

WP2T23 «APPLICATION de REFERENCE»
Prototypage rapide
d'une application de traitement
d'images avec SynDEx

EADS

Avec l'appui de l'INRIA www-rocq.inria.fr/syindex

I. Présentation du sujet

II. Présentation de l'algorithme

- Cahier des charges
- Graphe flot de contrôle extrait du programme C

III. Présentation de SynDEx

- Méthodologie AAA/SynDEx
- Intérêts

IV. Implantation avec SynDEx v5

- Démarche
- Résultats de la simulation
- Test sur architecture réelle
- Conclusion de l'implantation avec SynDEx V5

V. Implantation avec SynDEx v6

- Démarche
- Exploration des implantations de l'algorithme sur différentes architectures
- Analyse des résultats
- Conclusion de l'implantation avec SynDEx V6

VI. Conclusion

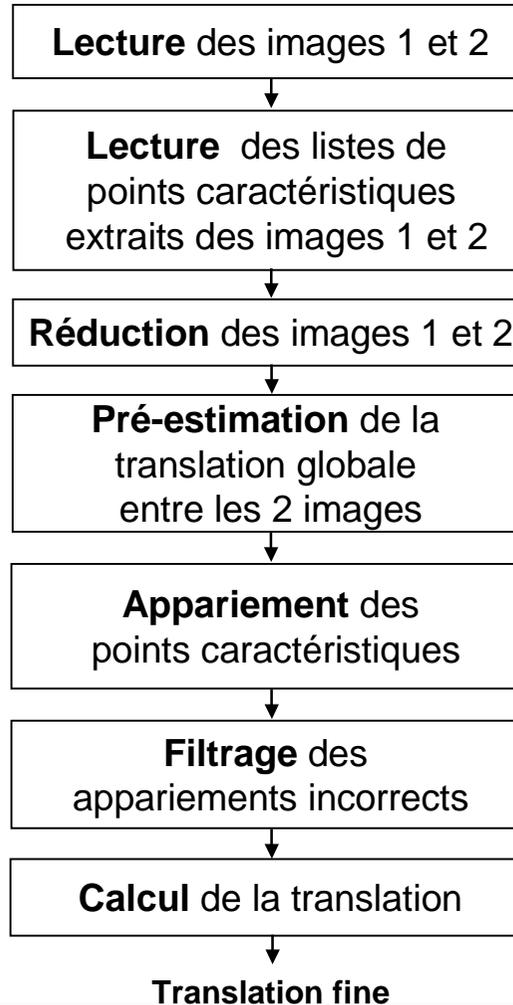
- Étude du prototypage d'une application de traitement d'images avec SynDEX
- Simulation de l'algorithme de traitement d'images sur différentes architectures à base de processeurs ADSP21060

Cahiers des charges

- **Algorithme** de calcul d'une translation entre deux images
- **Livraison** du document n° E/SCS-2000-759-A contenant :
 - présentation générale de l'algorithme
 - programme en langage C
 - synoptique de l'architecture matérielle

PRESENTATION DE L'ALGORITHME

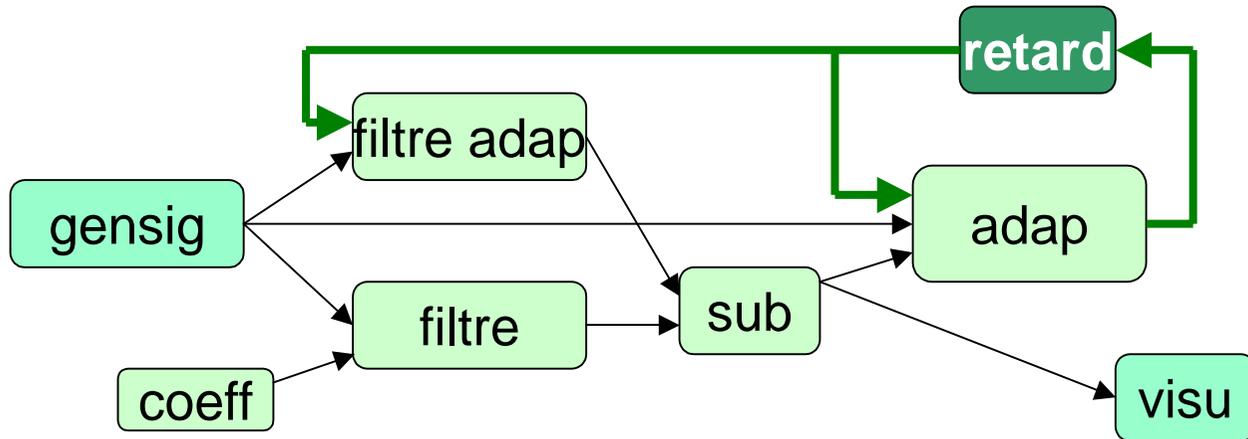
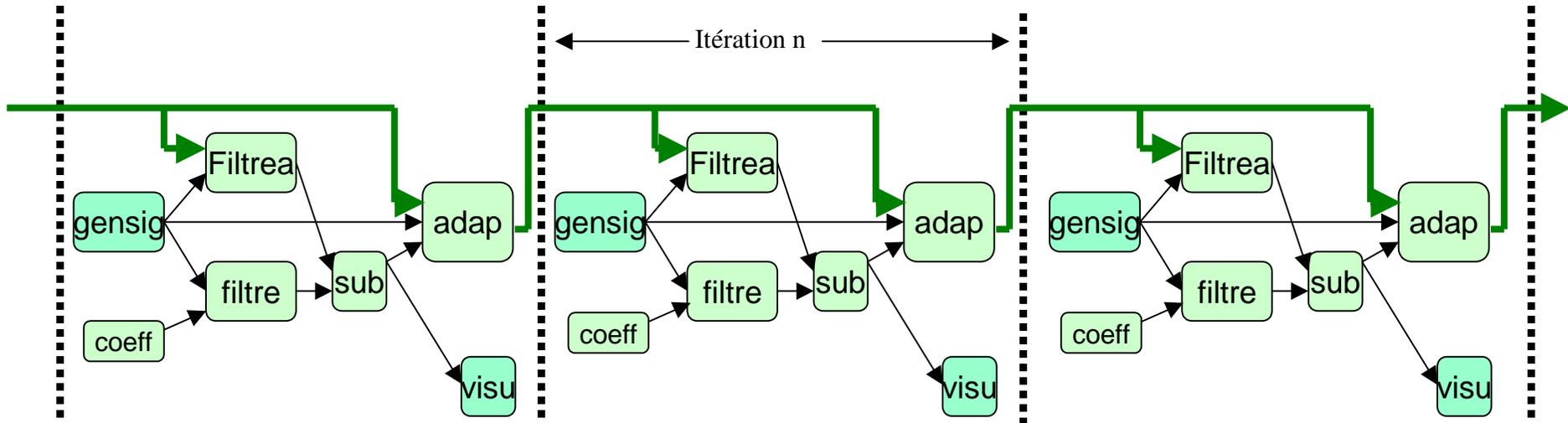
Graphe flot de contrôle extrait du programme C



- **Approche globale** fondée sur une extension de la sémantique des langages synchrones
- **Modèle unifié** : graphes
 - **Algorithme** : opération / dépendance : donnée, conditionnement
 - **Architecture** : opérateur calcul-commu., mémoire / connexion
 - **Implantation** : transformations de graphes : distribution/ordonnancement
- **Adéquation** : implantation optimisée
- **Macro-génération**
 - Exécutifs pour multicomposant : réseau de processeurs et de circuits intégrés spécifiques

