Modification de PadicoTM afin de fournir une interface de type Madeleine

Christophe Frezier

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

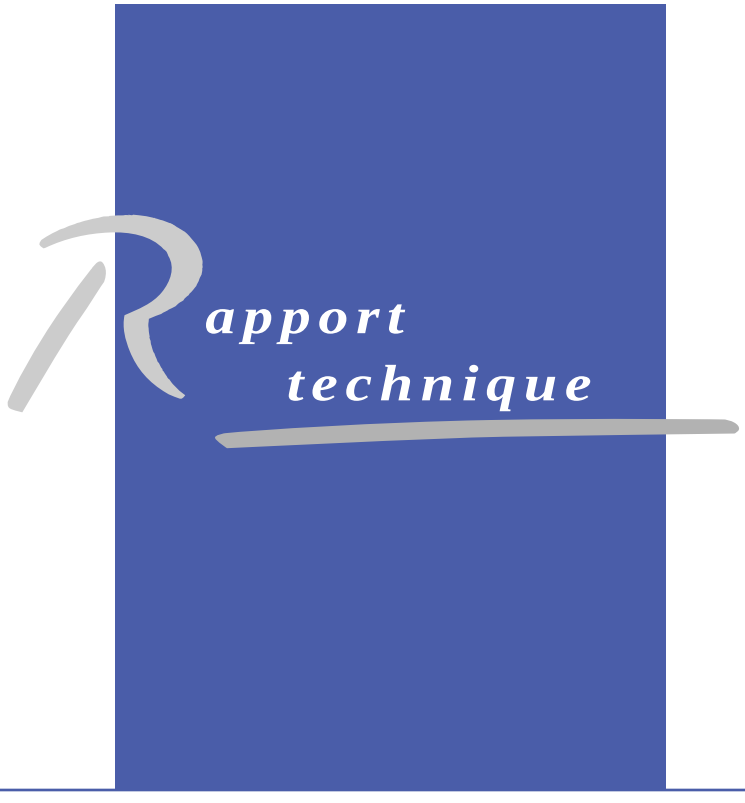
Modification de PadicoTM afin de fournir une interface de type Madeleine

Christophe Frézier

N° 0334

Janvier 2007

Thème NUM



ISSN 0249-0803 ISRN INRIA/RT--0334--FR+ENG

Modification de PadicoTM afin de fournir une interface de type Madeleine

Christophe Frézier

Thème NUM — Systèmes numériques
Projet Runtime

Rapport technique n° 0334 — Janvier 2007 — 8 pages

Résumé : Ce document décrit les améliorations récentes apportées à PadicoTM. Elles concernent des aspects utilisation, comme l'interface graphique, ainsi que les fonctionnalités fournies par le logiciel. Certaines des interfaces bas-niveau ont été modifiées et un nouveau service, Madico, a été créé afin de permettre aux applications utilisant l'interface de communication Madeleine de s'exécuter au-dessus de PadicoTM.

Mots-clés : Madico, Madeleine, grille, communications haute performance, Grid'5000, interface graphique, mémoire partagée, composant

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Changes in PadicoTM in order to provide a Madeleine-like interface

Abstract: This document describes the recent improvements of the PadicoTM software. They are related to usability aspects, such as the graphical interface, and to some of the functionalities provided by the software. Some low-level interfaces have been modified and a new service, Madico, has been created to allow applications using the Madeleine communication library to run on PadicoTM.

Key-words: Madico, Madeleine, grid, high performance communications, Grid'5000, graphical interface, shared memory, adapter

1 Préface

1.1 Cadre

Le projet de recherche Runtime s'inscrit dans le cadre du calcul parallèle haute performance. Il contribue à l'étude et à l'élaboration des supports d'exécution sur lesquels s'appuieront les environnements de programmation et les applications de calcul intensif de demain. La plupart des activités de cette équipe de recherche débouchent sur la réalisation de plateformes logicielles disponibles sur de nombreuses architectures et librement accessibles sur Internet. Les travaux de recherche menés portent sur une nouvelle génération de supports exécutifs capables d'exploiter au mieux les architectures parallèles de type *grappe* ou *grille*, comme celle mise en place par le projet Grid'5000.

