



L'air de rien N° 9

Séminaires CSI de mai et juin

L'aléastriel du Laboratoire de Recherche et de Développement de l'EPITA¹

Numéro 9, Mai 2007

Édito

par *Maxime van Noppen*

Une première pour L' AIR DE RIEN qui, le temps d'un numéro, double de taille afin de vous présenter les séminaires CSI qui vont émailler la vie du labora-

toire et des écoles du pôle plus d'un mois durant. Nous vous invitons tous à venir partager avec nous ces hauts moments² de la vie du LRDE.

Dates à retenir

9 mai 14h00–18h30 — amphi IP11	
Markov – Apprentissage	
14h00	Validation des performances d'un algorithme d'apprentissage — Johan Oudinet
14h30	Génération de texte en langage naturel — Jean-Philippe Garcia-Ballester
Vérification du locuteur	
15h45	De nouveaux outils — Julien Ramakichenin
16h15	Adaptation client — Charles Melin
17h00	Approches sélectives — Geoffroy Querol
17h30	Compensation de canal par Factor Analysis — Charles-Alban Deledalle
18h00	Comparaison entre l'utilisation du noyau linéaire et non linéaire pour les systèmes de vérification du locuteur fondés sur les méthodes SVM — Reda Dehak

23 mai 14h00 – 17h45 — amphi 2	
Vaucanson – Automates finis	
14h00	Un format universel de description d'automates et son utilisation dans Vaucanson — Florian Lesaint
14h30	Boosting Vaucanson - partie 1 — Guillaume Lazzara
15h00	Boosting Vaucanson - partie 2 — Jimmy Ma
15h45	Algorithmes génériques et performants de suppression des transitions spontanées dans Vaucanson — Vivien Delmon
16h15	Transducteurs synchronisés — Guillaume Leroi
Théorie des jeux	
16h45	Méthodes algorithmiques de recherche d'équilibres de Nash — Antoine Leblanc
Parallélisme	
17h15	Conception d'une bibliothèque générique de parallélisation en C++ s'appuyant sur l'existant — Elie Bleton

16 mai 14h00 – 17h15 — amphi 2	
Olena – Traitement d'image	
14h00	Une bibliothèque de conteneurs en C++ via SCOOP — Thomas Moulard
DD – Diagramme de Décision	
14h30	Diagrammes de décision à valeurs supprimées — Samuel Charron
Transformers – Transformation de programme	
15h00	Préprocesseur C/C++ réversible — Quentin Hocquet
15h45	Définition formelle de la Désambiguïsation avec des Grammaires Attribuées — Nicolas Pierron
16h15	Désambiguïsation guidée par la sémantique — Renaud Durlin
16h45	Centaur : Une infrastructure pour la transformation de C++ — Benoît Sigoure

6 juin 15h00 – 18h45 — amphi 2	
(Lisp)	
15h00	Étude de problèmes de performances en C et en Common Lisp — William Caldwell
Olena – Traitement d'image	
15h30	Olena, comme les autres ? — Thierry Géraud
16h00	Génie logiciel au coeur d'Olena — Nicolas Ballas
16h30	Canevas d'algorithmes morphologiques — Ugo Jardonnet
17h15	Étude comparative de Concept C++ et perspective d'intégration dans SCOOP — Alexandre Abraham
Transformers – Transformation de programme	
17h45	SCOOL, orientation objet d'un langage statique — Maxime Van Noppen
18h15	Typage du C++ par grammaires attribuées — Florian Quèze

¹L'air de rien, <http://publis.lrde.epita.fr/LrdeBulletin>.

²Nous vous rappelons que les séminaires des étudiants sont sanctionnés par une note comptant dans leur cursus.

Markov

Validation des performances d'un algorithme d'apprentissage

par *Johan Oudinet* ☉ 9 mai, 14h00, amphi IP11

En apprentissage supervisé, les chercheurs n'ont que peu de données pour évaluer les performances de leurs algorithmes d'apprentissage. Ils doivent donc utiliser des heuristiques de test. Mais ces heuristiques, comme leur nom l'indique, peuvent faire des erreurs. Nous évaluerons deux types d'erreurs que ces heuristiques sont susceptibles de commettre : le test trouve qu'il y a une différence significative entre deux algorithmes alors que ce n'est pas le cas (Type I), ou au contraire, le test ne détecte pas la différence existante entre deux algorithmes d'apprentissage (Type II).