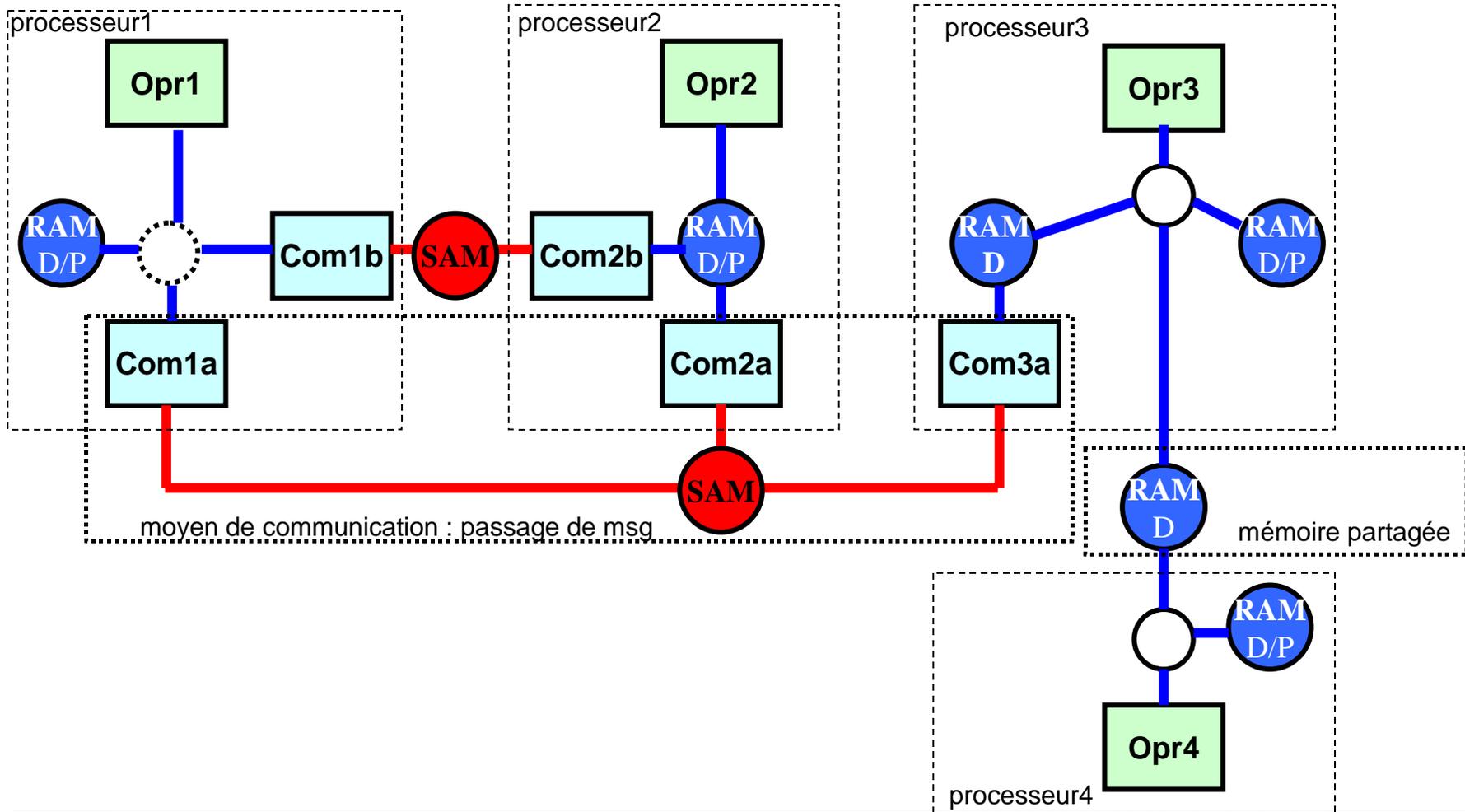
PRESENTATION DE SynDEX

Exemple d'algorithme : égaliseur adaptatif



PRESENTATION DE SynDEX

Exemple d'architecture

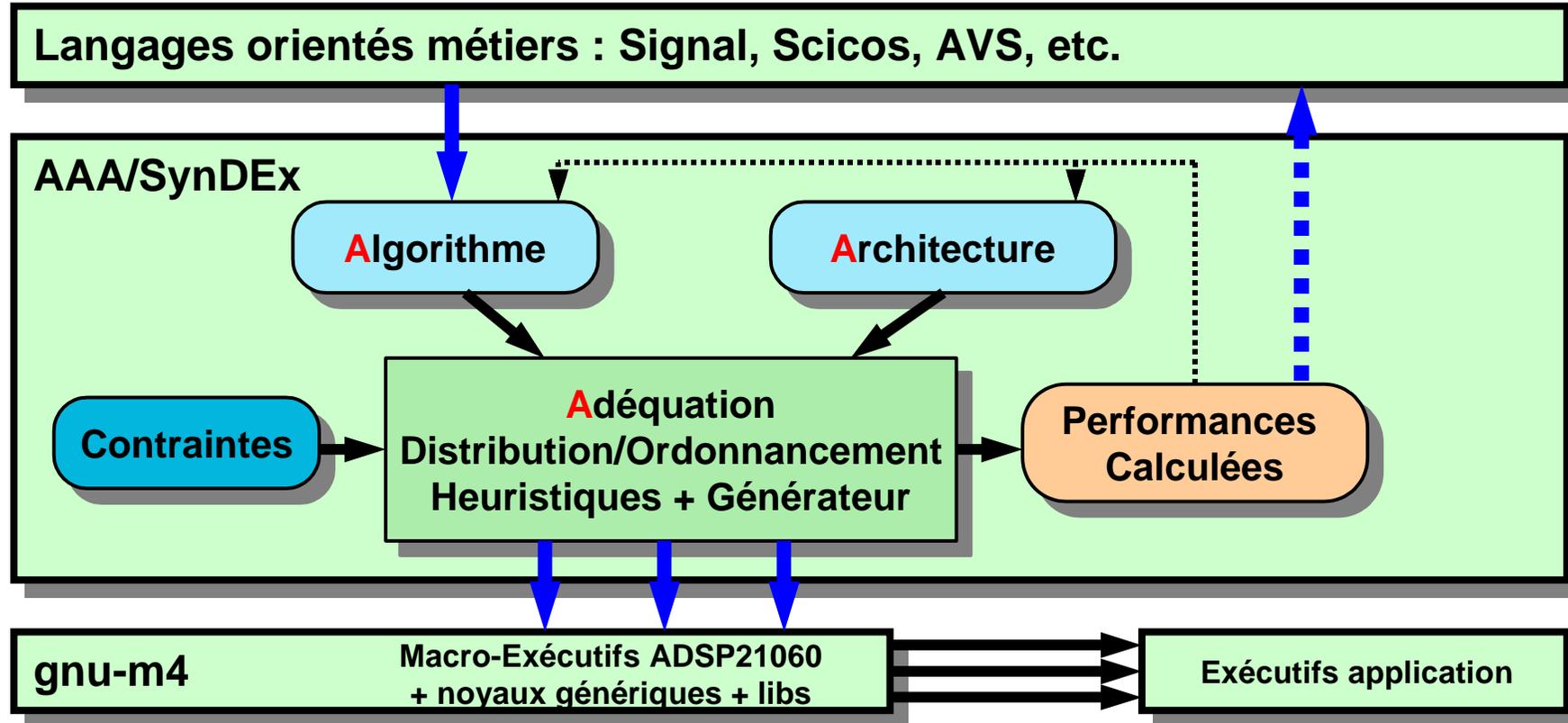


Optimisations

- **Caractérisation opérations/opérateurs**
 - Mesures : durée, mémoire, interférences calculs/com.
- **Choix d'une implantation parmi toutes**
 - Qui respecte les contraintes temps réel et de distribution
 - Qui minimise les ressources
- **Distribution/ordonnancement** : hors-ligne sans préemption
- **Problèmes NP-difficiles** : heuristiques
 - Rapides (prototypage) : gloutonne, ordonnancement liste

