

# **Appel à projet 2007 pour le DIM VI**

## **« Logiciels et Systèmes Complexes »**

### **du RTRA Digito**

#### *Projet de recherche VEHICULAIRE*

*Nous présentons ici un projet sur la conception d'un logiciel de calcul de haute performance pour la vérification (model checking) d'une algèbre de réseaux de Petri de haut-niveau. Ce logiciel sera basé sur un algorithme parallèle avec équilibrage de charge dynamique et programmé de manière modulaire dans un langage parallèle de haut-niveau. Cette vérification pourra s'appliquer à des modèles tant informatiques (programmes de calcul haute performance comme des programmes C+MPI) que physiques ou biologiques. La demande pour ce projet est essentiellement une allocation doctorale.*

# I Généralités sur le projet VEHICULAIRE

## a) Résumé et laboratoires concernés

**Titre du projet :** Vérification Efficace de Haut-niveau multi-proCesseeur et modULAIRE (VEHICULAIRE)

**Mots-clés :** vérification, modularité, parallélisme de haut-niveau, réseaux de Petri, modélisation

**Laboratoires et responsables :**

- 1) **Laboratoire d'Algorithmique, Complexité et Logique** (EA 4213), laboratoire francilien du projet
  - a. *Directeur :* [Gaétan Hains](#)
  - b. *Adresse complète :* LACL, Université Paris XII-Val de Marne,  
UFR de Sciences et Technologie, Bâtiment P2, 2<sup>ème</sup> étage  
61 avenue du Général de Gaulle, 94010 Créteil
  - c. *Téléphone :* 01 45 17 66 00
  - d. *Télécopie :* 01 45 17 66 01
  - e. *Site Internet :* <http://www.univ-paris12.fr/lacl/>
- 2) **Laboratoire d'Informatique Fondamentale d'Orléans** (EA 4022), laboratoire partenaire non francilien
  - a. *Directeur :* [Christel Vrain](#)
  - b. *Adresse complète :* LIFO, Université d'Orléans  
Bâtiment IIIA, Rue Léonard de Vinci  
B.P. 6759, F-45067 ORLEANS cedex 2
  - c. *Téléphone :* 02 38 41 72 89
  - d. *Télécopie :* 02 38 41 71 37
  - e. *Site Internet :* <http://www.univ-orleans.fr/lifo/>
- 3) **Coordinateur du projet :**
  - a. [Frédéric Gava](#)
  - b. *Fonction :* MCF au LACL
  - c. *Email :* [gava@univ-paris12.fr](mailto:gava@univ-paris12.fr)

## b) Résumé et objectif du projet

La vérification automatique de modèles (appelé *model checking* [22] ) est une approche séduisante mais généralement coûteuse en termes de temps de calcul et d'espace mémoire nécessaires (on parle de l'explosion de l'espace des états ou « *state-space problem* »). Beaucoup de travaux dans le domaine sont dédiés à l'accélération des calculs et à la réduction de l'espace d'exploration. La parallélisation de ce calcul est l'une des techniques envisageables. Cependant, elle est en général employée avec des modèles de bas-niveau (tant au niveau du modèle étudié que du langage et du parallélisme utilisés pour implanter la *model checker*) et avec une distribution plus ou moins naïve des données.

L'objectif de ce projet est tout d'abord la conception d'un algorithme parallèle BSP de construction de l'espace d'exploration d'une algèbre de réseaux de Petri colorés de haut-niveau, appelé M-Net (ou sur un sous-ensemble clairement défini pour nos types d'applications) ainsi que de la vérification de propriétés logiques sur ce graphe (certaines logiques comme LTL [21] ou autres sont souvent très expressives mais un sous-ensemble expressif et vérifiable efficacement sera nécessaire). L'utilisation d'un tel modèle fortement structuré nous permettra d'en déterminer des caractéristiques structurelles ouvrant la voie à un parallélisme de haut-niveau (plus structuré) et donc plus efficace (et portable) qu'est le modèle BSP. Ensuite, une implantation modulaire et polymorphe (indépendante des types de données, donc idéale pour le *model cheking* symbolique et pour une algèbre de haut-niveau) sera effectuée avec une bibliothèque de programmation parallèle de haut-niveau, appelée BSMLlib (basé sur le langage OCaml [23] ), développée conjointement au LACL et au LIFO. Enfin, des tests appliqués de notre logiciel à des problèmes de sécurité informatique (thème fédérateur du LACL) ainsi que sur des propriétés de programmes hautes performances (C, Ada ou Fortran avec une bibliothèque de calculs parallèles telle que MPI) seront effectués sur les différentes grappes de PCs du LACL et du LIFO. Des adaptations et des ajouts seront par la suite (ou au fur et à mesure) ajoutés pour modéliser et vérifier des propriétés logiques de problèmes physiques ou biologiques qui seront rencontrés dans la littérature.

L'intérêt sociétal d'un tel logiciel est de pouvoir vérifier de manière automatique des propriétés logiques de grands systèmes tant industriels que scientifiques. En effet, les besoins de logiciels sûrs (c'est-à-dire fiables et répondant à leurs exigences) est un problème majeur pour la sécurité des systèmes car ils peuvent mettre en cause des vies ou de grands investissements humains. Prouvée

mathématiquement cette sûreté logicielle est complexe et coûteuse en temps. Un logiciel de calcul de haute performance adapté peut fortement réduire ces coûts et faire gagner des gains de productivité à l'industrie du logiciel sûr.

### **c) Étapes clés du projet**

Le projet (et donc essentiellement la thèse avec l'aide des permanents du projet) se déroulera en 3 étapes clés :

- 1) Etude bibliographique, conception du modèle et étude du bon sous ensemble logique à utiliser
- 2) Conception d'un algorithme parallèle de génération du graphe des états et d'un autre algorithme pour la vérification de propriétés logiques sur ce graphe : premières implantations et tests (dans le cadre d'un stage de Master 2 recherche)
- 3) Amélioration des deux algorithmes et leur intégration dans un seul et même logiciel ; tests de cas dans le cadre d'un stage de Master 2 professionnel.

### **d) Planning prévisionnel**

Le planning prévisionnel correspond aux trois années de thèse du doctorant ainsi que d'une quatrième année pour mettre en pratique, communiquer et distribuer les résultats par les membres permanents du projet. Il est le suivant :

*Première année :* bibliographie<sup>1</sup> sur les algorithmes de génération d'un graphe des états ; bibliographie sur les logiques employés pour le *model checking* ; sélection du bon sous-modèle à utiliser pour le *model-checking* ; sélection des bonnes propriétés logiques à vérifier (c'est donc une approche dédiée, pour maximiser l'efficacité de la vérification) ; premières études de cas (système biologique ou de sécurité, programmes MPI) afin de vérifier les bons choix du projet.  
**Livrable :** définition sur papier du modèle, format informatique du modèle, première modélisation d'étude de cas, définition sur papier des logiques et propriétés, format informatique de ces propriétés, première modélisation d'étude de cas.