Nous commencerons par exposer les différentes sources de variabilité qui peuvent induire en erreur une méthode de test, puis nous étudierons en détail les performances de différentes heuristiques de test, dans l'objectif de trouver la méthode que devraient utiliser les chercheurs pour valider avec certitude les performances de leurs algorithmes.

Génération de texte en langage naturel

par *Jean-Philippe Garcia-Ballester* ☉ 9 mai, 14h30, amphi IP11

Le domaine du traitement du langage naturel est un domaine à la frontière de la linguistique et de l'informatique, et il existe ainsi deux approches à

ce problème : les approches basées sur des règles (liées aux propriétés du langage, comme la grammaire), donc plutôt orientées linguistique, et des approches probabilistes, basées sur des modèles mathématiques, donc plutôt orientées informatique. Le but de cette présentation est de présenter les méthodes de génération de texte. Bien que toutes les méthodes existantes utilisent à la fois des approches linguistiques et probabilistes, nous ne présenterons que les méthodes majoritairement probabilistes. Nous commencerons par exposer les différentes méthodes, puis nous les comparerons, d'abord suivant les domaines dans lesquels les algorithmes sont adaptés ou non, puis suivant l'occupation mémoire et le temps CPU.

Apprentissage pour la vérification

par *Nicolas Neri* ☉ 9 mai, 15h00, amphi IP11

APMC est un outil distribué de vérification de systèmes probabilistes. Dans APMC, il existe une étape de modélisation de système. Cette étape est longue en temps et très coûteuse en espace mémoire car celle-ci est réalisée à la main. En effet il est possible de devoir vérifier des systèmes ayant plus de 10^{99} états ce qui nécessiterait un temps considérable. En ces termes un apprentissage du système peut être utile pour éviter cette lourde étape. Nous nous intéresserons donc au problème de l'apprentissage pour la vérification en étudiant les différents modèles d'apprentissage et leurs domaines de définition.

Reconnaissance du locuteur

De nouveaux outils pour la vérification du locuteur

par *Julien Ramakichenin* ☉ 9 mai, 15h45, amphi IP11

L'état de l'art des systèmes de vérification du locuteur utilise des mélanges de gaussiennes (GMM). Ces modèles sont utilisés pour représenter le modèle du monde (UBM) à partir duquel sont construits les modèles des locuteurs. Nous avons développé un ensemble d'outils pour la manipulation et l'exploitation de ces types de modèles.

Après une introduction à la vérification du locuteur, nous présenterons ces outils et les améliorations qu'ils apportent.

Adaptation client pour la vérification du locuteur

par *Charles Melin* ☉ 9 mai, 16h15, amphi IP11

La vérification du locuteur nécessite l'élaboration de modèles représentatifs des locuteurs. Pour cela, il faut en premier lieu construire un modèle du monde, puis, pour chaque locuteur, créer un modèle particulier. Le manque de données pour le locuteur et la façon de sélectionner les plus représentatives d'entre elles font partie des axes majeurs pour l'amélioration des performances de ces systèmes.

Nous présenterons les techniques qui seront utilisées dans le cadre de notre nouvel outil pour répondre à cette problématique.

Sélection d'événements acoustiques en VAL

par *Geoffroy Querol* © 9 mai, 17h00, amphi IP11

La vérification automatique du locuteur (VAL) possède aujourd'hui de solides bases dans les modèles probabilistes : ceux-ci représentent un modèle permettant de prendre en compte les spécificités de chaque locuteur. Une question que l'on peut se poser est comment extraire l'information qui est pertinente pour un locuteur particulier. Ainsi, à partir des modèles probabilistes, je vais présenter deux méthodes qui permettent de sélectionner ces informations.

Enfin, afin de confirmer cette expérience, je vais présenter la fusion de systèmes de reconnaissance afin de déterminer si les critères utilisés sont complémentaires ou redondants.