PRESENTATION DE SynDEX

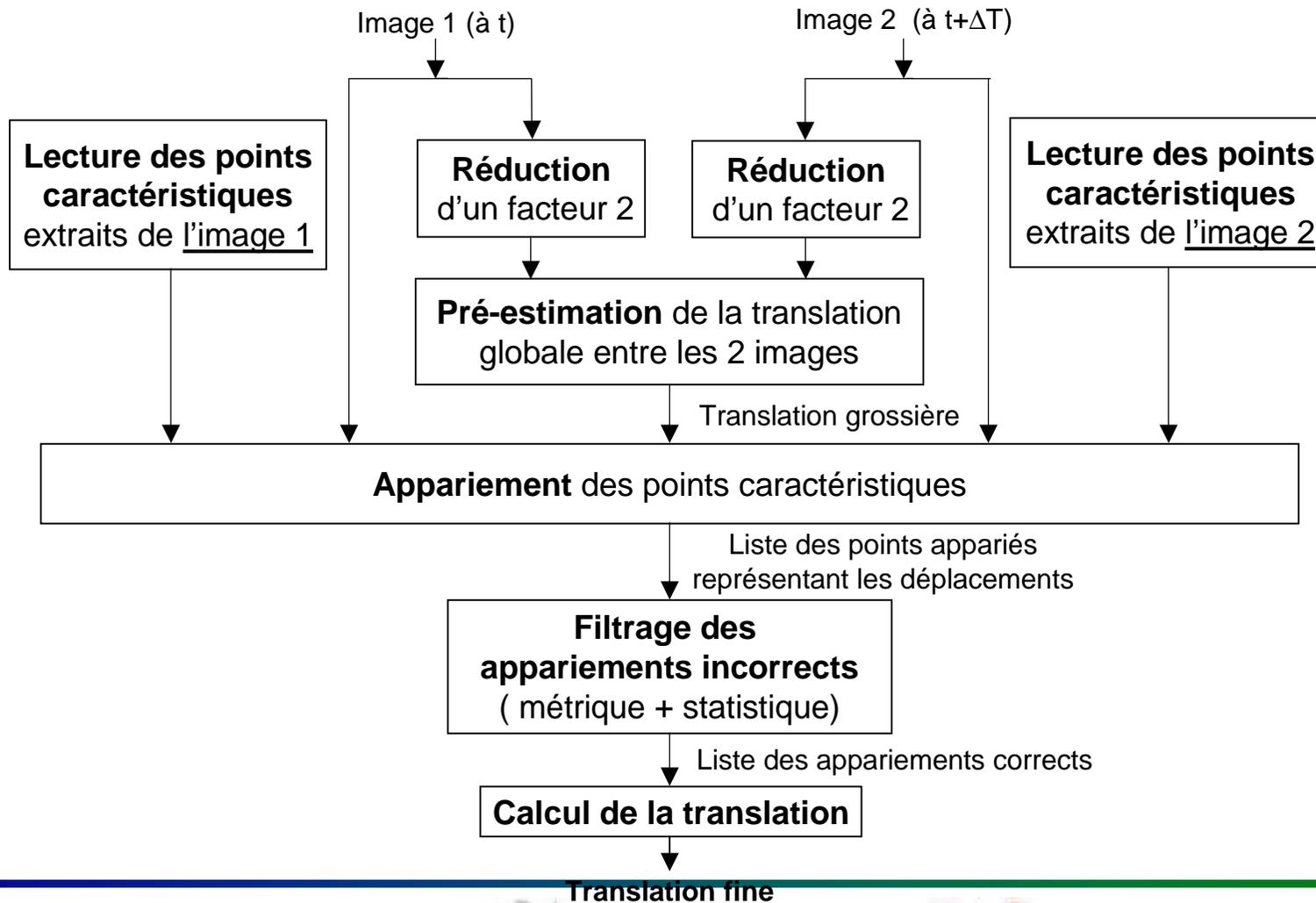
Implantation optimisée AAA/SynDEX



Intérêts

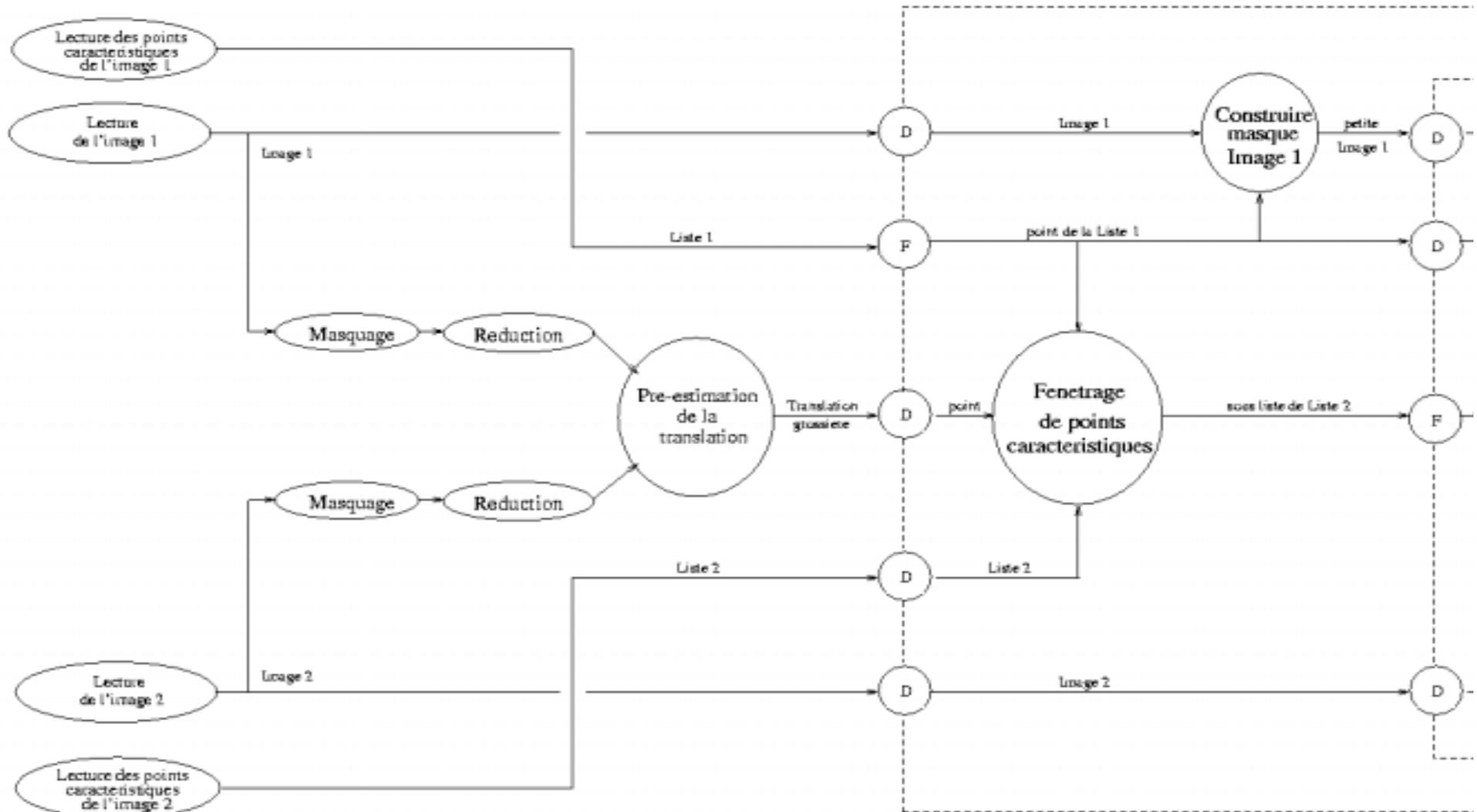
- **Langage de spécification** orienté métier + AAA/SynDEx = environnement cohérent de modélisation, simulation, implantation optimisée, sans rupture dans le cycle de développement
- **Prototypage rapide** : simulation temporelle, étude parallélisme potentiel, dimensionnement architecture
- **Génération automatique de code** conservant les propriétés de la spécification, code prototype = code de série
- **Cycle de développement très réduit**

Graphe faisant apparaître du parallélisme d'opérations



IMPLANTATION AVEC SynDEX V5

Graphe flots de données (utilisable avec SynDEX V6)



Limitations de SynDEx V5

- **SynDEx v5** ne fait pas apparaître de parallélisme de données (même opération répétée spatialement sur des données différentes)
 - Il faut exprimer explicitement les répétitions d'opérations (fastidieux)
 - Pour simplifier on encapsule dans une unique opération les opérations répétées.
- **Conditionnement**, non disponible.

IMPLANTATION AVEC SynDEx V5

Implantation d'une première partie de l'algorithme

The screenshot displays the SynDEx V5 interface. The main window shows a data flow graph with the following nodes and connections:

- Input nodes:** Lirelm1 (labeled 'lmaq'), HEIGHT (labeled 'lheight'), WIDTH (labeled 'lwidth').
- Intermediate nodes:** Mask1 (labeled '?lmaq', '?height', '?width'), Reduc1 (labeled '?lm', '?height', '?width').
- Output node:** Ecrirelm1 (labeled '?lmaqRed').
- Root node:** root LU.

The 'Schedule' panel on the right shows a vertical timeline starting at 0. The blocks are arranged as follows:

- Lirelm1:** Starts at 0 and ends at approximately 1/3 of the total duration.
- Mask1:** Starts at the end of Lirelm1 and ends at approximately 2/3 of the total duration.
- Reduc1:** Starts at the end of Mask1 and ends at approximately 3/4 of the total duration.
- Ecrirelm1:** Starts at the end of Reduc1 and ends at approximately 7/8 of the total duration.

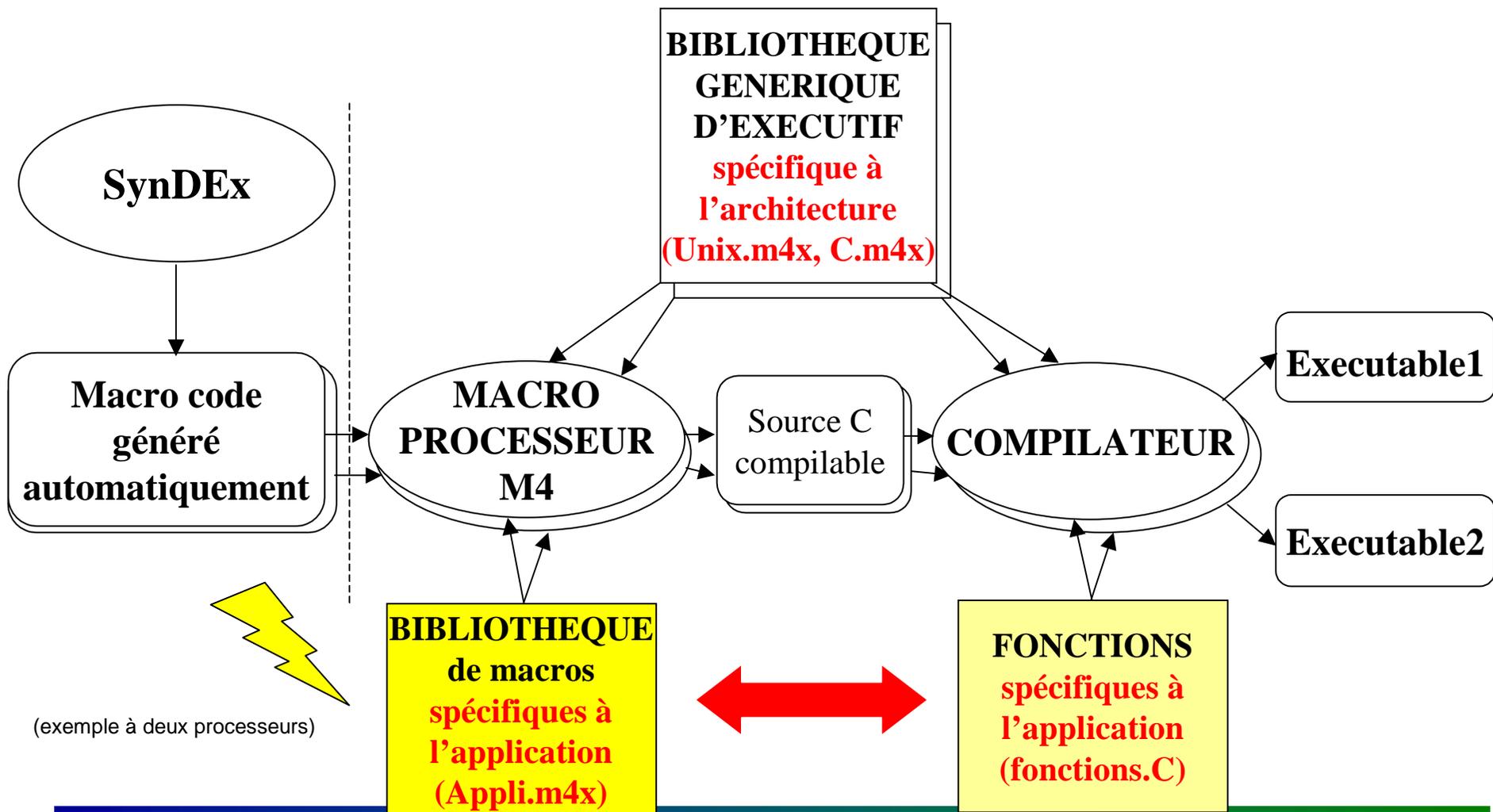
The bottom status bar shows: SynDEx: Select New or Open in File Menu, SynDEx>

Génération automatique de code produit des fichiers

- root.m4 (macrocode dédié à ce processeur).....