*Deuxième année :* définition d'un premier algorithme parallèle pour la génération du graphe des états ainsi que de la vérification des propriétés logiques des systèmes ; première intégration des deux types d'algorithmes dans un seul et même algorithme.  
**Livrable :** définitions sur papier des algorithmes et premières implantations

*Troisième année :* optimisation et implantation des algorithmes ; intégration des deux algorithmes dans un seul et même outils de vérification ; premiers tests de cas réels.  
**Livrable :** premier outils logiciel de génération de graphes, définitions sur papier des algorithmes de vérification

*Quatrième année :* tests de cas industriels et scientifiques réels ; mise en pratique par des membres extérieurs au projet notamment avec des squelettes algorithmiques<sup>2</sup>.  
**Livrable :** logiciel complet de calcul haute performance pour le *model-checking*

---

<sup>1</sup> Notons qu'un premier état de l'art a déjà été effectué par les membres du projet afin d'estimer les possibilités et la faisabilité du projet.

<sup>2</sup> Un squelette algorithmique est une « fonction » qui peut être implantée en parallèle : chaque type de parallélisme (diviser-pour-régner, *pipeline* etc.) se voit doté d'un squelette ; cela permet de paralléliser facilement et de manière plus sûre des programmes, en utilisant des squelettes appropriés et qui « représentent » bien le parallélisme intrinsèque du problème

## II Détails du projet VEHICULAIRE

### **a) Thématique du projet**

La thématique principale du projet (et donc de la demande d'allocation doctorale) est la vérification parallèle c'est-à-dire la vérification de propriétés logiques de systèmes par calcul haute performance. Cette vérification se portera sur une algèbre de processus basée sur les réseaux de Petri mais seul un sous-ensemble fortement structuré (et donc plus facilement analysable) sera utilisé. Les algorithmes de vérification seront basés sur un modèle de parallélisme structuré qu'est le modèle BSP afin de permettre des analyses de coûts des calculs pour permettre un équilibrage des charges dynamique optimal.

### **b) Allocation doctorale**

Ecole Doctorale de rattachement : École Doctorale Sciences et Ingénierie : Matériaux - Modélisation – Environnement (SIMME) de l'Université de Paris 12

Nom du responsable de l'ED : Frank Pacard ([pacard@univ-paris12](mailto:pacard@univ-paris12))

Numéro de téléphone du responsable de l'ED : 01 45 17 65 99

### **c) Coordonnateur responsable du projet**

Nom : GAVA Frédéric

Fonction : MCF

Institution / Laboratoire : LACL, Laboratoire d'Algorithmique, Complexité et Logique (EA 4213),

Adresse postale : 61 avenue du Général de Gaulle, Université de Paris 12, 94010 Créteil Cedex

Téléphone : 01 45 17 66 00

Fax : : 01 45 17 66 01

Email : [gava@univ-paris12.fr](mailto:gava@univ-paris12.fr)

### **d) Partenaire francilien du projet**

Laboratoire d'Algorithmique, Complexité et Logique (EA 4213)

Directeur : [Gaétan Hains](#)

Adresse complète : LACL, Université Paris XII-Val de Marne,

UFR de Sciences et Technologie, Bâtiment P2, 2<sup>ème</sup> étage

61 avenue du Général de Gaulle, 94010 Créteil

Téléphone : 01 45 17 66 00

Télécopie : 01 45 17 66 01

Site Internet : <http://www.univ-paris12.fr/lac1/>

### **e) Partenaires non franciliens du projet**

Laboratoire d'Informatique Fondamentale d'Orléans (EA 4022)

Directeur : [Christel Vrain](#)

Adresse complète : LIFO, Université d'Orléans

Batiment IIIA, Rue Léonard de Vinci

B.P. 6759, F-45067 ORLEANS Cedex 2

Téléphone : 02 38 41 72 89

Télécopie : 02 38 41 71 37

Site Internet : <http://www.univ-orleans.fr/lifo/>

### III Participants (permanents) au projet

NOM	Prénom	Laboratoire de rattachement	Emploi actuel	Date de nomination dans le poste	Date de naissance
<a href="#">BAMHA</a>	Mostafa	LIFO	MCf	01/09/2003	09/05/1974
<a href="#">LOULERGUE</a>	Frédéric	LIFO	Pr	01/09/2005	18/05/1973
<a href="#">POMMEREAU</a>	Franck	LACL	MCf	01/09/2004	09/06/1974
<a href="#">GAVA</a>	Frédéric	LACL	MCf	01/09/2006	10/10/1979

#### Franck Pommereau :

*Thème de recherche* : Mes travaux de recherche portent sur des *modèles structurés* explicitant la concurrence et qui s'appliquent au domaine du temps-réel. J'ai notamment développé une représentation du temps et de la préemption dans le cadre d'une *algèbre de réseaux de Petri colorés* (algèbre des M-Nets [13]). Ces travaux ont été appliqués à la vérification des systèmes temps-réel et à la sémantique des langages de programmation parallèles ou concurrents. Plus récemment, mes travaux ont évolué vers les problématiques d'exécution et de *vérification efficaces* d'un sous-ensemble ce formalisme.

*Equipe* : Systèmes communicants ;

*Page perso et publications* : <http://www.univ-paris12.fr/lacl/pommereau/>

#### Frédéric Loulergue :

*Thème de recherche* : Les activités menées au LIFO autour de la *programmation de haut-niveau* pour les *architectures parallèles*, distribuées et globalisées suivent deux grands axes. Le premier traite de la conception de langages et bibliothèques de haut-niveau [15], basés sur des *sémantiques formelles*, et le développement d'applications avec ces langages et bibliothèques. Le second traite de la *vérification de programmes parallèles* de haut-niveau : preuves de programmes parallèles, correction des compilateurs et des machines virtuelles associées à nos langages, outils d'analyse statique.

*Equipe* : Parallélisme Réalité virtuelle et Vérification de système (PRV), co-responsable

*Page perso et publications* : <http://www.univ-orleans.fr/lifo/Members/loulergu/>

#### Frédéric Gava :

*Thème de recherche* : Mon travail au LACL consiste à fournir des outils (sémantiques, algorithmes, implantations correctes) permettant la *portabilité*, la *sûreté* et la *certification* [10] des composantes d'une bibliothèque dédiée à la programmation parallèle de haut-niveau (appelé BSMLib) ainsi que de l'étendre dans un environnement de *méta-computing* [11]. Cette bibliothèque permet de programmer de manière fonctionnelle des algorithmes BSP (*Bulk Synchronous Parallelism*), un modèle de parallélisme permettant la *prévision des performances* [15] sur une grande variété d'architectures. Plus récemment, mes travaux ont évolué vers les problématiques de *vérification parallèle* de l'algèbre M-Net.

*Equipe* : Systèmes communicants

*Page perso et publications* : <http://www.univ-paris12.fr/lacl/gava/>

#### Mostafa Bamha :

*Thème de recherche* : Mes recherches portent sur la programmation parallèle de haut-niveau : portabilité des performances (modèle BSP), outils de programmation et conception d'algorithmes pour le *re-balancement des données dirigé pour les coûts parallèles BSP* avec notamment des applications aux bases de données [2].

