can be naturally extended to the multicomponent case where several sources of information have to be treated in competition by the same multithread server. Typical applications of interest are web applications with concurrent services. In this case, we consider M arrival rates $\Phi_{i;k}$, $k = 1, M$. Let us also denote the total arrival rate

$$\Phi_i \; = \; \sum_{k=1}^{M} \Phi_{i,k}.$$

Then, by the same modeling principles, we find a system of partial differential evolution equations for the partial densities ρ_k of information number k:

$$\partial_t \rho_k \; + \; \partial_x \frac{\rho_k \Phi_o}{\max(1, \sum_{l=1}^M \int_0^1 \rho_l(x,t)\,dx)} \; = \; 0,$$

The equations are coupled by the nonlocal term of total current load $m(t)$ now defined as

$$m(t) \; = \; \sum_{l=1}^{M} \int_0^1 \rho_l(x,t)\,dx.$$

The compatibility conditions of continuity of the vector flux at $x = 0$ give the upstream boundary conditions

$$\Phi_{i;k}(t) \; = \; \frac{\rho_k(0,t)\Phi_o}{\max(1, m(t))}$$

that can also be seen as nonhomogeneous Dirichlet conditions on each ρ_k at the left boundary. The velocity of propagation is the same for each source k, still given by expression (11). We can also reformulate the problem as a coupled nonlinear system of ODE-PDEs, like (11)-(13):

$$v(t) \; = \; \frac{\phi_o}{\max(1, m(t))}, \tag{15}$$

$$\partial_t \rho_k \; + \; \partial_x \rho_k v(t) \; = \; 0, \;\; x \in]0,1[, \; t > 0, \;\; k = 1, M, \tag{16}$$

$$\frac{dm}{dt} \; = \phi_i(t) \; - \; \left(\sum_{l=1}^{M} \rho_l(1,t) \right) v(t). \tag{17}$$

1.4 Comments about the extension to a network of multithread servers

So far, we have modeled the behaviour of only one server with its own multithread policy. But a complete electronic process is generally made of several interconnected servers that communicate and form a network. Thus, we need to add some flux interfaces through all couples of nodes. If we suppose that the network has infinite bandwidth, we only give us flux compatibility conditions between departure and arrival rates of two neighbouring nodes. The hyperbolic nature of the equations leads us to impose prescribed upstream boundary conditions for a node. This means that the arrival rate is given. On the other hand, the outflow

(departure rate) is a consequence of the internal flow inside a node. If a node number (1) sends information to the node (2), then we only say that the arrival rate of (2) is the departure rate of node (1) which depends on the internal state of (1).
To summarize, let say that our models attach PDEs (or systems of PDEs) at each node of the network (behaviour model), whereas the model of communication is expressed by some flux compatibility conditions. This can of course be extended to the vector multi-source case.

2. Computation of the service times

If we want to use that model for performance assessment purposes, we more need to know the service delays at any time. This is equivalent to find the time passed through each particle to go from $x = 0$ to $x = 1$. From the differential point of view (integration of characteristics), this time depends on both Past and Future: when a request "enters" the system, the state of the system depends on the past because of memory effects. When this request is being processed, its effective trajectory will depend on future arrival rates. Those last ones can for example slow down, see block the system (case of stagnant particles). From the PDE point of view, time memory can be changed into space memory. Let us denote $d(x, t_0, x_0)$ the travel time of a request particle at position x with old position $x_0 \in]0, 1[$ at time t_0. Using the particle derivative and Lagrangian representation, we need to solve the differential problem

$$\frac{D}{Dt} d = 1,$$

$$d(x, t_0, 0) = 0.$$

But we also know that, at time t the particle velocity is given by $v(t)$. Then the Eulerian representation of the problem is then given by the PDE problem:

$$\partial_t d + v(t)\, \partial_x d = 1, \quad x \in]0, 1[, \ t > 0, \tag{18}$$

$$d(x = 0, t) = 0. \tag{19}$$

Because we want to know the whole residence time of a particle into the system, the variable of interest is in fact the service time $D(t)$ given by