1.2 Présentation de l'existant

Madeleine, Marcel, Padico et PM2 sont développés dans le projet Runtime pour arriver à cette nouvelle génération de supports exécutifs. PadicoTM, développé par Alexandre Denis, est un support exécutif concernant les grilles informatiques (ou grappes) développé en collaboration avec l'équipe de recherche de PARIS. Il se compose d'un noyau qui fournit un cadre de travail pour la gestion de réseau et le multi-threading, ainsi que des services supplémentaires. Marcel fournit la gestion des threads et Madeleine fournit les communications hautes performances, logiciels par la suite logicielle PM2. Mais pour l'instant, il faut plus parler "d'empilement" des bibliothèque plutôt que d'une véritable coopération. Le noyau de PadicoTM vise à offrir les différents services en même temps d'une manière plus coopérative que concurrentielle. On peut voir l'architecture de PadicoTM sur la figure 1 ; le cœur de Padico est formé d'un micro noyau chargé des opérations de base sur les modules, qui est l'entité de base pour le chargement dynamique dans PadicoTM. Il contient aussi un gestionnaire de threads et un module d'arbitrage d'accès au réseau appelé NetAccess. Circuit et Vsock sont des services fournis par Padico (qui fournissent respectivement une interface à passage de messages et réception par messages actifs, et une interface socket classique) et qui sont directement utilisables par les utilisateurs. On peut leur adjoindre d'autres services pour ajouter ou modifier des fonctionnalités (cryptage SSH, interface MPI, etc.) C'est le NetSelector qui s'occupe de savoir dans certains cas, comment empiler les composants à partir d'un fichier de configuration. Le duo Madeleine et Marcel (PM2) sont les bibliothèques qui servent à communiquer et à gérer les threads respectivement.

2 Travaux réalisés

Dans cette plateforme basée sur une architecture à composants, de nombreux changements ont été apportés, afin d'améliorer les performances et les fonctionnalités. Le but a consisté à faire coopérer plus activement PadicoTM et Madeleine, notamment utiliser les possibilités de Madeleine en termes d'optimisation plutot que de simplement lui faire envoyer

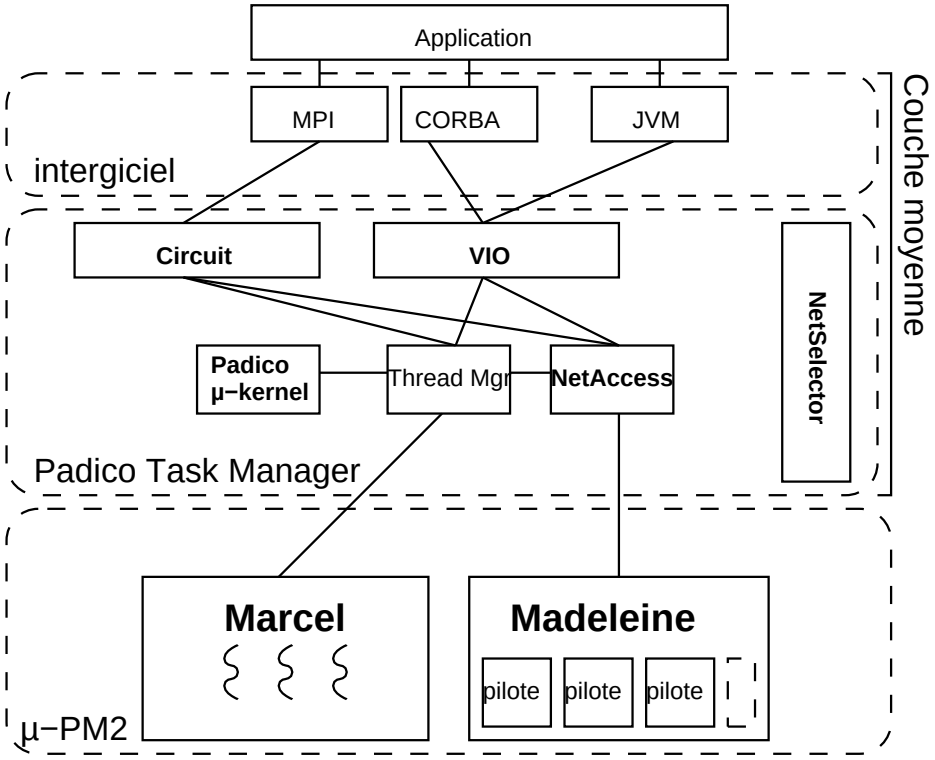


FIG. 1 – L'architecture de PadicoTM

les paquets. Plus précisément, cela a consisté en la création d'un composant `Madico` permettant d'utiliser directement PadicoTM avec l'interface Madeleine tout en gardant l'ensemble des fonctionnalités, puis à modifier ensuite certaines autres parties du logiciel (notamment la partie `Circuit` offrant des fonctionnalités demandées par le paradigme de programmation par passage de message, comme par exemple MPI) pour profiter pleinement du potentiel offert par cette modification.