*Equipe* : Réalité virtuelle et Vérification de système (PRV)

*Page perso et publications* : <http://www.univ-orleans.fr/lifo/Members/bamha/>

## IV Description détaillée du projet VEHICULAIRE

### a) Contexte scientifique, défis et enjeux

La modélisation d'un système en donne une représentation symbolique, le *modèle*, dont la sémantique peut être analysée plus ou moins automatiquement. Lorsque le but est la *vérification* d'une propriété du système (*model checking*), l'analyse doit être automatique, et l'immense majorité des sémantiques considérées sont alors des systèmes de transitions représentant les états possibles du modèle ainsi que les transitions le faisant passer d'un état à un autre. On appelle cela *l'espace des états* du modèle. Plusieurs caractéristiques des systèmes considérés amènent au problème connu sous le nom d'*explosion* de l'espace des états [9] : les causes en sont généralement une *forte combinatoire* des états de sous-systèmes relativement indépendants ou la présence dans le modèle de *données sur de grands ensembles*.

Afin de contrer l'explosion de l'espace des états, plusieurs grandes familles de techniques existent : *réduction du nombre d'états représentés* en utilisant des classes d'états, *abstraction* [1] *symbolique* [3] des données, *représentation compacte de l'espace des états*, *vérification modulaire* [16] de sous-systèmes ou de sous-espaces, *parallélisation* de la vérification [14] *etc.*

Sur le plan international, la *communauté de la vérification* est très active sur la plupart des techniques d'accélération mais on peut constater un certain *morcellement de l'effort* dans ce contexte assez concurrentiel. Des tentatives de rapprochement entre différentes techniques existent mais restent des cas isolés.

### b) Objectifs, concepts, outils et intérêt pour la Région Île-de-France

Le projet s'intéresse à la vérification de systèmes modélisés à base de *réseaux de Petri de haut niveau*, plus particulièrement une algèbre de réseaux structurée appelée *M-Net* (ou un sous-ensemble bien choisi pour une vérification efficace mais néanmoins suffisamment expressive pour modéliser nombre de systèmes) ainsi que de la vérification efficace de propriétés logiques sur ce graphe (certaines logiques comme LTL [21] ou autres sont souvent très expressives mais un sous-ensemble expressif et efficacement vérifiable sera nécessaire à définir [20]). Ce formalisme a été choisi pour différents critères :

- 1) Il relève du domaine de compétence des membres de l'équipe « Systèmes communicants » du LACL ;
- 2) Les réseaux de Petri commencent à être utilisés de manière *pratique* par l'industrie et des communautés différentes que celles de la vérification des systèmes ;
- 3) Il présente des caractéristiques autorisant d'intéressantes techniques d'accélération : les réseaux de Petri utilisés sont *composables*, autorisant ainsi une *modélisation structurée* ;
- 4) L'algèbre est de *haut-niveau* et permet donc une représentation plus compacte des problèmes à modéliser et une *explosion des états moindres* du fait de cette description plus symbolique ;
- 5) Les techniques d'accélération utilisées pour ce modèle (basées sur la concurrence des actions) sont souvent complémentaires de celles développées pour d'autres (abstraction, réduction, *etc.*).

Le projet a pour but principal la conception d'un logiciel calcul haute performance<sup>3</sup> qui comportera un *algorithme parallèle BSP de construction du graphe des états* d'une classe dédiée des M-Nets ainsi que la *vérification de propriétés logiques* dans ce graphe. Nous nous baserons sur la *structuration* d'une telle classe et du modèle BSP (*via* son modèle de performance) pour réduire les communications de l'algorithme et permettre un *re-balancement optimal* des données et des calculs pour la construction de ce graphe. En effet, la structuration de ces réseaux nous permettra de connaître de manière *dynamique* les

---

<sup>3</sup> Nous rappelons que le calcul haute performance, c'est-à-dire le calcul massivement parallèle, est considéré, comme la solution à la fin des possibilités d'accélération (loi de Moore) des processeurs séquentiels. De nombreux responsables scientifiques (comme celui du CEA dans le numéro spécial « calcul haute performance » de mai de la revue « La recherche ») ou industriels (Intel) résumant parfaitement la situation : **parallélisez ou périssez.**

tailles maximales des sous-graphes des sous-réseaux et ainsi de re-balancer au mieux les calculs et les communications (afin d'optimiser les performances de l'outil). La vérification des propriétés logiques en sera alors tout autant simplifiée et accélérée. L'utilisation du modèle de coûts BSP nous donnera aussi une *prédiction des performances* du *model-checker* et ce de manière très portable : il sera alors possible à l'utilisateur de prédire le temps d'exécution de l'algorithme sur une machine parallèle et donc de choisir au mieux la machine qu'il souhaite utiliser (ce travail, dans le futur, pourra être fait de manière automatique sur une *grille de calcul* via l'utilisation très aisée d'agents mobiles). Ceci diffèrera des précédents résultats [8] sur des algorithmes distribués de *model-checking* où les performances sont imprédictibles et où les calculs (ainsi que les communications) sont mal balancés : une distribution randomisée des données sur les processeurs est en général utilisée et se révèle trop souvent inefficace à cause du mauvais balancement des calculs.

Pour implanter cet algorithme nous utiliserons le langage de *programmation de haut-niveau* (*fonctionnelle, polymorphe, modulaire, etc.*) *Objective Caml* (OCaml<sup>4</sup> [23] ) et son extension BSP via une bibliothèque développée conjointement au LACL et au LIFO (la *BSMLlib*). L'utilisation des *modules* et du *polymorphisme* de OCaml nous permettra une exécution *native* (compilation en code machine) des réseaux sans la nécessité d'un interpréteur de réseau : celui-ci sera en fait un foncteur dépendant d'un module décrivant le type de jeton présent dans les places et d'un autre module décrivant les calculs faits par les transitions entre les places du réseau. Cette programmation de haut-niveau est donc idéale pour une *indépendance* de l'algorithme aux types de données (symboliques ou non) représentées par les jetons. Il en résultera donc une autonomie vis-à-vis des autres techniques de réduction des graphes des états.

Ce projet est innovant à plus d'un titre, en particulier sur ces deux points principaux :

- 1) A notre connaissance il n'existe pas de techniques de *modularisation* et de *parallélisation* (*structurée*) basées sur la bonne structuration d'un modèle de haut-niveau ;
- 2) Contrairement à la tradition de la communauté de vérification qui tend à privilégier les langages de bas-niveau (C ou C++), nous utiliserons un *langage de haut-niveau* dont les bénéfices attendus seront une meilleure *portabilité*, une maintenance plus aisée, une meilleure *évolutivité* et la capacité d'exécuter de manière native les transitions d'un réseau de Petri de haut-niveau (les transitions pourront être directement écrites dans du code OCaml).