$$D(t) = d(1, t). \tag{20}$$

Problem (18),(19) is a transport problem with constant source term. Let us remark that, by the simple change of variable $u = t - d$, we can also formulate this last problem as an homogeneous one (free of source term):

$$\partial_t u \; + \; v(t)\, \partial_x u \; = \; 0, \quad x \in]0,1[, \; t > 0, \tag{21}$$

$$u(x=0,t) = t, \tag{22}$$

$$D(t) = t - u(1,t). \tag{23}$$

In what follows, we will name equations (21)-(23) the Service Time Equations (STE).

3. Numerical discretization of equations

We consider a uniform space discretization of the interval $]0,1[$ with computational points $x_j = (j - \frac{1}{2})h$, $j = 1, N$ and space step $h = \frac{1}{N}$. We also consider a time step τ^n at time t^n and then define the next discrete time t^{n+1} as $t^{n+1} = t^n + \tau^n$. Let us denote by $\lambda^n = \frac{\tau^n}{h}$ the ratio of discretization steps at time t^n. Because of the hyperbolic nature of the transport equations, we consider classical upwind schemes for stability purposes. The scheme, written in its semi-implicit form, reads

$$\rho_j^{n+1} \; = \; \rho_j^n \; - \; \lambda^n \frac{(\rho_j^{n+1} - \rho_{j-1}^{n+1})\phi_o}{\max(1, h \sum_{l=1}^N \rho_l^n)} \tag{24}$$

which leads to an unconditionally stable scheme. About implementation considerations, we need to solve a two-diagonal triangular linear system at each time step, which can be in fact written and solved explicitely so that the numerical complexity is of the order of the one of an explicit scheme.
In expression (24), we have discretized $m(t^n)$ by the truncated Riemann series $h \sum_{l=1}^N \rho_l^n$. One could ask if this quadrature formula is not too coarse, unadapted or irrelevant. On the contrary, this choice allows us to enforce a consistency relation with the balance equation (10) at the discrete level (while staying first order accurate). Indeed, because the scheme is conservative, summing up all the indices j in (24) leads to the following expression

$$\sum_{j=1}^N \rho_j^{n+1} \; = \; \sum_{j=1}^N \rho_j^n \; - \; \lambda^n \left(\frac{\rho_N^{n+1}\phi_o}{\max(1, m^n)} - \phi_i^n \right)$$

or again

$$m^{n+1} \; = \; m^n + \tau^n \left(\phi_i^n - \frac{\rho_N^{n+1}\phi_o}{\max(1, m^n)} \right)$$

which is consistent with the balance equation (10) of m.
To summarize, we implement the numerical method as follows:

$$\begin{aligned} v^n &= \frac{\phi_o}{\max(1, m^n)}, && (25) \\ \rho_1^{n+1} &= \rho_1^n - \lambda^n(\rho_1^{n+1}\, v^n - \phi_i^n), && (26) \\ \rho_j^{n+1} &= \rho_j^n - \lambda^n(\rho_j^{n+1} - \rho_{j-1}^{n+1})\, v^n,\ j = 2, N, && (27) \\ m^{n+1} &= m^n + \tau^n \left(\phi_i^n - \rho_N^{n+1}\, v^n\right). && (28) \end{aligned}$$

Let us recall that, although equations (27) are implicit, the computational cost of the scheme is of the order of an explicit one (up to scalar divisions) because the resulting linear system is two-diagonal triangular.

Finally, the numerical discretization of the STE is straightforward. We use the homogeneous formulation (21)-(23) and propose the full implicit numerical scheme

$$d_1^{n+1} = t^{n+1}, \quad (29)$$

$$d_j^{n+1} = d_j^n - \lambda^n \left(d_j^{n+1} - d_{j-1}^{n+1}\right) v^n, \quad j = 2, N, \quad (30)$$

$$D^{n+1} = t^{n+1} - d_N^{n+1}. \quad (31)$$

Let us that emphasize that, even if the services times are very small with respect to the current time step, they are however accurately computed. That makes this numerical method very powerful for computing large scale heavy traffic dynamical systems. In our numerical experiments, we observe that we can use large CFL numbers up to 10^6 while keeping very good estimates of the services times for smooth inflows.