2.1 Madico

Dans le but de faire tourner des applications Madeleine sur PadicoTM et ainsi de profiter des avantages offerts par ce dernier, le composant `Madico` a été ajouté en tant que service de PadicoTM. Ce dernier est composé d'un "faux" driver logiciel inclus dans Madeleine, ainsi que du composant `Madico` à proprement parler. En utilisant directement Madeleine avec un canal (canal de communication, représentant l'abstraction physique d'un réseau ou un réseau virtuel, ce qui est le cas ici) reposant sur ce pilote logiciel, nous pouvons utiliser le

composant `Madico` qui se sert lui-même du composant `Circuit` de PadicoTM pour fournir les communications. On peut voir le trajet des données dans les deux cas sur la figure 2. Ainsi, cela nous permet de traverser les couches logicielles de PadicoTM et de profiter des performances et des fonctionalités offertes, en particulier l'optimisation de l'envoi de paquets (qui est dans la cadre d'une thèse au sein du projet Runtime) :

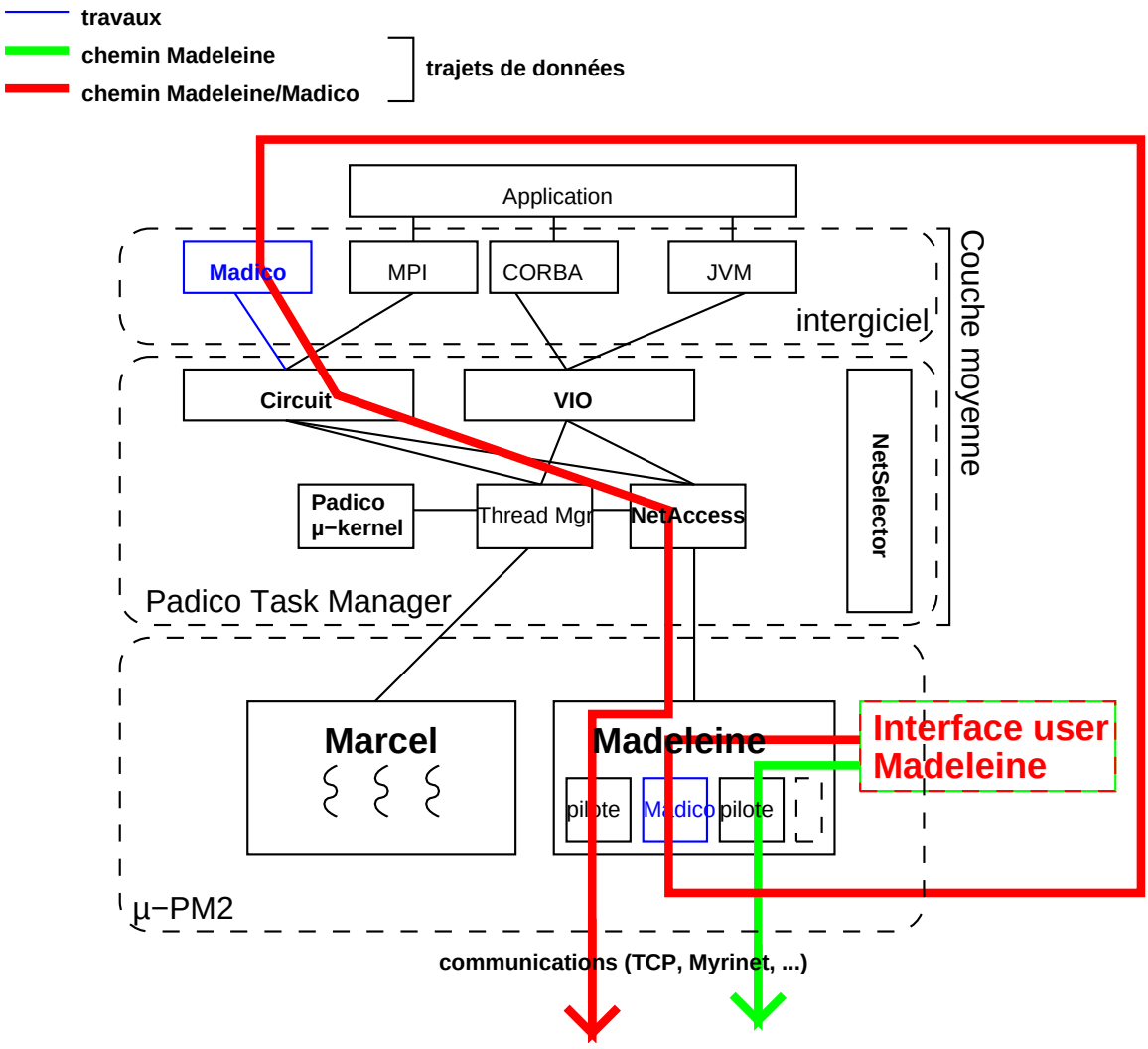


FIG. 2 – Trajets des données dans les couches logicielles lors des envois et réceptions avec l'interface Madeleine avec PM2 et avec Madico

2.2 Mémoire partagée et PadicoSimplePackets

à la vue du nombre croissant de machines multi-processeurs, un module fournissant des communications en mémoire partagée performantes au sein de PadicoTM était nécessaire. De plus, l'interface PadicoSimplePackets, qui existait à l'état d'ébauche a l'arrivée de Christophe dans le projet Runtime, est une interface flexible de bas niveau permettant d'utiliser directement Madeleine. Cela correspondait avant à un cablâge en dur au niveau du code à l'utilisation de Madeleine par PadicoTM. Elle est maintenant devenue une véritable intégration dans le sens où Padico repose maintenant sur les PSP pour atteindre Madeleine, et que Madeleine repose sur PadicoTM (utilisation possible grâce à `Madico`).