Pour commencer à vérifier expérimentalement notre travail, nous appliquerons notre implantation à des *protocoles cryptographiques* qui ont déjà été modélisés par des membres du LACL [4] avec une classe particulière des M-Nets (la sécurité et la sûreté des systèmes étant les thèmes fédérateurs du LACL). Pour continuer cette expérimentation, il sera envisagé de modéliser avec notre sous-ensemble des M-Nets, puis vérifier avec notre outil de *model-checking* :

- les systèmes étudiés au LIFO par l'équipe PRV (notamment les *protocoles* pour le *commerce électronique*)
- des systèmes physiques ou biologiques que nous pourrions rencontrer
- des programmes hautes performances (C, Ada ou Fortran avec une bibliothèque de calculs parallèles telle que MPI) ou des programmes parallèles MPI avec squelettes algorithmiques qui ont été développé aux LIP (Anne Benoît) de Lyon ainsi qu'à Edinburgh (Murray Cole).

Les vérifications des propriétés s'effectueront pour commencer sur les différentes grappes de PCs du LACL et du LIFO. Et si ces tests sont concluants, d'autres, plus importants, seront réalisés dans des centres de calcul (avec des machines parallèles plus conséquentes pour ces cas plus longs à vérifier).

La thèse qui sera rédigée dans le cadre de ce projet, se portera en premier temps aux techniques de parallélisation et d'équilibrage de charge. Dans un second temps, une implantation efficace sera développée (ainsi que son expérimentation). Le doctorant devra donc profiter des compétences des différents membre du projet : équilibrage de charge et vérification logique de réseaux de Petri au LIFO

---

<sup>4</sup> <http://www.pauillac.inria.fr/caml>

d'Orléans et algorithmes parallèles, implantation parallèle de haut-niveau et réseaux de Petri de haut-niveau du LACL à Paris 12.

L'intérêt pour l'Île-de-France est double : une meilleure collaboration d'un de ses laboratoires d'informatique avec celui d'Orléans ainsi que la conception d'un logiciel complexe (calcul haute performance) pour la vérification automatique de logiciels (systèmes) complexes (plus le système est complexe, plus une modélisation de haut-niveau est nécessaire).

### **c) Positionnement par rapport à l'état de l'art et à la concurrence nationale et internationale**

Concernant la parallélisation, il semble que peu ou pas de spécialistes du calcul massivement parallèle se soient penchés sur la question. La seule collaboration qui existe, à notre connaissance, entre la communauté de la vérification et celle du calcul haute-performance est l'équipe VASY de l'INRIA, mais leurs algorithmes sont *distribués* [8] (et donc n'utilisant pas un parallélisme structuré qui serait plus efficace) et pour des modèles de *bas-niveau non structurés* (donc avec une représentation peu compacte et *non modulaire* des problèmes) [12] .

On trouve aussi le système *quasar*<sup>5</sup> développé par le laboratoire CEDRIC du CNAM. Celui-ci est aussi basé sur les réseaux de Petri mais pas sur un sous-ensemble structuré. Des techniques de parallélisations empiriques ont été étudiées et intégrées dans ce système mais sans se baser ni sur un modèle de parallélisme structuré (qu'est le modèle BSP et donc plus efficace) ni sur des algorithmes avec modèle de coûts permettant un équilibrage de charge (prouvé par les coûts) des calculs du graphe des états ainsi que des propriétés à vérifier.

On commence à trouver dans la littérature[18] [19] des premiers travaux sur la structuration des modèles à étudier afin d'optimiser la vérification parallèle à l'aide de ces nouvelles propriétés. Mais cette approche n'est qu'à ces balbutiements et les auteurs n'utilisent pas encore, ni un langage de programmation parallèle de haut niveau afin de modulariser leurs implantations (et de facilement l'optimiser) qui serait le plus à même pour la vérification d'une algèbre basée sur des réseaux de Petri de haut-niveau, ni un modèle d'exécution parallèle de haut-niveau permettant l'optimisation (prouvée par les coûts) des algorithmes.

### **d) Programme des travaux**

Dans une première phase, une étude approfondie de l'existant et une mise en commun des connaissances semblent indispensables. Cela permettra de plus aux membres du projet (notamment le doctorant) de se forger des références communes alors qu'ils sont issus de cultures différentes. Cette phase devrait être assez courte, car la plupart des membres de l'équipe sont déjà largement familiarisés avec les problématiques d'accélération de la vérification ou du parallélisme de haut-niveau, même si leurs savoirs sont issus de différentes communautés (d'où le nom du projet). Une importante bibliographie sur le sujet de la vérification parallèle ou distribuée a par ailleurs été déjà rassemblée.

Par la suite, les travaux porteront sur la recherche du bon sous-ensemble des M-Nets à utiliser, de la bonne logique (ainsi que les propriétés à vérifier) et des algorithmes parallèles à proprement parler. Cela nous amènera probablement à hybrider des formalismes (différentes variantes des réseaux de Petri) et des techniques (équilibrage des charges, distribution des données, prédiction des performances, *etc.*) afin d'obtenir les meilleurs algorithmes possibles. A mesure de la progression du projet et surtout de la conception des algorithmes et des implantations des prototypes avec la BSMLlib, des études de cas pourront être proposées à des stagiaires des Masters recherche ou professionnels (au LACL ou au LIFO). Nous pensons proposer au moins deux *sujets de Master recherche* dès l'année prochaine. Plus tard, l'implantation d'un outil intégrant les différentes approches développées et son application à un problème d'envergure pourront faire l'objet d'un stage (ou plusieurs) de *Master professionnel* avec le(s) partenaire(s) industriel(s) habituels d'un des membres du projet.

Enfin, tout au long du projet, une veille technologique sera effectuée afin de rester en contact avec les dernières innovations en matière d'accélération de la vérification. Cela semble la condition *sine qua non* pour formuler des propositions pertinentes au regard de l'état de l'art en la matière.

---

<sup>5</sup> <http://quasar.cnam.fr/>

## e) Echancier (planning prévisionnel : date début, date fin et étapes clés)

Le planning prévisionnel correspond aux 3 années de thèse du doctorant ainsi que d'une quatrième année pour mettre en pratique, communiquer et distribuer les résultats par les membres permanents du projet. Il est le suivant :

01/09/2007 → 01/09/2008 : bibliographie<sup>6</sup> sur les algorithmes de génération d'un graphe des états; bibliographie sur les logiques à employer pour le *model checking* ; sélection du bon sous-modèle à utiliser pour le *model-checking* ; sélection des bonnes propriétés logiques à vérifier (c'est donc une approche dédiée, pour maximiser l'efficacité de la vérification) ; premières études de cas (systèmes biologiques ou de sécurité, programmes MPI) afin de vérifier les bons choix du projet.  
**Etape clé** : étude bibliographique, du modèle et du bon sous ensemble logique à utiliser.