Compensation de canal par Factor Analysis

par *Charles Deledalle* © 9 mai, 17h30, amphi IP11

Dans le cadre de la vérification du locuteur, le LRDE participe depuis 2 ans aux évaluations de reconnaissance du locuteur organisées par le NIST. Le NIST fournit une vaste base d'enregistrement audio, une partie est destinée à l'entraînement du système et l'autre aux tests. Cette année, ces deux bases sont réalisées sur des canaux différents : téléphone pour l'entraînement et microphone pour les tests. Pour

pallier cette difficulté, l'information provenant du canal doit être retirée du signal. Parmi différentes techniques de compensation de canal, je vous présenterai une méthode prometteuse à base de Factor Analysis que j'étudie et développe en ce moment.

Comparaison entre l'utilisation du noyau linéaire et non linéaire pour les systèmes de vérification du locuteur fondés sur les méthodes SVM

par *Reda Dehak* © 9 mai, 18h00, amphi IP11

Je présenterai les résultats d'une comparaison entre l'utilisation d'une fonction noyau linéaire et une fonction noyau non linéaire dans le cas des systèmes de vérification du locuteur fondés sur les méthodes SVM. Ces deux noyaux sont construits à partir d'une distance définie dans l'espace des paramètres GMM. Je présenterai le lien existant entre ces deux fonctions noyaux et comment exploiter les résultats des méthodes de compensation du canal (NAP) dans les deux cas.

On a démontré l'importance de la normalisation des paramètres des modèles GMM dans le cas de la fonction noyau non linéaire. Toutes nos expérimentations ont été conduites sur la base d'évaluation NIST-SRE 2006 core condition (all trial). Le meilleur score (un EER de 6.3%) est obtenu avec un noyau non linéaire sur des GMMs normalisés.

OLENA

Une bibliothèque de conteneurs en C++ via SCOOP

par *Thomas Moulard* © 16 mai, 14h00, amphi 2

SCOOP est le paradigme de programmation utilisé par Olena. Il permet l'expression de mécanismes utiles qui ne sont pas directement disponibles dans le C++ actuel tels que les concepts, les types virtuels ou bien les where-clauses. Ce paradigme présente également l'avantage d'être très rapide et sûr car l'ensemble des vérifications sont réalisées à la compilation. Ces outils peuvent également être utilisés dans d'autres contextes tels que la réécriture de la bibliothèque de conteneurs du C++ comme nous le verrons ici.

Cette présentation se déroulera en trois parties. Tout d'abord, un tour d'horizon rapide du paradigme sera réalisé. Puis, on expliquera comment implémenter une hiérarchie SCOOP via les outils fournis par les bibliothèques d'Olena. Enfin, la nouvelle

bibliothèque de conteneurs sera présentée, ainsi que les nouvelles possibilités offertes par rapport à la bibliothèque traditionnelle. En particulier, celle-ci permet l'écriture de morphers pouvant effectuer des transformations sur les types de conteneurs.

Olena, comme les autres ?

par *Thierry Géraud* © 06 juin, 15h30, amphi 2

Si, d'un point de vue lointain, toutes les bibliothèques de traitement d'images se ressemblent, elles n'ont néanmoins pas été pensées avec les mêmes objectifs et, par conséquent, ne partagent pas le même ensemble de caractéristiques. Cet exposé, non technique, a pour but de présenter les traits principaux qui permettent de définir l'identité de la bibliothèque Olena. Nous expliquerons nos intentions et nous justifierons les options que nous avons choisies, ainsi que les compromis que nous avons dû réaliser. En particulier, nous insisterons sur les aspects originaux de notre démarche qui, d'une part, peuvent per-

mettre à Olena de se distinguer des autres bibliothèques et, d'autre part, peuvent inciter des utilisateurs potentiels à choisir notre bibliothèque.

Génie logiciel au coeur d'Olena

par *Nicolas Ballas*

© 06 juin, 16h00, amph 2

Pour s'assurer de la qualité d'un logiciel et de la facilité de son développement, le génie logiciel définit un certain nombre de critères à respecter.

Dans le domaine du traitement d'images, un nombre important de types différents d'images existent. Il est donc difficile de construire une bibliothèque, dédiée à ce domaine, proposant des outils réutilisables, facilement extensibles ou encore compatibles avec des logiciels tiers.