4. Application: optimization of capacity sharing between two sources

Let us consider a server with crude processing capacity denoted by ϕ_o (which is homogeneous to a rate). Two different sources (1) and (2) arrive at the server and are treated separately in parallel. For that, the real server is logically splitted up into a network of two virtual servers numbered (1) and (2) with independent parallel multithread policy.
Of course, the crude common capacity ϕ_o has also to be shared between the two virtual servers. We then introduce a time-varying function $\alpha(t) \in [0, 1]$ and two local respective capacities $\alpha(t)\phi_o$ and $(1 - \alpha(t))\phi_o$.

The leading equations thus are

$$\partial_t \rho_1 \ + \ \partial_x \rho_1 v_1(t) \ = \ 0, \tag{32}$$

$$\partial_t \rho_2 \ + \ \partial_x \rho_2 v_2(t) \ = \ 0, \tag{33}$$

with respective transport velocity and mass

$$v_1(t) \ = \ \frac{\alpha(t)\phi_o}{\max\left(1, m_1(t)\right)}, \ m_1(t) = \int_0^1 \rho_1(x,t)\, dx, \tag{34}$$

$$v_2(t) \ = \ \frac{(1-\alpha(t))\phi_o}{\max\left(1, m_2(t)\right)}, \ m_2(t) = \int_0^1 \rho_2(x,t)\, dx. \tag{35}$$

For well-posedness, we are looking for a function α belonging to the functional space $BV(0,T)$ with values in $[0,1]$.
The corresponding respective STE and services times $D_1(t)$ and $D_2(t)$ for each virtual server (1) and (2) are solutions of

$$\partial_t d_\ell + v_\ell \partial_x d_\ell = 1, \ x \in]0,1[, \ d_\ell(0,t) = 0, \ D_\ell(t) = d_\ell(1,t) \tag{36}$$

for each $\ell = 1,2$. Because v_ℓ depends on α, it is clear that each D_ℓ also depends on the time function α. Below, we explicity exhibit that dependency by using the notation $D_\ell = D_\ell(t;\alpha)$, $\ell = 1,2$.
We are interested in an optimal control problem set in the time window $]0,T[$. We introduce two constant parameters of Quality of Service (QoS) of the system $D_{max;1}$ and $D_{max;2}$. Those are bounds of service delay. We are looking for a function $\alpha \in BV(0,T;]0,1[)$ that realizes the minimum of functional J defined by

$$J(\alpha) \ = \ \frac{1}{T}\int_0^T \left\{ (D_1(t;\alpha) - D_{max;1})^2 + (D_2(t;\alpha) - D_{max;2})^2 \right\} \, dt \tag{37}$$

Because the problem is formulated in a space of infinite dimension, we need to approximate it for its numerical resolution. We then formulate a finite dimensional optimization problem of unknown vector α,

$$\alpha \ = \ (\alpha^1, \alpha^2, ..., \alpha^M),$$

where α^n, $n = 1, M$ is an approximation of $\alpha(t^{M;n})$ with $t^{M;n} = \frac{(n-1)T}{M}$.
We introduce a time-continuous reconstruction operator $\mathcal{J}$ from $\mathbb{R}^M$ into the space of piecewise constant functions such that, for all n,

$$\mathcal{J}\alpha_{|[t^{M;n}, t^{M;n+1}[} \ = \ \alpha^n, \ n = 1, M.$$

The resulting functional is then given by J^M,

$$J^M(\alpha) \;=\; \frac{1}{T}\int_0^T \left\{(D_1(t;\mathcal{J}\alpha) - D_{max;1})^2 + (D_2(t;\mathcal{J}\alpha) - D_{max;2})^2\right\} dt \tag{38}$$