01/09/2008 → 01/09/2009 : définition d'un premier algorithme parallèle pour la génération du graphe des états ainsi que de la vérification des propriétés logiques des systèmes ; première intégration des deux types d'algorithmes dans un seul et même algorithme ;  
**Etape clé** : conception d'un algorithme parallèle de génération du graphe des états et d'un autre algorithme pour la vérification de propriétés logiques sur ce graphe ; premières implantations et tests dans le cadre d'un stage de Master 2.

01/09/2009 → 01/12/2010 : optimisation et implantation des algorithmes ; intégration des deux algorithmes dans un seul et même outil de vérification ; premiers tests de cas réels.  
**Etape clé** : amélioration des deux algorithmes et leur intégration dans un seul et même logiciel ; tests de cas dans le cadre d'un stage de Master 2 professionnel.

01/12/2010 → 01/07/2011 : tests de cas industriels et scientifiques réels ; mise en pratique par des membres extérieurs au projet notamment avec des squelettes algorithmiques<sup>7</sup>.  
**Etape clé** : fin du projet par la diffusion du logiciel et sa mise en pratique dans la communauté scientifique et industrielle

## f) Partage des tâches entre les partenaires

Nous présentons le travail attendu par les membres du projet (avec l'aide du futur doctorant) :

- *Franck Pommereau* sélectionnera et adaptera les éléments intéressants dans les M-Nets ;
- *Mostafa Bamha* aidera à la conception et l'adaptation des techniques de re-balancement (rééquilibrage) des calculs basés sur des algorithmes BSP génériques qu'il a déjà conçus ;
- *Frédéric Gava et Frédéric Loulergue* participeront à l'implantation efficace de l'algorithme avec la BSMLlib ;
- *Jean-Michel Couvreur*, Professeur au LIFO (et associé du projet) apportera sa haute expertise dans le domaine du *model checking* des réseaux de Petri pour les besoins des systèmes d'informations et de l'industrie.
- *Elisabeth Pelz*, Professeur au LACL (et associé du projet) apportera sa haute expertise dans le domaine des réseaux de Petri de haut niveau (comme les M-net) et la vérification.
- *Le doctorant* définira et testera les algorithmes parallèles nécessaires pour le logiciel

## g) Critères proposés pour juger du succès du projet

Le critère principal d'un tel projet (et de sa thèse) est une bonne accélération des calculs du logiciel de haute performance par rapport à une version séquentielle des calculs. En effet, dans la communauté calcul haute performance (parallélisme) on mesure les gains d'un algorithme parallèle par rapport à son pendant séquentiel : l'idée sous jacente est que plus le problème à traiter est important (en taille), plus le

<sup>6</sup> Notons qu'un premier état de l'art a déjà été effectué par les membres du projet afin d'estimer les possibilités et la faisabilité du projet.

<sup>7</sup> Un squelette algorithmique est une « fonction » qui peut être implantée en parallèle : chaque type de parallélisme (diviser-pour-régner, *pipeline* etc.) se voit doté d'un squelette ; cela permet de paralléliser facilement et de manière plus sûre des programmes, en utilisant des squelettes appropriés et qui « représentent » bien le parallélisme intrinsèque du problème

nombre de processeurs nécessaires doit être grand pour conserver un temps de calcul raisonnable. Une accélération linéaire au nombre de processeurs serait l'idéal.

Le second critère le plus pertinent serait une accélération des temps de calcul par rapport aux précédentes approches proposées pour une même modélisation d'un système. Cela permettrait de comparer notre méthode à celles de la communauté de la vérification distribuée.

Un dernier critère mais plus difficile à juger à si court terme<sup>8</sup> est l'emploi d'un tel logiciel dans la communauté scientifique et dans le tissu industriel. C'est l'objectif principal de la dernière étape du projet mais aussi le plus difficile à mettre en œuvre. La modélisation d'au moins un système industriel et d'un système scientifique (biologique ou physique) serait déjà un gage de succès du projet.

## **h) Livrable**

Le livrable sera un logiciel complet de calcul haute performance pour le *model-checking* c'est-à-dire la vérification automatique de propriétés logiques d'un système à partir de sa modélisation dans un formalisme. Le formalisme employé sera celui d'un sous-ensemble bien construit (pour le calcul haute performance) de l'algèbre des M-nets (basée sur des réseaux de Petri de haut-niveau). La vérification sera effectuée à partir d'un sous ensemble bien construit (pour le calcul haut performance) d'une logique appropriée à la vérification des systèmes.

Le doctorant ainsi formé sera aussi un spécialiste dans deux domaine de compétence : le calcul haute performance et la vérification des systèmes. Les futurs bénéficiaires de cette expertise pourront être nécessaire pour l'industrie de pointe et pour les laboratoires de recherche.

## **i) Partenariat et travaux antérieurs des proposant**

### **Contrats précédents ou en cours proches du sujet traité :**

- Certains membre du projet ont fait partie du projet CARAML<sup>9</sup> financé par l'ACI GRID. Ce projet avait pour but le développement de bibliothèques pour le calcul haute-performance et globalisé autour du langage OCaml.
- Jean-Michel Couvreur a participé au projet RNTL MORSE<sup>10</sup> (Méthodes et Outils pour la Vérification Formelle de Systèmes Intéropérables Embarqués critiques) qui consistait à fournir une solution pour le développement d'applications industrielles certifiables dans le secteur de l'avionique embarquée et plus particulièrement pour les drones.

**Expériences de collaborations antérieures entre les proposant :** Frédéric Louergue a été l'encadrant de thèse de Frédéric Gava et ils ont tous les deux travaillé dans le projet PROPAC<sup>11</sup> financé par l'ACI « Jeunes chercheurs ». Jean-Michel Couvreur (associé du projet) et Frédéric Louergue co-dirigent l'équipe PRV du LIFO. Frédéric Gava et Franck Pommereau appartiennent à la même équipe du LACL et travaillent ensemble depuis quelques mois sur cette nouvelle thématique. Elisabeth Pelz (associé du projet) dirige l'équipe « système communicants » du LACL.

### **Liste des thèses encadrées par les proposant qui ont été soutenues au cours des cinq dernières années et devenir des docteurs ainsi formés :**

Frédéric Louergue a encadré Frédéric Gava de 2002 à 2005 (prix de thèse de la fondation d'entreprise EADS). Frédéric Gava est maintenant MCF à l'université de Paris 12.