Nous verrons comment différentes techniques, issues du paradigme de programmation SCOOP mis en place dans Olena, répondent à ce problème.

Enfin nous analyserons des solutions fournies par d'autres bibliothèques de traitement d'images.

Canevas d'algorithmes morphologiques

par *Ugo Jardonnet*

© 06 juin, 16h30, amph 2

Olena est une bibliothèque générique de traitement d'images développée au LRDE. Elle propose un grand nombre d'algorithmes morphologiques. La morphologie mathématique, inventée à l'origine pour traiter des images binaires, offre des outils très puissants de traitement et d'analyse d'images. Des similarités apparaissant dans l'écriture des algorithmes, il est possible de classer ces outils et, ainsi,

de proposer un certain nombre de « canevas » d'algorithmes. Il s'agit alors d'apporter des solutions d'implémentation non redondante de ces canevas. Celles-ci doivent être cohérentes avec les notations mathématiques manipulées.

A travers une brève initiation à la morphologie mathématique, l'exposé présentera différents canevas d'algorithmes retenus par Olena.

Étude comparative de ConceptC++ et perspective d'intégration dans SCOOP

par *Alexandre Abraham*

© 06 juin, 17h15, amph 2

La fin de cette décennie verra l'avènement de C++0x et avec lui du nouveau paradigme de « concepts ». Les concepts fourniront un mécanisme de typage abstrait pour les types paramétrés ainsi que tout l'équipement d'adaptation des types concrets à ces types abstraits comme le fait actuellement la bibliothèque *Static*, composant du projet Olena.

Nous ferons donc un comparatif de ces approches en exhibant leurs points forts et faibles ainsi que leurs capacités particulières afin de proposer un support de documentation et une base pour la future intégration des concepts dans le paradigme SCOOP.

Les concepts pourraient en effet simplifier l'écriture du code client et enrichir SCOOP de fonctions supplémentaires.

DDD

Diagrammes de décision à valeurs supprimées

par *Samuel Charron*

© 16 mai, 14h30, amph 2

Les diagrammes de décision sont des structures permettant de représenter de grands ensembles de données. Le partage de données communes aux éléments de l'ensemble permet une grande compacité en mémoire.

Différentes techniques ont déjà été proposées pour certaines familles de diagrammes de décisions afin de réduire encore l'utilisation mémoire. L'une d'elles, la réduction, existe dans le domaine des Diagrammes de Décision Binaires (BDD).

En appliquant cette technique aux Diagrammes de Décisions de Données (DDD), nous espérons un gain de mémoire. Cependant, la définition de la réduction doit être adaptée aux DDD et SDD (Set Decision Diagrams) afin de conserver leurs propriétés. De même la manipulation usuelle des DDD doit être adaptée pour tenir compte de la réduction.

Durant la présentation, nous verrons comment nous avons réalisé l'adaptation de la réduction des BDD aux DDD. Puis à travers une implémentation générique des diagrammes de décision, nous montrerons les abstractions réalisées afin de pouvoir utiliser de manière générique cette technique aussi bien pour les BDD, que pour les DDD et SDD.

TRANSFORMERS

Préprocesseur C/C++ réversible

par *Quentin Hocquet* © 16 mai, 15h00, amphi 2

La transformation de programme C++ présente une difficulté supplémentaire par rapport à la plupart des autres langages : la phase de preprocessing. Il s'agit d'une étape complexe, dans la mesure où elle n'a absolument pas été pensée pour être inversible. Pourtant l'utilisateur d'un système de transformation de C++ souhaite retrouver son code d'origine, avec les directives de compilation et la mise en page initiale.

Ce séminaire présentera les techniques utilisées pour rendre le processus réversible et pour créer sa réciproque.

Formalisation de la désambiguïsation avec des grammaires attribuées

par *Nicolas Pierron* © 16 mai, 15h45, amphi 2

Le problème actuel de la désambiguïsation dans TRANSFORMERS avec des grammaires attribuées est que l'on ne possède pas de preuve permettant de certifier cette approche. L'usage actuel des grammaires attribuées pour la désambiguïsation du C et d'une partie du C++ laisse à penser que cette méthode est correcte.