IMPLANTATION AVEC SynDEx V5

Chaîne de compilation



Association opération-SynDEx / fonction-C

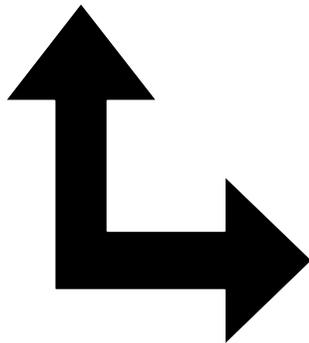
Fonction Reduction

Appli.m4x

divert

```
void reduction(int height, unsigned char *im, int width, unsigned short *imreduc);
divert(-1)
```

```
# function Reduc1 Reduction1 {uchar[65536] ?!m int ?height int ?width ushort[16384] !Imreduc}
def ( 'Reduction1', `ifelse(MGC,LOOP, `proto_(uchar[65536]?$1, int?$2, int?$3, ushort[16384]!$4)
    Ccall_(void, `reduction', int $2, uchar *$1, int $3, ushort *$4)')')
```



```
void reduction(int height, unsigned char *im, int width, unsigned short *imreduc)
{
    int i, j, k;
    for (j = 0; j < height; j++)
    {
        for (i = 0; i < width; i++)
        {
            k=(j >> 1)*(width >> 1)+(i >> 1);
            imreduc[k] += im[j*width+i];
        }
    }
}
```

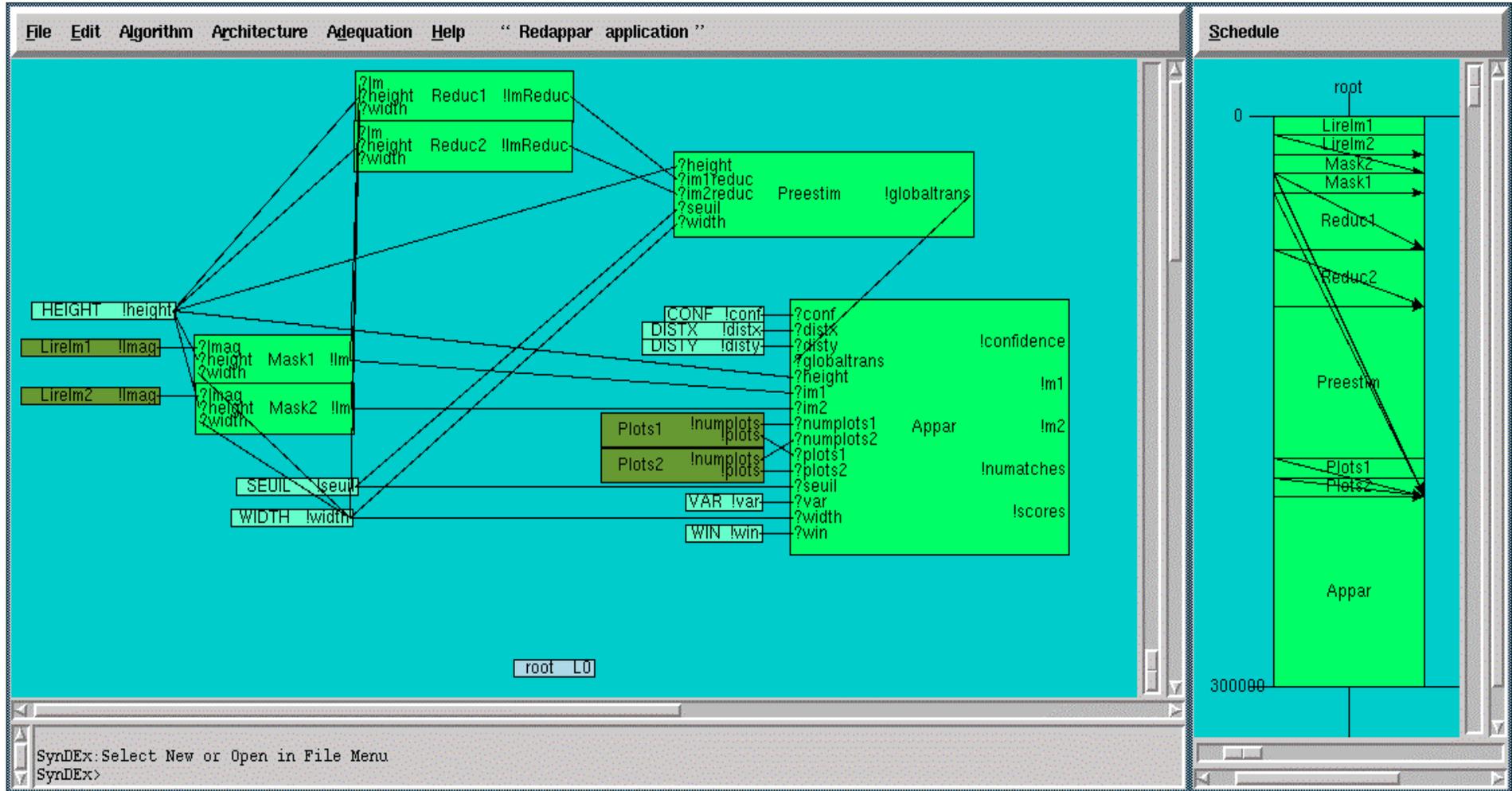
Fonction C

Tests

- **Modification** du programme C d'origine pour générer des fichiers intermédiaires
- **Validation** par comparaison du contenu des fichiers issus du programme d'origine, et ceux obtenus avec SynDEx
- **Chronométrage** automatique des durées d'exécution (sur station de travail)
- **Validation incrémentale**

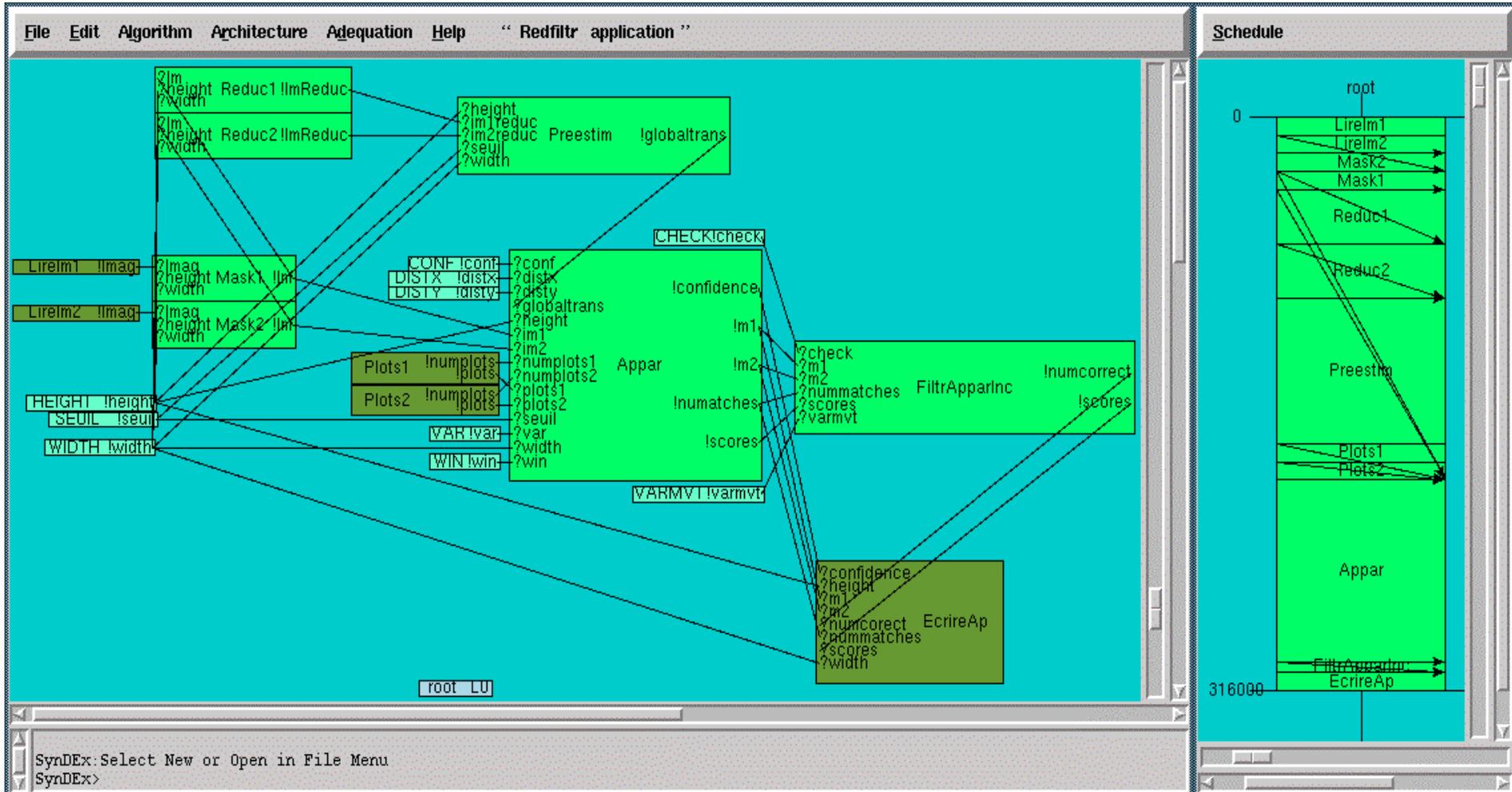
IMPLANTATION AVEC SynDEX V5

Validation incrémentale (3)