Thèses en cours :

- Louis [GESBERT](#) (directeur : Frédéric Louergue)
- Radia [BENHEDDI](#) (directeur : Frédéric Louergue)
- [AL HAJJ HASSAN Mohamad](#) (directeur Mostafa Bamha)

<sup>8</sup> Il est de notoriété publique qu'il est souvent très long de parvenir à distribuer les fruits de la recherche en dehors de sa propre communauté

<sup>9</sup> <http://caraml.free.fr/>

<sup>10</sup> <http://morse.lip6.fr/>

<sup>11</sup> <http://wwwpropac.free.fr/>

## V Références

- [1] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. *Discrete abstractions of hybrid systems*. Proceedings of the IEEE, 2000.
- [2] M. Bamha and G. Hains. An Efficient equi-semi-join Algorithm for Distributed Architectures. . In V. S. Sunderam, G. Dick van Albada, P. M. A. Sloot, and J. Dongarra, editors, *International Conference on Computational Science (ICCS 2005), Part II*, number 3515 in LNCS, Springer-Verlag, 2005.
- [3] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. PhD Thesis, volume 189, Collection des Publications de la Faculté des Sciences Appliquées de l'Université de Liège, 1999.
- [4] R. Bouroulet, H. Klaudel, and E. Pelz. *A semantics of security protocol language using a class of composable high-level petri nets*. In M. Kishinevsky and Ph. Darondeau, editors, *Application of Concurrency to System Design*, volume 4th ACSD, pages 99-108. IIEE, 2004.
- [5] Jean-Michel Couvreur, Sébastien Grivet, Denis Poitrenaud: *Unfolding of Products of Symmetrical Petri Nets*. ICATPN 2001: 121-143
- [6] Jean-Michel Couvreur, Emmanuelle Encrenaz, Emmanuel Paviot-Adet, Denis Poitrenaud, Pierre-André Wacrenier. *Data Decision Diagrams for Petri Net Analysis*. ICATPN 2002: 101-120
- [7] Jan Duennweber, Sergei Gorlatch, Anne Benoit and Murray Cole *Integrating MPI-Skeletons with Web Service* in Proceedings of *ParCo 2005*.
- [8] H. Garavel, R. Mateescu, C. Joubert and al. *DISTRIBUTOR and BCG\_MERGE: Tools for Distributed Explicit State Space Generation*, March 2006.
- [9] G. Gardey, O. H. Roux and O. F. Roux. *State space computation and analysis of time Petri nets*. Theory and Practice of Logic Programming (TPLP). Special Issue on Specification Analysis and Verification of Reactive Systems, 2005.
- [10] F. Gava. Formal Proofs of Functional BSP Programs. *Parallel Processing Letters*, 13(3):365-376, 2003.
- [11] F. Gava and F. Loulergue. A Functional Language for Departmental Metacomputing. *Parallel Processing Letters*, 15(3):289-304, 2005.
- [12] Christophe Joubert and Radu Mateescu Distributed On-the-Fly Model Checking and Test Case Generation, April 2006, 24 pages.
- [13] Hanna Klaudel and Franck Pommereau. *M-nets, a survey*. Acta Informatica, À paraître.
- [14] L. Kristensen and L. Petrucci. *An approach to distributed state space exploration for coloured Petri nets*. ICATPN'2004, LNCS 3099. Springer, 2004
- [15] F. Loulergue, F. Gava, and D. Billiet. *Bulk Synchronous Parallel ML: Modular Implementation and Performance Prediction*. In V. S. Sunderam, G. Dick van Albada, P. M. A. Sloot, and J. Dongarra, editors, *International Conference on Computational Science (ICCS 2005), Part II*, number 3515 in LNCS, pages 1046-1054. Springer-Verlag, 2005.
- [16] M. Mäkelä. *Modular reachability analyzer for high-level Petri nets*. 5th Workshop on Discrete Event Systems. Kluwer Academic Publishers, 2000.
- [17] J. L. Traeff and J. Worrigen. Verifying collective MPI calls. In Proceedings of the 11th EuroPVM/MPI conference, LNCS. Springer, 2004.
- [18] D.Petcu, Parallel explicit-state reachability analysis and state space construction, Proceedings of Second International Symposium on Parallel and Distributed Computing, ISPDC 2003, 13-14 October 2003, Ljubljana, IEEE Computer Society Press, Los Alamitos
- [19] Jean-Michel Couvreur, Yann Thierry-Mieg: Hierarchical Decision Diagrams to Exploit Model Structure. FORTE 2005: 443-457
- [20] J.-M. Couvreur and D. Poitrenaud. Dépliage pour la vérification de propriétés temporelles. In *Vérification et mise en œuvre des réseaux de Petri - Tome 2*, chapter 3, pages 127-161. Hermès, 2003.
- [21] Automatic verification of finite state concurrent systems using temporal logic, E.M. Clarke, E.A. Emerson, and A.P. Sistla, ACM Trans. on Programming Languages and Systems, 8(2), pp. 244–263, 1986
- [22] Model Checking, Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled, MIT Press, 1999
- [23] D. Rémy. Using, Understanding, and Unravelling the OCaml Language. In G. Barthe, P. Dyjber, L. Pinto, and J. Saraiva, editors, *Applied Semantics*, number 2395 in LNCS, pages 413–536. Springer, 2002.

## *Curriculum Vitae*

**Nom:** GAVA  
**Prénom:** FREDERIC  
**Fonction:** Maître de Conférences - Université de Paris 12  
**Unité de recherche:** LACL, Laboratoire d'Algorithmique, Complexité et Logique, Université de Paris 12  
61 avenue du Général de Gaulle, CMC-P2, 94010 Créteil cedex  
**Equipe de recherche:** Systèmes communicants  
**Email:** gava@univ-paris12.fr  
**Web:** <http://www.univ-paris12.fr/lacl/gava/>

### **THÈMES DE RECHERCHE**

- approches de haut-niveau pour la programmation parallèle, en particulier langages fonctionnels,
- modularité, certification, prévision de performances de la programmation parallèle
- vérification parallèle d'algèbre de processus de réseaux de Petri de haut-niveau

### **ACTIVITÉS DE RECHERCHE**

Mon travail au LACL consiste à fournir des outils (sémantiques, algorithmes, implantations correctes) permettant la portabilité, la sûreté et la certification des composantes d'une bibliothèque dédiée à la programmation parallèle de haut-niveau (appelé BSMLlib) ainsi que de l'étendre dans un environnement de méta-computing. Cette bibliothèque permet de programmer de manière fonctionnelle des algorithmes BSP (Bulk Synchronous Parallelism), un modèle de parallélisme permettant la prévision des performances sur une grande variété d'architectures. Plus récemment, mes travaux ont évolué vers les problématiques de vérification parallèle de l'algèbre M-Net.