Afin de supprimer tout doute, une définition et une formalisation de notre approche sont nécessaires. Ce travail comporte deux volets. La première partie porte sur la preuve de la validité de l'approche utilisée dans TRANSFORMERS. La seconde partie est consacrée à la correction et au re-développement des outils existants afin de correspondre au modèle défini.

Désambiguïsation guidée par la sémantique

par *Renaud Durlin* © 16 mai, 16h15, amphi 2

Une approche élégante pour gérer les grammaires ambiguës consiste à utiliser un parseur LR généralisé qui produira non pas un arbre mais une forêt de parse. Une étape supplémentaire, appelée désambiguïsation, survenant juste après le parsing, est alors nécessaire. Celle-ci consiste à analyser cette forêt pour obtenir l'unique arbre valide correspondant à l'entrée en prenant en compte les règles de sémantiques contextuelles.

C'est cette approche qui a été retenue dans TRANSFORMERS avec le formalisme des grammaires attribuées. Le travail effectué présentera une comparaison entre ce formalisme et deux autres techniques de désambiguïsation : la première à l'aide

d'ASF+SDF et la deuxième à l'aide du langage *Stratego*.

Le but de cette comparaison sera double : montrer que les grammaires attribuées sont parfaitement adaptées à ce problème et exhiber les faiblesses de celles-ci par rapport aux deux autres méthodes en vue d'une amélioration possible du système utilisé dans TRANSFORMERS.

Centaur : Une infrastructure pour la transformation de C++

par *Benoit Sigoure* © 16 mai, 16h45, amphi 2

Transformers a choisi de suivre la grammaire du standard pour ses parseurs de C et C++. Ces grammaires ayant été conçues pour des parseurs LALR, elles sont relativement difficiles à manipuler lorsqu'il s'agit d'écrire des transformations ou faire de l'analyse statique. Le but de Centaur est de fournir aux utilisateurs de Transformers une bibliothèque de fonctions permettant de manipuler aisément du code C++. Elle permettra d'accéder simplement aux informations disponibles dans l'AST et ses annotations (pour répondre à des requêtes telles que : lister les éléments d'un namespace, rechercher des méthodes dans une classe en fonction de plusieurs critères, lister les classe parentes d'une classe, etc.)

SCOOL, orientation objet d'un langage statique

par *Maxime van Noppen* © 06 juin, 17h45, amphi 2

SCOOL est un langage statique orienté objet qui a été créé afin de pouvoir utiliser toute la puissance du C++ statique avec une plus grande facilité grâce à une syntaxe plus expressive et agréable. Il n'a pas pour but d'être directement compilé mais d'être traduit en C++ équipé du paradigme SCOOP. C'est ce paradigme qui va nous permettre d'ajouter au C++ les fonctionnalités statiques dont nous avons besoin. Les fonctionnalités principales du langage sont de permettre l'héritage statique, les types virtuels, les clauses *where* dans les fonctions paramétrées ainsi que les multiméthodes.

Le travail de cette année s'est essentiellement porté sur la gestion de la couche objet dans le langage et en particulier l'héritage statique. Une attention toute particulière s'est également portée sur l'environnement de travail fourni à l'utilisateur, comme par exemple les messages d'erreurs fournis par le traducteur.

Typage du C++ par grammaires attribuées

par Florian Quèze

© 06 juin, 18h15, amphi 2

TRANSFORMERS est un ensemble d'outils permettant la manipulation de programmes C++ en Stratego/XT. La grammaire standard du C++, utilisée par le projet TRANSFORMERS, est ambiguë. Lors de l'analyse syntaxique d'un code en C++, il est donc nécessaire d'effectuer une désambiguïsation.

Le projet TRANSFORMERS désambiguïse à l'aide des grammaires attribuées. Des attributs sont associés à chaque symbole de règle de la grammaire, permettant de stocker directement dans l'arbre d'analyse syntaxique les informations nécessaires.

Cependant, pour achever la désambiguïsation du C++, il est nécessaire de typer. Ce calcul des types peut lui aussi se faire à l'aide des grammaires attribuées. L'objectif de cet exposé est de montrer comment, puis de conclure sur les perspectives pour le futur du projet TRANSFORMERS.