IMPLANTATION AVEC SynDEX V5

Validation incrémentale (4)



Chronométrage de l'implantation mono-processeur de la première partie de l'algorithme

Chronometric profiling:

label date dt=date-datePrec-chronoOverhead (microseconds)

1001 453705274 chronoOverhead=1

1002 453705802 527

1003 453708275 2472

1004 453710722 2446

1005 453715995 5272

1006 453720405 4409

1007 454201925 481519

1008 454204444 2518

1009 454206064 1619

1010 454864253 658188

1011 454867663 3409

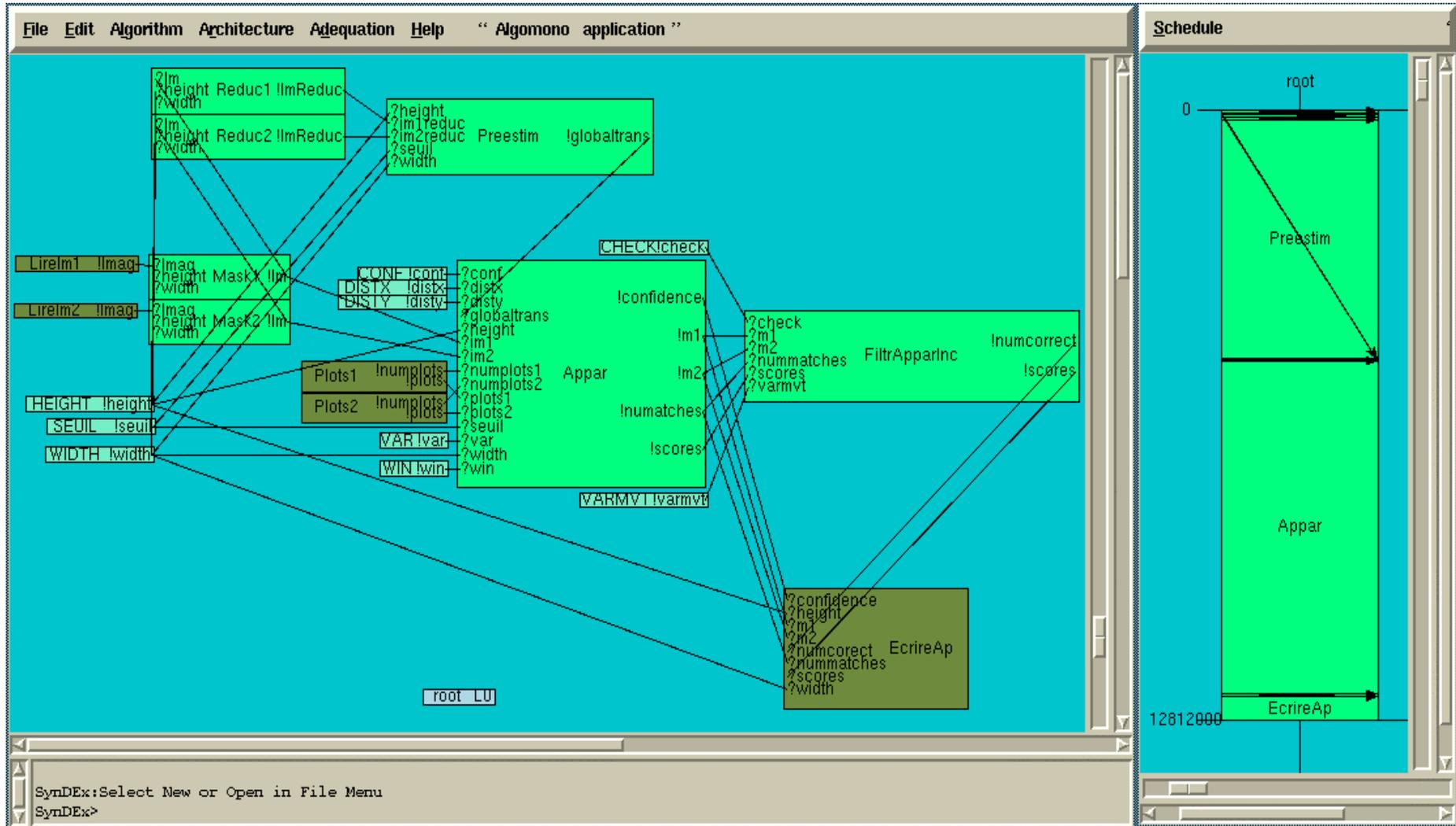
1012 454927254 59590

root.c:chrono	1001	=	root_LineIm1
root.c:chrono	1002	=	root_LineIm2
root.c:chrono	1003	=	root_Mask2
root.c:chrono	1004	=	root_Mask1
root.c:chrono	1005	=	root_Reduc1
root.c:chrono	1006	=	root_Reduc2
root.c:chrono	1007	=	root_Preestim
root.c:chrono	1008	=	root_Plots1
root.c:chrono	1009	=	root_Plots2
root.c:chrono	1010	=	root_Appar
root.c:chrono	1011	=	root_FiltrApparInc
root.c:chrono	1012	=	root_EcrireAp