and the final optimal control problem is to find the vector α that realizes

$$\min_{\alpha\,=\,(\alpha^1,\alpha^2,\ldots,\alpha^M)} J^M(\alpha).$$

For solving the optimization problem, we decide to use a genetic algorithm (GA). We have used the GAOT Toolbox (see [1]) that can be found on the web. About the parameters of simulation, we use $\alpha \in \mathbb{R}^{48}$ (that means that the value of $\alpha(t)$ is updated each thirty minutes over 24 hours) and a population of 20 genes. All the initial genes (candidates for optimal vector α) are chosen randomly with components into $[0, 1]$. As experiment, we choose $\Phi_o = 120$ req/sec ($D = {\Phi_o}^{-1} = 8.3\ 10^{-3}$ sec), $D_{max;1} = 4$ sec and $D_{max;2} = 3$ sec. On figure 1, we first plot the two given profiles of arrival rates within a whole day of observation (24 hours from 0:00 a.m.). The rate of source (1) is chosen to be important during the morning period whereas source (2) is concentrated over the afternoon. The second plot compares the capacity of the server with respect to the sum of the two arrival rates. What we see if that there exists two threshold crossovers corresponding to two periods of congestion within the day (from 10 a.m. to 0:30 p.m. and from 5:30 p.m. to 7 p.m.). Finally, the third plot is the numerical optimal profiles of α obtained by the optimisation process. About only 60 iterations of GA are necessary to reach a vector very close to the global minimum so that the computational time stays reasonable. The fourth and fifth plots of figure 1 represent the profiles of service time for respective sources (1) and (2). In that configuration, we note that the constraints of quality of service are respected even if the system is congested two times within the day. Of course, a more congested configuration would violate the desired QoS, but anyway the optimal process would find a “best effort” solution. It is interesting to note that our model can capture the uncongested flow as well as the congested one with no difficulty of transition. In particular, this simulation handles services times of order $D = 8.3\ 10^{-3}$ sec up to congested response times of order 4 sec, what makes a ratio of about 500 between the two extreme bounds. Notice finally that the present problem can have several minima, and different initialization parameter can lead to different numerical minima. But it is observed that all the mimima stay close together, and, more precisely are close to the following time function in closed form:

$$\alpha^c(t) \;=\; \frac{\Phi_1(t)}{\Phi_1(t) + \Phi_2(t)}$$

which corresponds to the fraction of source (1) that enters the system. Consequently, this function, which is not far from the optimal solution, should be used for the real-time control as the control policy for that particular system.

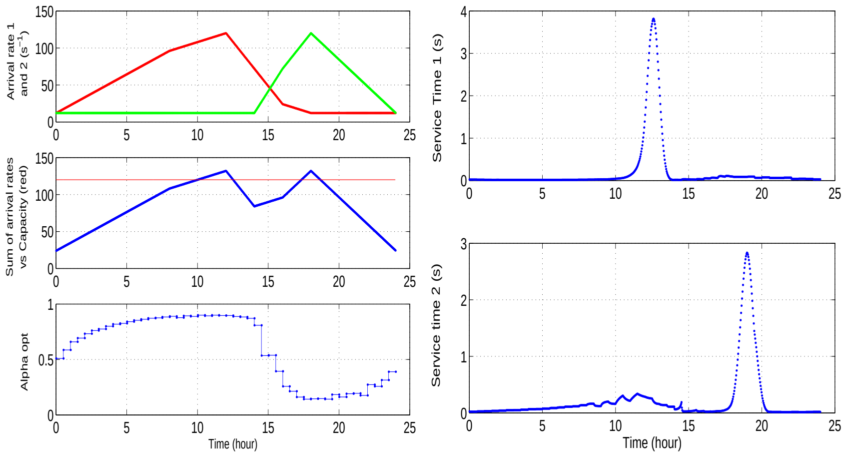

Figure 1. Optimization of capacity sharing between two sources. The sum of arrival rates compared to departure rate (capacity) shows the two regions of bottleneck. Optimized service times for sources (1) and (2) and corresponding α profile within a complete day are shown.