VAUCANSON

Un format de description d'automates et son utilisation dans VAUCANSON

par Florian Lesaint

© 23 mai, 14h00, amphi 2

La proposition XML présentée à CIAA 2005 (Conférence on Implementation and Application of Automata) montre certaines lacunes. La gestion des étiquettes complexes doit notamment y être ajoutée.

Ce séminaire a pour but de finaliser la proposition d'un format universel pour la description d'automates afin de faciliter la communication entre les divers outils qui leur sont consacrés.

Cette dernière fera l'objet d'une soumission pour le Workshop FSMNLP 2007 (Finite-State Methods and Natural Language Processing 2007) et se verra proposée lors d'une éventuelle session XML à CIAA 2007.

Une seconde étape consistera à modifier VAUCANSON pour lui permettre de supporter ce format, via une réimplémentation de son parseur XML. Ce sera l'occasion de passer d'un modèle DOM à un modèle SAX, afin d'en réduire l'utilisation mémoire et espérer ainsi pallier les piètres performances de VAUCANSON.

Boosting Vaucanson - partie 1

par Guillaume Lazzara

© 23 mai, 14h30, amphi 2

Suite aux séminaires de l'année dernière, il en ressort que les performances globales de Vaucanson pouvaient largement être améliorées par l'usage de tables de hachage et plus particulièrement celles de la bibliothèque Boost Multi Index.

Pour ce séminaire, nous chercherons à tirer parti des nouvelles fonctionnalités offertes par Boost. Ceci impliquera l'apparition d'une nouvelle implémentation de graphe. Les automates n'étant plus représentés par des ensembles d'états et de transitions mais

uniquement par un seul grand ensemble indexé de transitions.

L'étude de cette nouvelle implémentation ayant été réalisé indépendamment de Vaucanson, son intégration pourrait impliquer des remaniements dans l'API interne. Nous présenterons au cours de ce séminaire les enjeux induits par ces changements sur l'implémentation.

Boosting Vaucanson - partie 2

par Jimmy Ma

© 23 mai, 15h00, amphi 2

Au cours de ces séminaires, Vaucanson a été muni d'une nouvelle implémentation d'automates. Notre but étant d'évaluer l'impact de ces changements sur les performances, en particulier dans les algorithmes.

Dans un premier temps, nous dresserons un bilan brut sur les nouvelles performances. Le but étant de trouver les algorithmes où les nouvelles fonctionnalités apportées ne sont pas pleinement exploitées.

Dans un deuxième temps, nous fournirons une nouvelle implémentation de certains algorithmes en utilisant la nouvelle API.

Enfin, nous dresserons un bilan global sur l'état actuel de Vaucanson.

Algorithmes génériques de suppression des transitions spontanées

par Vivien Delmon

© 23 mai, 15h45, amphi 2

Beaucoup d'algorithmes sur les automates prennent en entrée des automates sans transitions spontanées. L'algorithme de suppression de ces dernières est donc un point central dans VAUCANSON.

L'implémentation actuelle de l'algorithme générique est assez rapide mais gourmande en mémoire. Une autre version proposée par Sylvain LOMBARDY devrait être moins gourmande et plus rapide par la même occasion. Par ailleurs, un algorithme a été publié par Mehryar MOHRI des AT&T Labs et sera également mis à l'épreuve lors de ce séminaire.

Ces deux algorithmes seront implémentés d'une part sur l'API actuelle de VAUCANSON et d'autre part sur la future API qui devrait apporter de nouvelles fonctions grâce notamment à l'intégration d'une structure basée sur les Boost Multi Index.

Transducteurs synchronisés

par *Guillaume Leroi* © 23 mai, 16h15, amphi 2

Un transducteur synchronisé est un transducteur dont les étiquettes sont des couples de lettre et les fonctions finales sont de la forme (L,1) ou (1,L) (où 1 est le mot vide et L un langage rationnel).

Une opération de base dans le calcul d'un transducteur synchronisé est la circulation des sorties

d'un transducteur. La circulation consiste à déplacer un mot w en arrière à travers un état. Ce mot w est un préfixe commun à toutes les étiquettes des transitions sortantes de cet état. On le supprime donc pour le rajouter en suffixe de toutes les transitions entrantes de ce même état.