Le module Shm fournit l'interface PadicoSimplPackets (PSP) grâce à des échanges au travers d'une mémoire partagée, dans le cas de lancement de plusieurs noeuds sur une même machine. Les changements de l'interface sont les suivants : l'interface PSP comportait une fonction `init` qui a été conservée ainsi qu'un `get_sbuf` et un simple `send` qui ont été remplacés par un triplet classique de fonctions `new_message`, `pack` et `end_message`. Avec cette interface, la réception se fait par messages actifs.

2.3 Modification de l'interface graphique

La couche moyenne (couche d'abstraction) de PadicoTM se compose d'un coeur et de composants assemblés de manière libre et dynamique (visible sur la figure 1). Le processus d'assemblage est dirigé par un "décideur" configurable (le NetSelector). Il existe plusieurs implémentations de ce NetSelector dans PadicoTM. Le NetSelector était auparavant choisi à la compilation et ne pouvait pas être changé. L'interface graphique permet maintenant lors de l'exécution de pouvoir choisir le NetSelector, ainsi que le fichier de configuration de ce dernier. Il est donc possible de choisir le NetSelector lors de la compilation, ou bien de spécifier qu'il sera donné par l'interface graphique PadicoTMControl.

2.4 Modification de l'interface Circuit

Circuit est l'interface qui avait été développée pour fournir une interface de passage de messages dans PadicoTM. Madeleine présentait déjà une interface correspondant à ce besoin, sauf que celle-ci ne reposait pas sur des réceptions en messages actifs, mais sur des réceptions explicites. L'interface fournie par Circuit proposait des choses déjà faites dans Madeleine et l'interface Madeleine correspondait tout à fait avec les besoins des utilisateurs (pas besoin de massages actifs). Avec l'arrivée de `Madico`, l'interface Madeleine est fournie par PadicoTM : il était plus logique en effet d'unifier l'interface d'utilisation de Madeleine et de PadicoTM et utiliser uniquement l'interface `Circuit` à l'intérieur de PadicoTM (pour profiter des performances de la réception par messages actifs). Cela nous permet aussi de réutiliser le code de Madeleine.
De plus, l'interface PSP correspond tout à fait aux besoins pour une interface à passage de message. Nous avons donc créé un module permettant de fournir Circuit à partir de l'interface PSP, en implémentant une boîte aux lettres à la réception.