5. Application: two-source problem with guaranteed quality of service for the first source

Let us now consider a case with two sources $\phi_{i,1}$ and $\phi_{i,2}$ such that $\phi_{i,1} << \phi_o$ (source 1 is minority), but possibly $(\phi_{i,1}+\phi_{i,2})$ can be greater than ϕ_o (case of congested system). The first source 1 is expected to be processed with a prescribed guaranteed quality of service (QoS) expressed in terms of service time bounded by a constant $D_{max;1}$. As in the previous problem, the crude capacity of the real system is splitted up into two virtual multitask systems of respective capacity $\alpha(t)\Phi_o$ and $(1-\alpha(t))\Phi_o$. The first source partially feeds the first virtual system whereas the second one feeds both virtual systems with respective fluxes

$\beta(t)\phi_{i,2}(t)$ and $(1-\beta(t))\phi_{i,2}(t)$ (see figure 2).
The question is to both optimally feed the first virtual server with the second source and distribute the total capacity Φ_o while lowering as best as possible the service time of the second server. The optimization problem is subject to the QoS constraint on the first source

$$D_1(t) \;\leq\; D_{max;1} \;\forall t \in]0,T[.$$

We decide to introduce the following functional

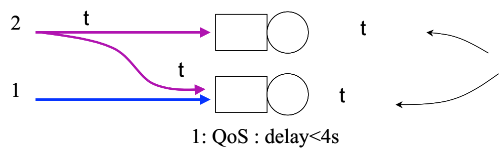

Figure 2. Description of the architecture of the second problem

$$J_\sigma(\alpha,\beta) \;=\; \sigma \max_{t\in[0,T]} \left(D_1(t;\alpha,\beta) - D_{max;1}\right)_+ \;+\; \frac{1}{T}\int_0^T \left(D_2(t;\alpha,\beta)\right)^2 dt, \tag{39}$$

where σ is a (large) weight coefficient that controls the force of the first term of the functional with respect to the second one. We have somewhat relaxed the constraint of quality of service by considering a large penalty term into the functional (39). The second term is aimed at lowering at best the service time of server number 2 for the optimal solution. A large value of σ gives priority on the first term of the functional and thus allows us to almost verify the guaranteed QoS. Remark that we here mix $L^\infty(0,T)$ and $L^2(0,T)$ norms. The resulting functional J_σ has low regularity because of the presence of the L^∞-norm. That prevents us from using pure gradient methods and rather invites us to use genetic algorithms (GA) for the numerical optimization process. About implementation, we still use the GAOT toolbox [1]. About the parameters of simulation, we now use $\alpha,\beta\in\mathbb{R}^{24}$, that means that the values of $\alpha(t)$ and $\beta(t)$ are updated each hour. We use again 20 genes, $\Phi_o = 120$ req/sec, $D_{\max;1} = 4$ sec. The results are given on figure 3. The given profiles of sources (1) and (2) now generate a congestion between 3:00 p.m. and 5:00 p.m (first and second plots). The third and fourth plots show the optimal profiles of piecewise constant time functions α and β. The fifth and sixth plots present the resulting optimal profiles of service time for the servers (1) and (2). What we see is that the constraint of guaranteed QoS is met (even if the limit bound at 4 seconds is attained in the

congested region). On the other hand, during the bottleneck, the service time of server (2) increases up to 550 seconds (about 9 minutes).
Although conditions of this test case apparently seem not to be too stiff, human tentatives of reasonable choice of profiles of functions α and β often lead to mean services times of the order of 10000 seconds with a complete violation of the guaranteed QoS ! That justifies the need of control automatons for such kinds of systems.