Cette opération de circulation est nécessaire pour permettre, par exemple, la minimisation d'un transducteur séquentiel ou la resynchronisation d'un transducteur.

Ce séminaire se propose de présenter l'implémentation de ces algorithmes dans VAUCANSON, ainsi que de proposer une structure de données pour les transducteurs synchronisés adaptable aux transducteurs à plus de deux bandes.

Théorie des jeux

Méthodes algorithmiques de recherche d'équilibres de Nash

par *Antoine Leblanc* © 23 mai, 16h45, amphi 2

La théorie des jeux est généralement décrite comme une approche mathématique de problèmes de prise de décisions. Les équilibres de Nash sont l'un des principaux concepts qu'elle a introduits. Dans un jeu, un équilibre de Nash est une position

dans laquelle aucun des joueurs n'a intérêt à changer sa stratégie. L'un des problèmes majeurs engendrés par ces équilibres est celui de la complexité algorithmique...

Le but de cet exposé sera non seulement de faire le point sur les avantages et inconvénients des algorithmes usuels, mais également de présenter une nouvelle méthode de calcul basée sur une approche géométrique du problème.

Parallélisation

Une bibliothèque générique de parallélisation en C++

par *Elie Bleton* © 23 mai, 17h15, amphi 2

Nous avons pour but la conception d'une bibliothèque générique de parallélisation et de distribution d'application en C++, la libpapa. Cette bibliothèque cible les développeurs souhaitant paralléliser ou répartir soit une application existante basée sur du code non modifiable, soit une nouvelle applica-

tion, qui peut alors tirer le meilleur parti de la bibliothèque.

La libpapa n'a pas vocation à remplacer les solutions existantes mais à fournir des abstractions de haut niveau pour le développeur.

Ce séminaire abordera les solutions existantes que libpapa se propose de reprendre ou d'encapsuler, les choix faits au niveau de la modélisation de cette bibliothèque ainsi que des exemples d'utilisation possible.

LISP

Étude de problèmes de performances en C et en Common Lisp

par *William Caldwell* © 06 juin, 15h00, amphi 2

Dans son article *Beating C in Scientific Computing Applications*, Didier Verna a laissé un certain nombre

de questions en suspens. Notre but est de chercher des explications aux performances inattendues en C et en Common Lisp concernant l'inlining et les algorithmes d'affectation et de division.

Ce travail vise à trouver des explications à des résultats expérimentaux exposés dans l'article.

En bref

Les nouvelles publications (disponibles sur publis.lrde.epita.fr)

- QUOC, C. L., BELLOT, P., AND DEMAILLE, A.. On the security of quantum networks : a proposal framework and its capacity. In *Proceedings of the 2007 International Conference on New Technologies, Mobility and Security (NTMS'07)*, Paris, France
- HAMEZ, A., KORDON, F., AND THIERRY-MIEG, Y.. libDMC : a library to operate efficient distributed model checking. In *Workshop*

on Performance Optimization for High-Level Languages and Libraries — associated to IPDPS'2007

- BERGER, CH., GÉRAUD, TH., LEVILLAIN, R., WIDYNSKI, N., BAILLARD, A., AND BERTIN, E.. Effective component tree computation with application to pattern recognition in astronomical imaging. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, San Antonio, TX, USA

Pour passer le temps...

Nous vous proposons ce petit sudoku :

							5	1
				3	6			
	4		5				8	
2						6		
					1			
				2		3	4	
	1		4			7		
6								

Que font les programmes suivants ? (À faire sans l'aide d'un compilateur ou d'un interprète, bien sûr !)

- C :

```
main(int j, char**V){char*R=V[1], i=0, k=48; for (;*R>k;***R||
puts(R-i))++i; for(++k<58;*R&&main(*R=k,V),*R=1) for(j=81; j
--;)*R*=R[j-i]-k||i/9^j/9&&i%9^j%9&&i/27^j/27|i%9/3^j%9/3;}
```

- Perl :

```
$_=$'. $_. $'.<>;split//;${/[@_[map{$i-($i="@-")%9+$_,9*$_+$i%
9,9*$_%26+$i-$i%27+$i%9-$i%3}0..8]}/o||do$0}for/0/||print..9
```