Dans un premier temps, une version de l'ancien `Circuit` basée sur la nouvelle interface PSP a été créée. Ensuite c'est l'interface fournie par `Circuit` qui a été modifiée. Cette partie est en cours de développement, et les premiers tests ont été assez concluants. Nous avons donc décidé d'améliorer encore le composant `Circuit` afin de le rendre compatible avec des grappes complexes, gérant plusieurs sous-circuits éventuellement de type différents pour assurer la connectivité. Cette nouvelle avancée nous a fait découvrir d'autres problèmes liés dans d'autres modules, comme par exemple la gestion de la topologie réseau, ainsi que le NetSelector, qui ne pouvait jusqu'à lors pas nous fournir les informations nécessaires à la construction de ces sous-circuits. Tous ces problèmes sont actuellement en train d'être réglés. Un exemple de cas d'utilisation des sous-circuits est donné sur la figure 3. En effet sur cette

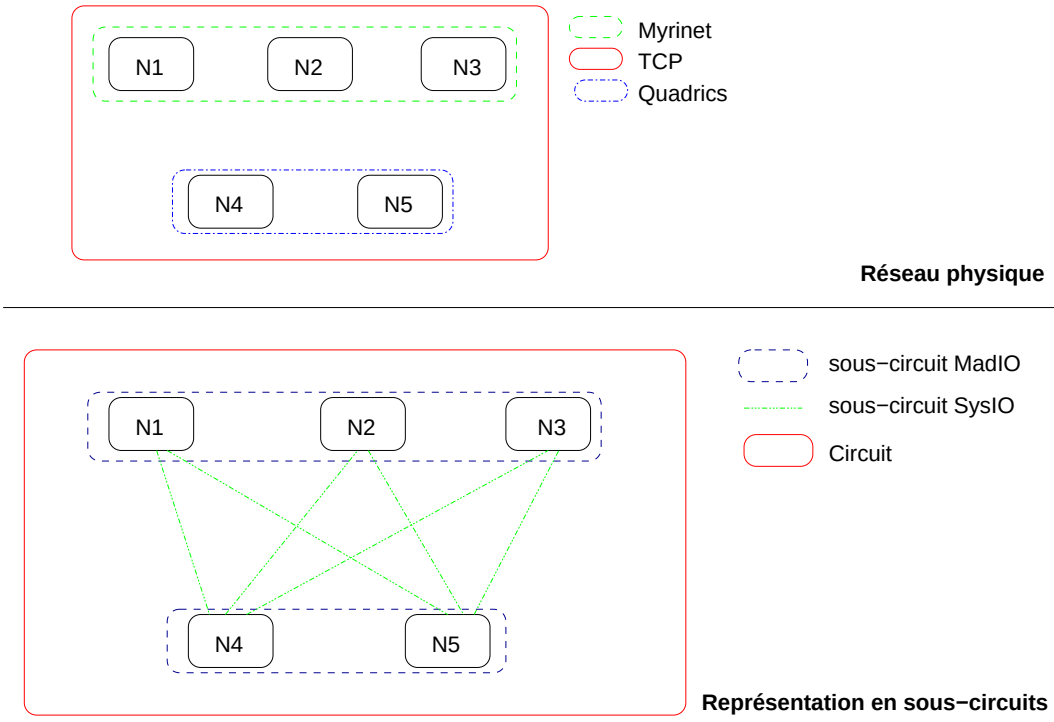


FIG. 3 – Exemple de cas d'utilisation avec nécessité des sous-circuits

figure, on peut voir un ensemble de machines reliées par TCP, et dont certaines sont reliées par d'autres réseaux. Dans ce cas, lors de la construction d'un circuit sur cet ensemble de noeuds, le NetSelector doit donner aux circuits les informations nécessaires pour construire l'ensemble des 8 sous-circuits qui permettront de fournir un circuit à la couche supérieure. Dans ce cas, chaque nœud est connecté à tous les autres par un sous-circuit. Avant la gestion

des sous-circuits, Circuit ne pouvait supporter qu'un seul de ces types de sous-circuits dans un circuit.

Table des matières

1 Préface **3**
1.1 Cadre 3
1.2 Présentation de l'existant 3

2 Travaux réalisés **3**
2.1 Madico 4
2.2 Mémoire partagée et PadicoSimplePackets 6
2.3 Modification de l'interface graphique 6
2.4 Modification de l'interface Circuit 6

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803