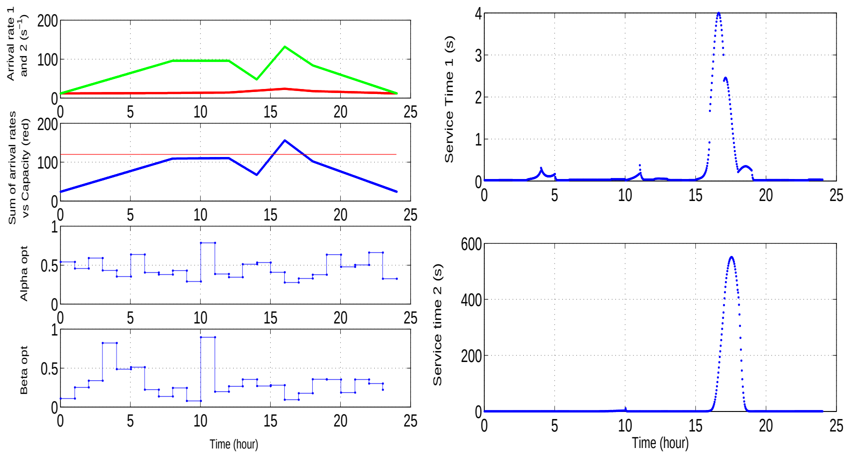

Figure 3. Optimisation of routing and capacity sharing for two sources and two virtual servers with a prescribed guaranteed QoS for the first source.

6. Concluding remarks and perspectives

We have proposed a PDE-ODE fluid model of multithread systems. The coupled system of transport equations is original in its form by the presence of a nonlocal term that expresses the congestion rate of the system. We have also proposed a new formalism for service time assessment which is also based on transport equations. That formalism notably improves a previous technique exposed in [5] that was based on level set methods for particle tracking. The equations exposed here allow for a better understanding of the behaviour of multithread systems like certain web servers or phone telecommunication value chains. Although those models have been initially contructed under the assumptions of heavy traffic and congestion regime, we have numerically demonstrated that they are also able to handle uncongested-congested regime transition and compute accurately mean service times at least in the congested case. Consequently, those are adapted for performance analysis because criteria of performance essentially depend on large response time due to congestion. As example and experiments, we have proposed two test cases of optimal control of quality of service and proved the utility of such models. About future works, we plan to refine the models where stochastic effects can no more be neglected. It would be also interesting to couple event-level and fluid level models with rigorous transition modeling and analysis. Finally, a promising work is to use such models for real system identification with low number of degrees of freedom. Parallely, we think that this kind of PDEs could be used in the context of time-dependent artificial neural networks for modelling memory-based systems.

References

[1] GAOT, the genetic algorithm optimization toolbox for matlab 5. http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/.

[2] F. Baccelli, D. R. McDonald, and J. Reynier. A mean-field model for multiple tcp connections through a buffer implementing red. Technical Report Tech. Rep. RR-4449, INRIA France, April 2002.

[3] F. De Vuyst. Feasibility of fluid transport modelling for buffer and processing systems : Information fluid dynamics, 2002. http://www.math.ntnu.no/conservation/2002/009.html.

[4] F. De Vuyst. *Fluid modelling of buffer and processing systems, finite volume discretization.* The International Symposium on Finite Volume for Complex Applications III (FVCA 3). Herbin and Kröner Eds, Hermes Penton Science, 2002.

[5] F. De Vuyst. Service time assessment in computational information fluid dynamics using level set methods, 2002. http://www.math.ntnu.no/conservation/2002/010.html.

[6] B. Liu, D. Figueiredo, Y. Guo, J. Kurose, D. Towsley, and W Gong. A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In *Proceedings of IEEE INFOCOM 2001*, April 2001.

[7] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. Gong. Fluid simulation of large scale networks: issues and tradeoffs. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, volume IV, pages 2136–2142, Las Vegas, 1999.

[8] G. B. Whitham. *Linear and Nonlinear Waves.* Wiley, John and Sons, ISBN: 0471359424, 1999.