**Prix de thèse de la fondation d'entreprise EADS**

### **PRINCIPALES RESPONSABILITES SCIENTIFIQUE ET ADMINISTRATIVES :**

- Co-responsable du séminaire hebdomadaire du LACL
- Co-responsable de la communication du LACL
- Coordinateur de l'administration des différentes grappes de PC du LACL
- Membres des comités de programmes des workshop PAPP 2007 et WLPP 2007
- Responsable de l'enseignement de l'informatique à l'UFR de Droit de Paris 12

### **PUBLICATIONS RECENTES :**

- [1] F. Gava Implementation of the Parallel Superposition in Bulk-Synchronous Parallel ML. International Conference on Computational Science (ICCS 2007), Part I, number 4487 in LNCS, pages 611-620. 2007. Springer-Verlag.
- [2] F. Gava. Une bibliothèque certifiée de programmes fonctionnels BSP, Techniques et Sciences Informatiques, 25(10):1261-1280. 2006
- [3] L. Gesbert, F. Gava and F. Loulergue. Bulk Synchronous Parallel ML with Exceptions. In P. Kacsuk, T. Fahringer and Z. Nemeth, editors, Distributed and Parallel Systems (DAPSYS 2006), LNCS. Springer-Verlag, to appear, 2006.
- [4] F. Loulergue, R. Benheddi, F. Gava, and D. Louis-Regis. Bulk Synchronous Parallel ML: Semantics and Implementation of the Parallel Juxtaposition. In International Computer Science Symposium in Russia (CSR 2006), number 3967 in LNCS, pages 475-486. Springer-Verlag, 2006.
- [5] F. Gava. External Memory in Bulk Synchronous Parallel ML. Scalable Computing: Practice and Experience, 6(4):43-70, 2005.
- [6] F. Gava and F. Loulergue. A Functional Language for Departmental Metacomputing. Parallel Processing Letters, 15(3):289-304, 2005.
- [7] F. Gava and F. Loulergue. A Static Analysis for Bulk Synchronous Parallel ML to Avoid Parallel Nesting. Future Generation Computer Systems, 21(5):665-671, 2005.
- [8] F. Loulergue, F. Gava, and D. Billiet. Bulk Synchronous Parallel ML: Modular Implementation and Performance Prediction. In V. S. Sunderam, G. Dick van Albada, P. M. A. Sloot, and J. Dongarra, editors, International Conference on Computational Science (ICCS 2005), Part II, number 3515 in LNCS, pages 1046-1054. Springer-Verlag, 2005.
- [9] F. Gava. A modular implementation of data structures in Bulk-Synchronous Parallel ML Third Workshop on High-Level Parallel Programming and Applications (HLPP'2005).

## Curriculum Vitae

**Nom:** LOULERGUE  
**Prénom:** Frédéric  
**Fonction:** Professeur - Université d'Orléans  
**Unité de recherche:** LIFO - Laboratoire d'Informatique Fondamentale d'Orléans,  
rue Léonard de Vinci, B. P. 6759, Orléans Cedex 2.  
**Equipe de recherche:** Parallélisme Réalité virtuelle et Vérification de systèmes (PRV).  
**Email:** frederic.loulergue@univ-orleans.fr  
**Web:** <http://www.univ-orleans.fr/lifo/Members/loulergu/>

### THÈMES DE RECHERCHE

- approches de haut-niveau pour la programmation parallèle, en particulier langages fonctionnels,
- modularité, certification, prévision de performances de la programmation parallèle

### ACTIVITÉS DE RECHERCHE :

Les activités menées au LIFO autour de la *programmation de haut-niveau* pour les *architectures parallèles*, distribuées et globalisées suivent deux grands axes. Le premier traite de la conception de langages et bibliothèques de haut-niveau, basés sur des *sémantiques formelles*, et le développement d'applications avec ces langages et bibliothèques. Le second traite de la *vérification de programmes parallèles* de haut-niveau : preuves de programmes parallèles, correction des compilateurs et des machines virtuelles associées à nos langages, outils d'analyse statique.

### PRINCIPALES RESPONSABILITÉS SCIENTIFIQUE ET ADMINISTRATIVES :

**Projets passés principaux :** ACI GRID, projet Caraml (2002-2004), coordinateur local (Paris 12) ; ACI Jeunes Chercheurs, projet "Programmation Parallèle Certifiée" (2004-2007), coordinateur

**Invitations :** Université de La Laguna, Espagne (1 semaine, 2004), Université de Tokyo (1 semaine, 2006), Université de Tokyo (JPSP Short Term Invitation Fellowship, 2 mois, 2007), Institut de la Francophonie pour l'Informatique, Hanoï (2 semaines, 2007)

**Président de comités de programmes :** (co)-président du comité de programme de la série de workshops Practical Aspects of High-Level Parallel Programming (2004, 2005, 2006, 2007, affiliés à l'International Conference on Computational Science, LNCS) et de la série de workshop High-Level Parallel Programming and Applications (Paris 2003, Coventry 2005, Tokyo 2007)

**Editeur invité :** (co)-éditeur invité de numéros spéciaux de Parallel Processing Letters (vol. 13 n°3 et en préparation), Scalable Computing: Practice and Experience (vol. 6 n°4, vol. 7 n°3 et en préparation), Computer Languages, Systems and Structures (vol. 33 n°3-4)

**Encadrement de thèses :** Louis Gesbert (en cours), Radia BENHEDDI (en cours), Frédéric Gava (MCF)

Co-responsable du projet "Parallélisme Réalité virtuelle et Vérification de systèmes" (PRV)

Directeur des études du Master CCI (Compétence Complémentaire en Informatique) de l'Université d'Orléans

### PUBLICATIONS RECENTES :

- [1] Bulk Synchronous Parallel ML (<http://bsml.free.fr>)
- [2] F. Gava and F. Loulergue. A Functional Language for Departmental Metacomputing. Parallel Processing Letters, 15(3):289-304, 2005.
- [3] F. Gava and F. Loulergue. A Static Analysis for Bulk Synchronous Parallel ML to Avoid Parallel Nesting. Future Generation Computer Systems, 21(5):665-671, 2005.
- [4] L. Gesbert, F. Gava, F. Loulergue, and F. Dabrowski. Bulk Synchronous Parallel ML with Exceptions. In Peter Kacsuk et al., editors, Distributed and Parallel Systems (DAPSYS 2006), pages 33-42. Springer, 2006.
- [5] F. Loulergue, R. Benheddi, F. Gava, and D. Louis-Regis. Bulk Synchronous Parallel ML: Semantics and Implementation of the Parallel Juxtaposition. In International Computer Science Symposium in Russia (CSR 2006), volume 3967 of LNCS, pages 475-486. Springer, 2006.
- [6] F. Loulergue. A Calculus of Functional BSP Programs with Projection. In International Parallel & Distributed Processing Symposium, 8th Workshop on Advances in Parallel and Distributed Computational Models. IEEE Computer Society Press, 2006
- [7] F. Loulergue, F. Gava, and D. Billiet. Bulk Synchronous Parallel ML: Modular Implementation and Performance Prediction. In Vaidy S. Sunderam et al., editors, International Conference on Computational Science, Part II, number 3515 in LNCS, pages 1046-1054. Springer, 2005

## *Curriculum Vitae*

**Nom:** POMMEREAU  
**Prénom:** FRANCK  
**Fonction:** Maître de Conférences - Université de Paris 12  
**Unité de recherche:** LACL, Laboratoire d'Algorithmique, Complexité et Logique, Université de Paris 12  
61 avenue du Général de Gaulle, CMC-P2, 94010 Créteil cedex  
**Equipe de recherche:** Systèmes communicants  
**Email:** pommereau@univ-paris12.fr  
**Web:** <http://www.univ-paris12.fr/lacl/pommereau/>

### **THÈMES DE RECHERCHE**

- algèbre de réseaux de Petri colorés
- Modèle de la concurrence et du temps-réel
- Vérification de systèmes
- Sémantique des langages de programmation parallèles ou concurrents

### **ACTIVITÉS DE RECHERCHE**

Mes travaux de recherche portent sur des modèles structurés explicitant la concurrence qui s'appliquent au domaine du temps-réel. J'ai développé au cours de ma thèse et par la suite une représentation du temps et de la préemption dans le cadre d'une algèbre de réseaux de Petri colorés ; ces approches constituent une alternative intéressante aux formalismes de référence du domaine. Ces travaux ont été appliqués à la vérification des systèmes temps-réel et à la sémantique des langages de programmation parallèles ou concurrents. Plus récemment, mes travaux ont évolué vers les problématiques d'exécution et de vérification efficaces des formalismes développés au cours de ma thèse, notamment par l'usage de la programmation massivement parallèle.