Use ``grep "^chrono" *.c`` to collect labels number/name correspondences

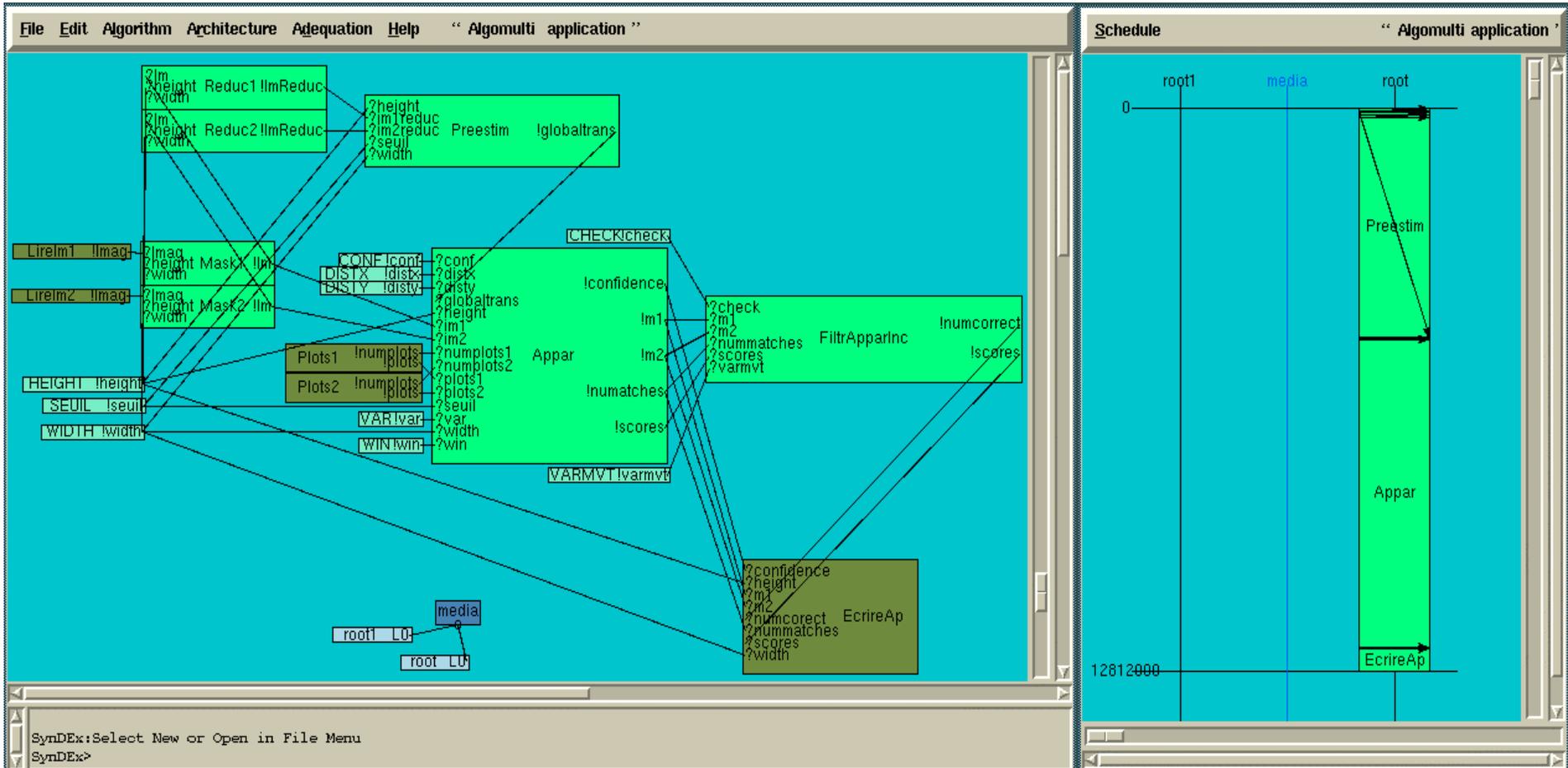
IMPLANTATION AVEC SynDEX V5

Simulation temporelle avec les durées chronométrées



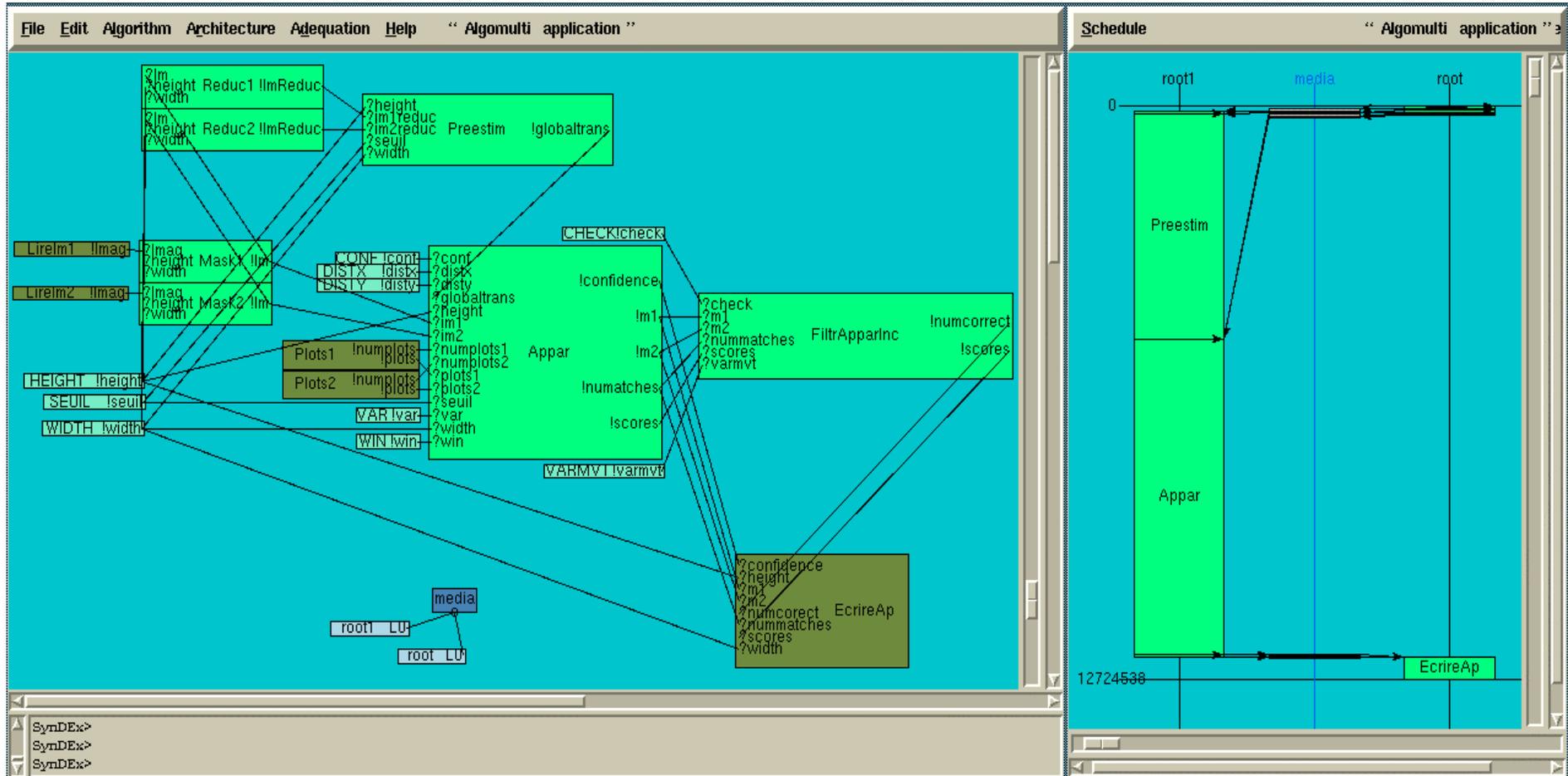
IMPLANTATION AVEC SynDEX V5

Simulation bi-processeur avec un média de communication lent



IMPLANTATION AVEC SynDEX V5

Simulation bi-processeur avec un média de communication 10x plus rapide



Conclusion de l'implantation avec SynDEx V5

- **Parallélisme d'opérations (de tâches) :** insuffisant pour diminuer la durée d'exécution en bi-processeur
 - Nécessité de spécifier du parallélisme de données au niveau des fonctions de pré-estimation ("Preestim") et d'appariement ("Appar")



Spécification avec SynDEx V6

SynDEx V6

- **Spécification hiérarchique**

Permet de représenter l'algorithme complet de façon plus claire

- **Factorisation**

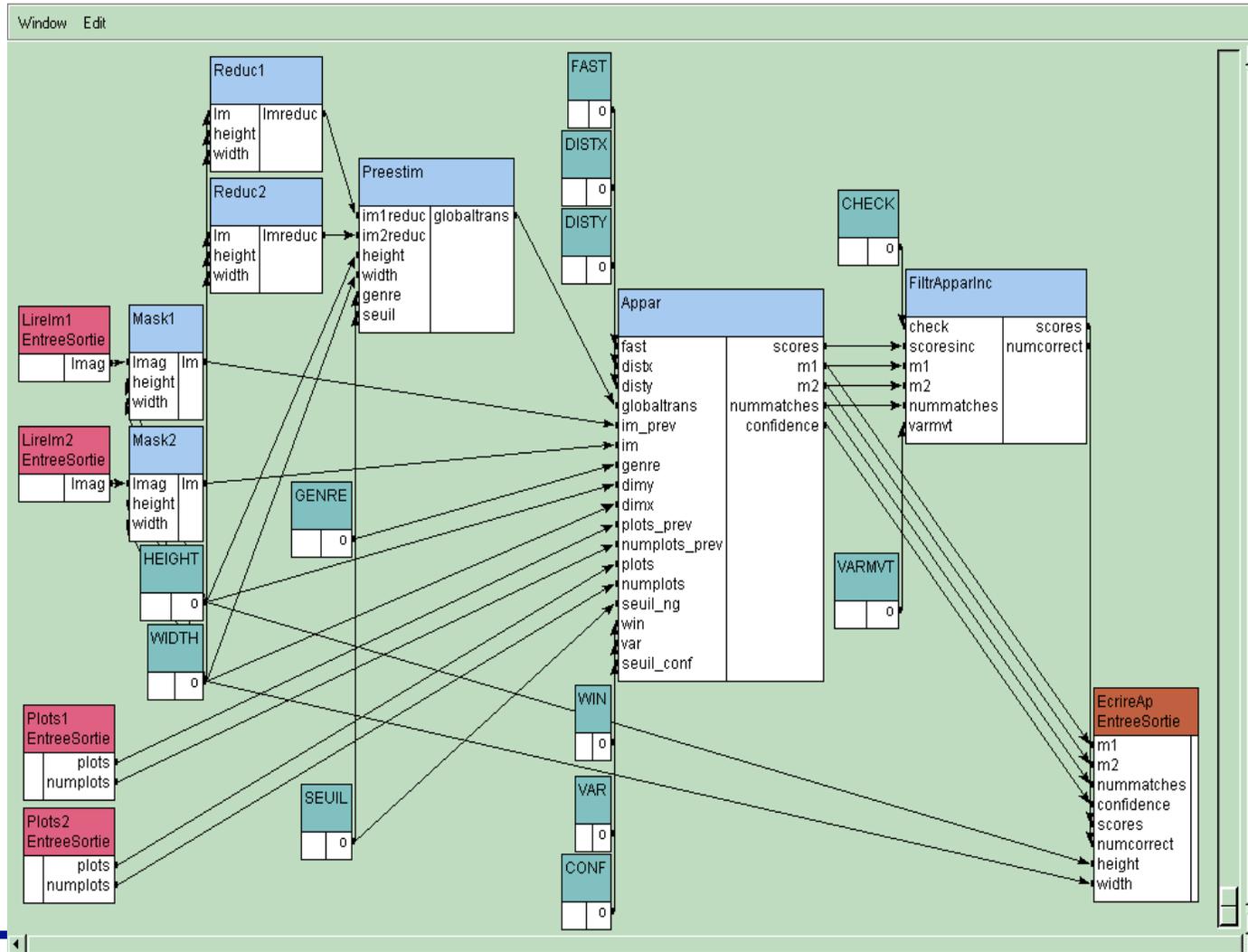
Mise en évidence du parallélisme de données (même opération répétée spatialement sur des données différentes)

- **Conditionnement par booléen et entier**

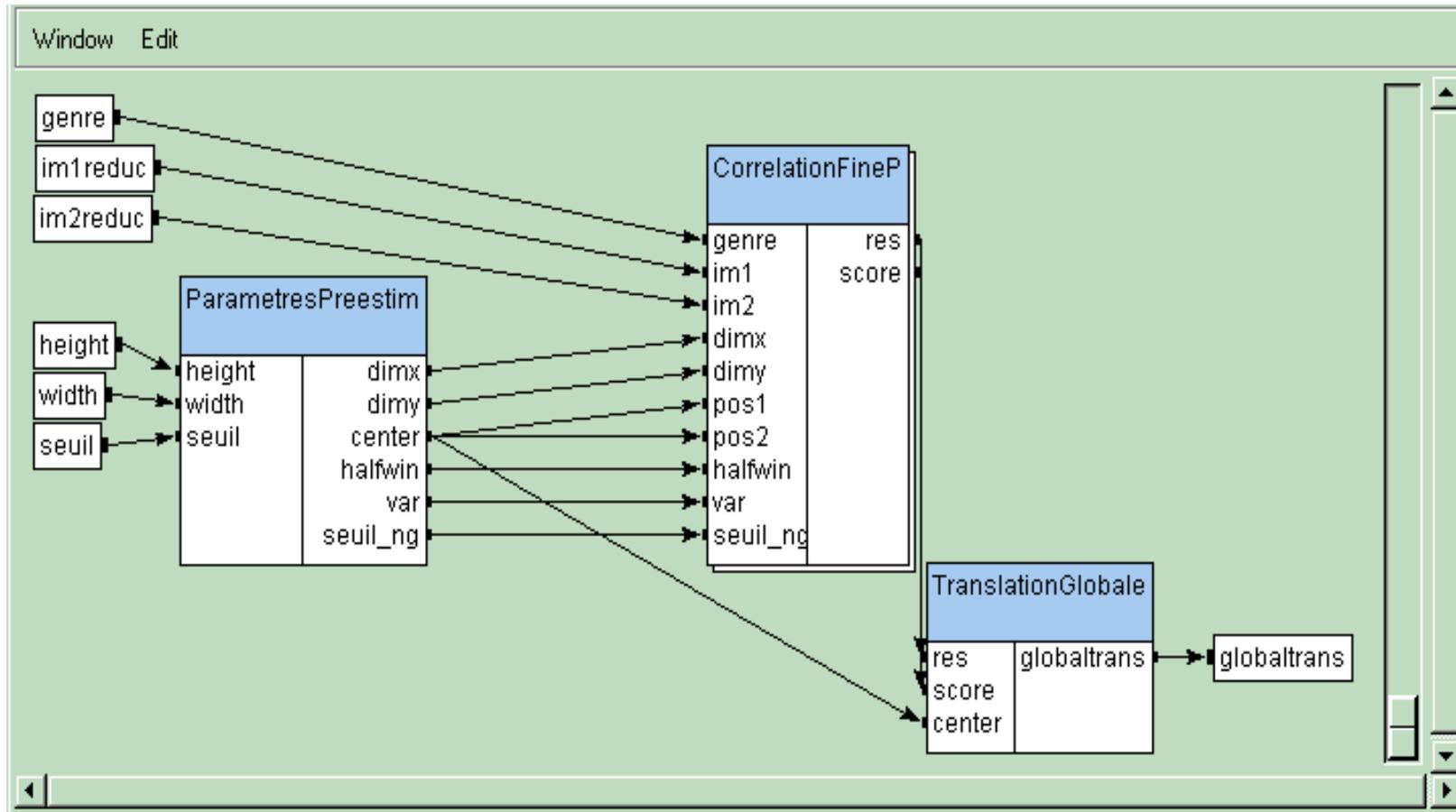
Prise en compte par l'heuristique, lors de l'optimisation, des cas où les opérations sont exécutées dans des cas exclusifs (jamais en même temps)

IMPLANTATION AVEC SynDEx V6

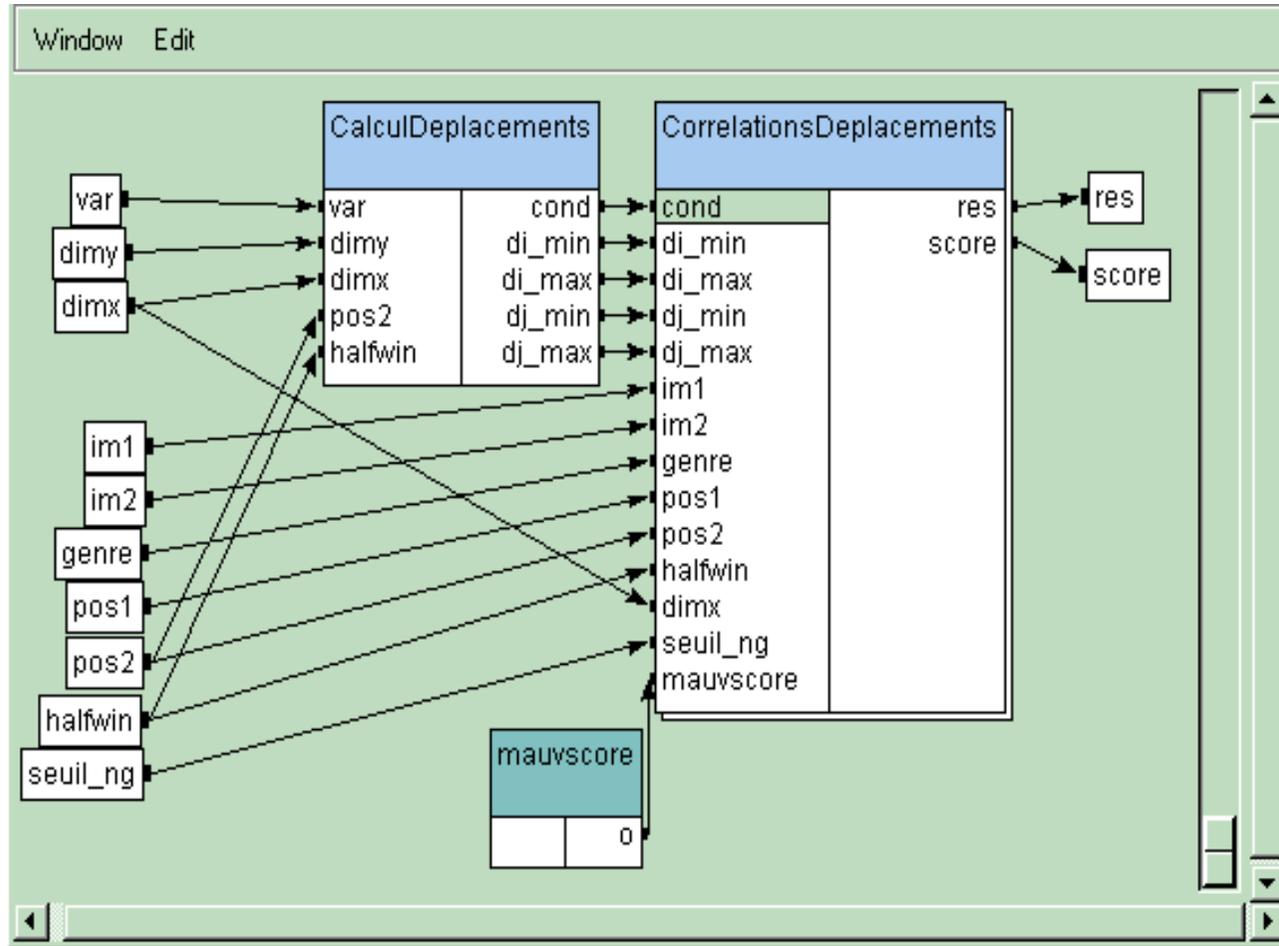
SynDEx V6 : Portage de l'implantation de l'algorithme complet spécifié en V5



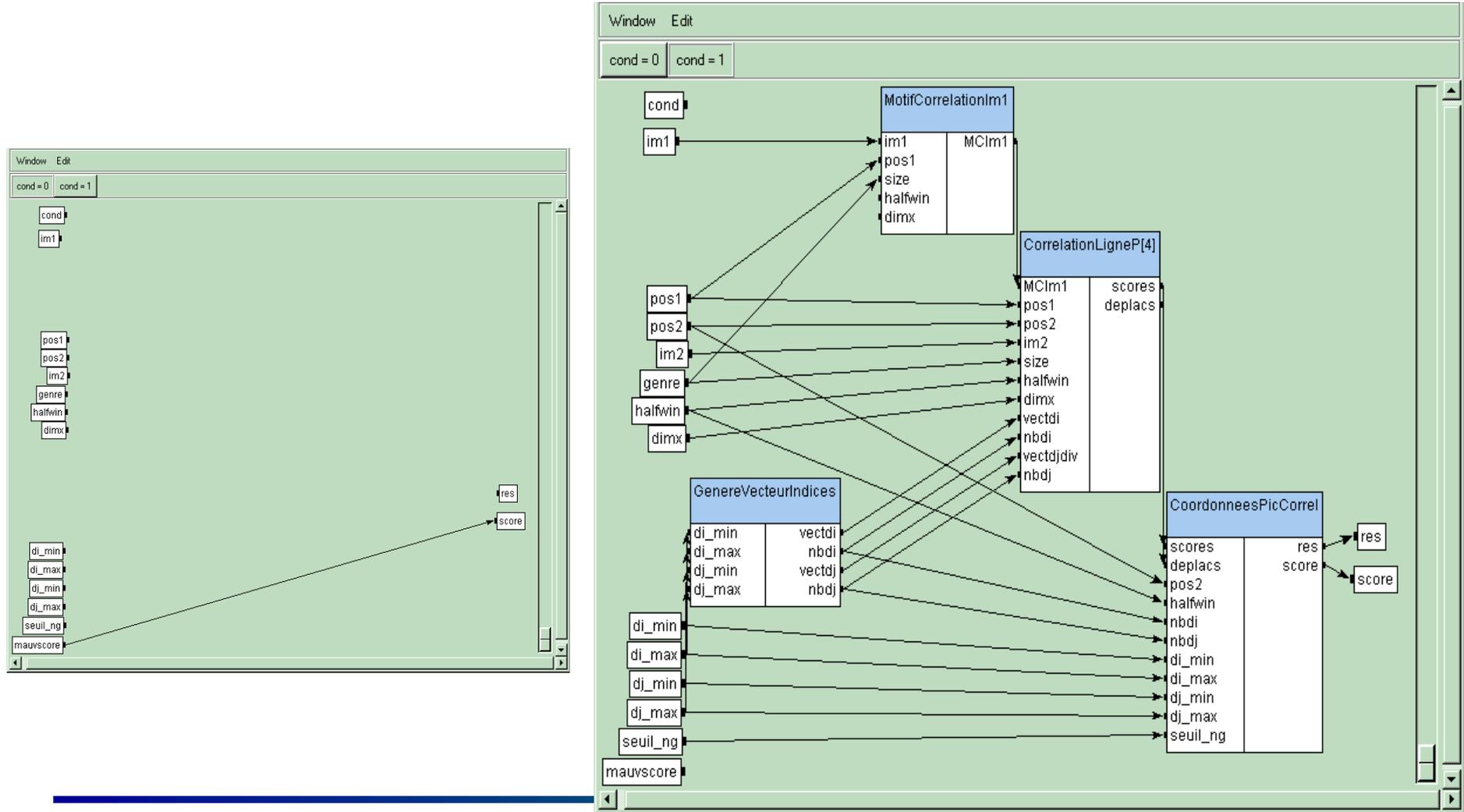
Spécification hiérarchique de l'opération « Preestim »