### **PRINCIPALES RESPONSABILITÉS SCIENTIFIQUE ET ADMINISTRATIVES :**

- Co-responsable du séminaire hebdomadaire du LACL
- Co-responsable des achats du LACL
- Administrateur de la grappe de PC du LACL
- Responsable du Master 2 ISYDIS de l'ISIAG-IUP MIAGE

### **PUBLICATIONS RECENTES :**

- [1] Petri nets as executable specifications of high-level timed parallel systems. Franck Pommereau SCPE/PDPC 6.4, pages 71-82, SWPS, 2005
- [2] Box Calculus with High-Level Buffers. Cécile Bui Thanh, Hanna Klaudel and Franck Pommereau Proc. of DADS/ASTC'04, SCS, 2004
- [3] Causal Time Calculus. Franck Pommereau. Proc. of FORMATS'03, LNCS 2791, Springer, 2004
- [4] Asynchronous Box Calculus. Raymond Devillers, Hanna Klaudel, Maciej Koutny and Franck Pommereau. Fundamenta Informaticae, 54(1), IOS Press, 2003
- [5] Petri nets with causal time for system verification. Cécile Bui Thanh, Hanna Klaudel and Franck Pommereau. Proc. of MTCS'02, ENTCS 68(5), Elsevier, 2003

## Curriculum Vitae

**Nom:** BAMHA  
**Prénom:** MOSTAFA  
**Fonction:** Maître de Conférences - Université d'Orléans  
**Unité de recherche:** LIFO - Laboratoire d'Informatique Fondamentale d'Orléans,  
rue Léonard de Vinci, B. P. 6759, Orléans Cedex 2.  
**Equipe de recherche:** Parallélisme Réalité virtuelle et Vérification de systèmes (PRV).  
**Email:** Mostafa.Bamha@univ-orleans.fr  
**Web:** <http://www.univ-orleans.fr/lifo/Members/bamha/>

### THÈMES DE RECHERCHE

- Parallélisme: Conception, mise en place et évaluation de performance d'algorithmes parallèles, Parallélisme et équilibrage dynamique de charges, Modèle de coût BSP,
- Bases de données parallèle et/ou distribuées : Parallélisme Intra-transaction, Parallélisme pipeliné,
- Programmation Parallèle Fonctionnelle.

### ACTIVITÉS DE RECHERCHE

L'émergence des applications des bases de données dans les domaines tels que le data warehousing, le data mining et l'aide à la décision qui font généralement appel à de très grands volumes de données rend la parallélisation des algorithmes des jointures nécessaire pour avoir un temps de réponse acceptable. Une accélération linéaire est l'objectif principal des algorithmes parallèles, cependant dans les applications réelles, elle est difficilement atteignable : ceci est dû généralement d'une part aux coûts de communications inhérents aux systèmes multi-processeurs et d'autre part au déséquilibre des charges des différents processeurs.

Mes travaux de recherche sur le domaine des bases de données parallèles ont pour objectif la mise en place d'un prototype de système de gestion de bases de données sur une architecture GRID traitant de manière efficace les problèmes de déséquilibre des données et des charges des différents processeurs tout en optimisant l'allocation des ressources de l'architecture parallèle utilisée. Les travaux menés actuellement au sein de l'équipe PRV, dans le domaine du parallélisme dans les bases de données consiste à

- Généraliser les travaux introduits avec succès sur les jointures pour le traitement des requêtes complexes sur des architectures distribuées où les données sont réparties de manière quelconque sur éventuellement sur plusieurs réseaux. Ces travaux doivent être accompagnés d'un modèle de coût pour l'analyse et d'évaluation des performances des solutions proposées (Coût de communication, de synchronisation, test d'efficacité et d'accélération ...).
- Généraliser les travaux réalisés par l'équipe, sur le parallélisme pipeliné au traitement des requêtes complexes sur des architectures GRID afin de réduire les coûts de communication sur de telles architectures et d'optimiser l'allocation des ressources sur des machines distribuées. Ces travaux pourront être exploités pour l'optimisation et la génération de plans d'exécution efficaces pour le traitement de requêtes complexes. L'objectif du parallélisme pipeliné est de réduire le temps d'exécution des requêtes complexes tout en offrant une grande flexibilité et efficacité dans l'allocation des ressources dans le parallélisme à grande échelle.
- Étendre le résultat des travaux sur l'utilisation des histogrammes pour le traitement des jointures à d'autres applications pour un traitement efficace de tous les facteurs de déséquilibre de données et de charges des différents processeurs.

### PRINCIPALES RESPONSABILITES SCIENTIFIQUE ET ADMINISTRATIVES :

- Encadrement de la thèse de AL HAJJ HASSAN Mohamad sur le parallélisme et équilibrage de charges dans les systèmes de gestion de bases de données sur des architectures distribuées.
- Responsable des enseignements en Informatique au département GEA,
- Membre du jury de validation IUT d'Orléans,
- Membre du jury C2I de l'université d'Orléans,
- Correspondant Apogée de L'IUT d'Orléans.

### PUBLICATIONS RECENTES :

- [1] • *Pipelining a Skew-Insensitive Parallel Join Algorithm*. M. Bamha and M. Exbrayat. "Parallel Processing Letters" journal, Number 3, Volume 13, 2003.
- [2] • *An Optimal Evaluation of "GroupBy-Join" Queries in Distributed Architectures* Mohamad Al Hajj Hassan and Mostafa Bamha. *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST'2007)*.
- [3] • *Parallel Processing Of "Group-By Join" Queries On Shared Nothing Machines* Mohamad Al Hajj Hassan and Mostafa Bamha. *Proceedings of the International Conference on Software Data Technologies (ICSOF'06)*.
- [4] • *An optimal skew-insensitive join and multi-join algorithm for distributed architecture* M. Bamha. *Proceedings of the International Conference on Database and Expert Systems Applications (DEXA'2005)*. LNCS 3588. pages 616-625, Springer-Verlag, 22 - 26 August 2005, Copenhagen, Denmark.
- [5] • *An efficient equi-semi-join algorithm for distributed architectures* M. Bamha and G. Hains. *Proceedings of the International Conference on Computational Science (ICCS'2005)*. LNCS 3515. Springer-Verlag. 22-25 May 2005, Atlanta, USA.