Spécification hiérarchique de l'opération « CorrelationFineP » : Preestim (2)

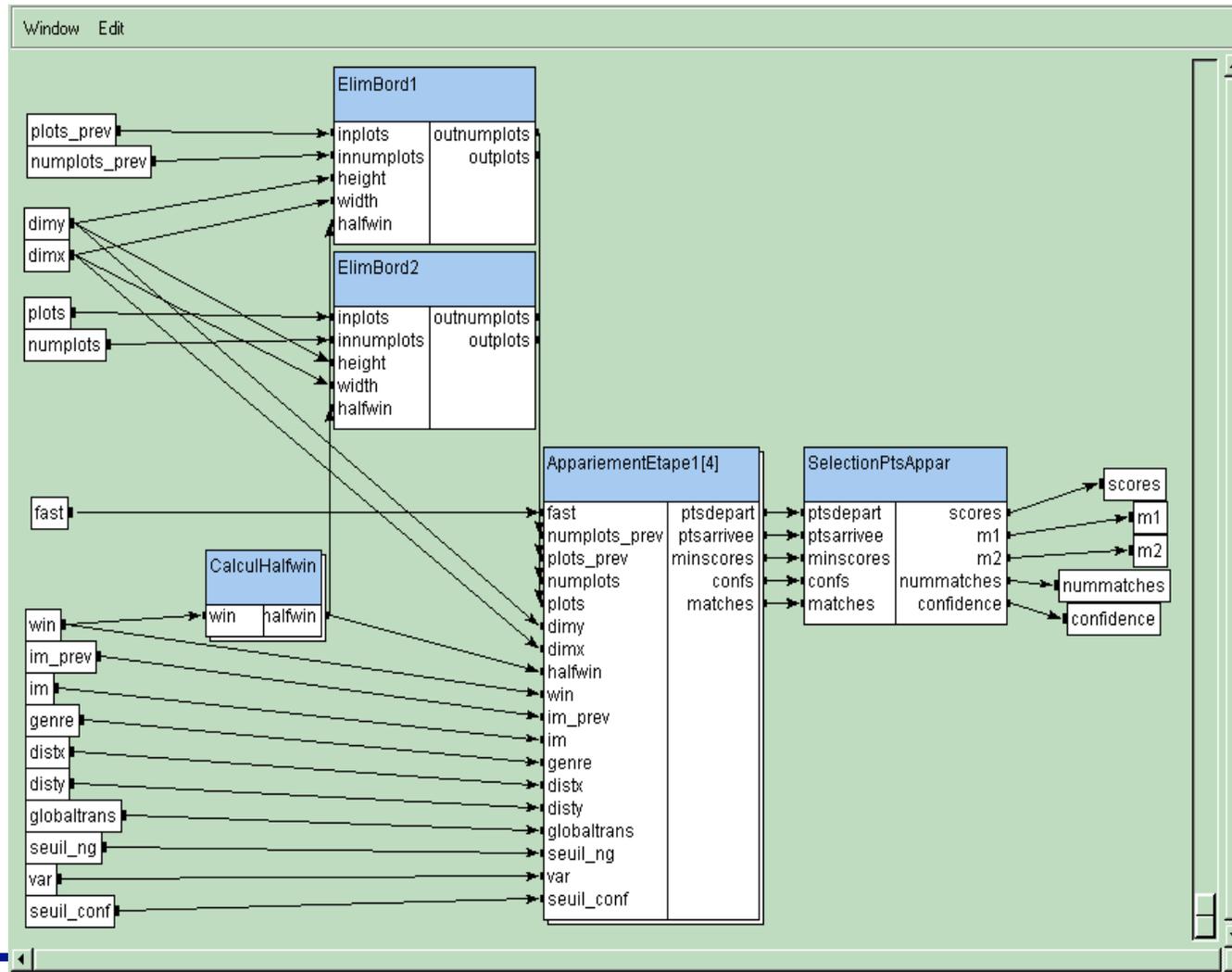


Spécification hiérarchique de l'opération conditionnée « CorrelationsDéplacements »

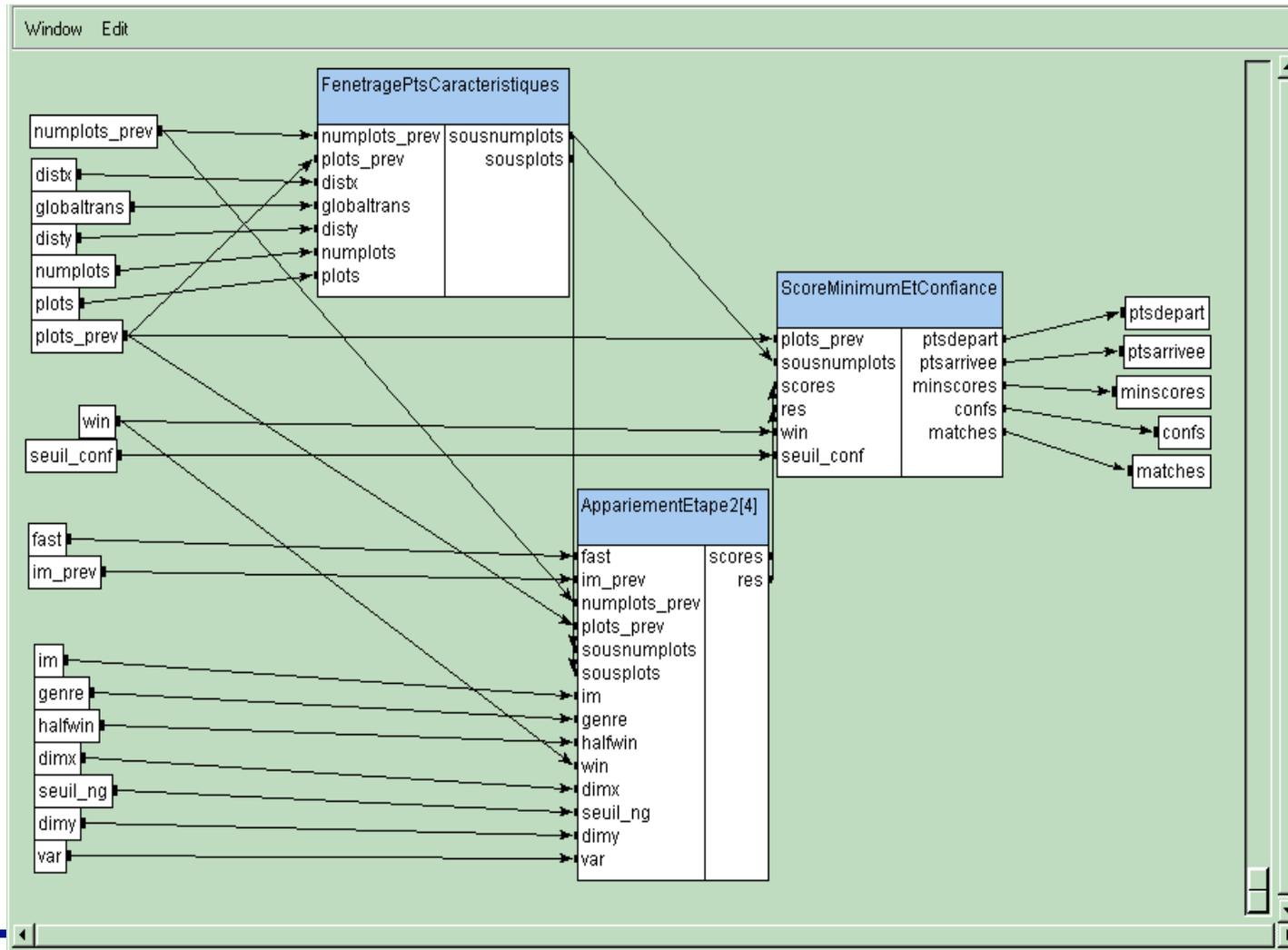


IMPLANTATION AVEC SynDEX V6

Spécification hiérarchique de l'opération « Appar »



Spécification hiérarchique de l'opération « AppariementEtape1 » : Appar (2)



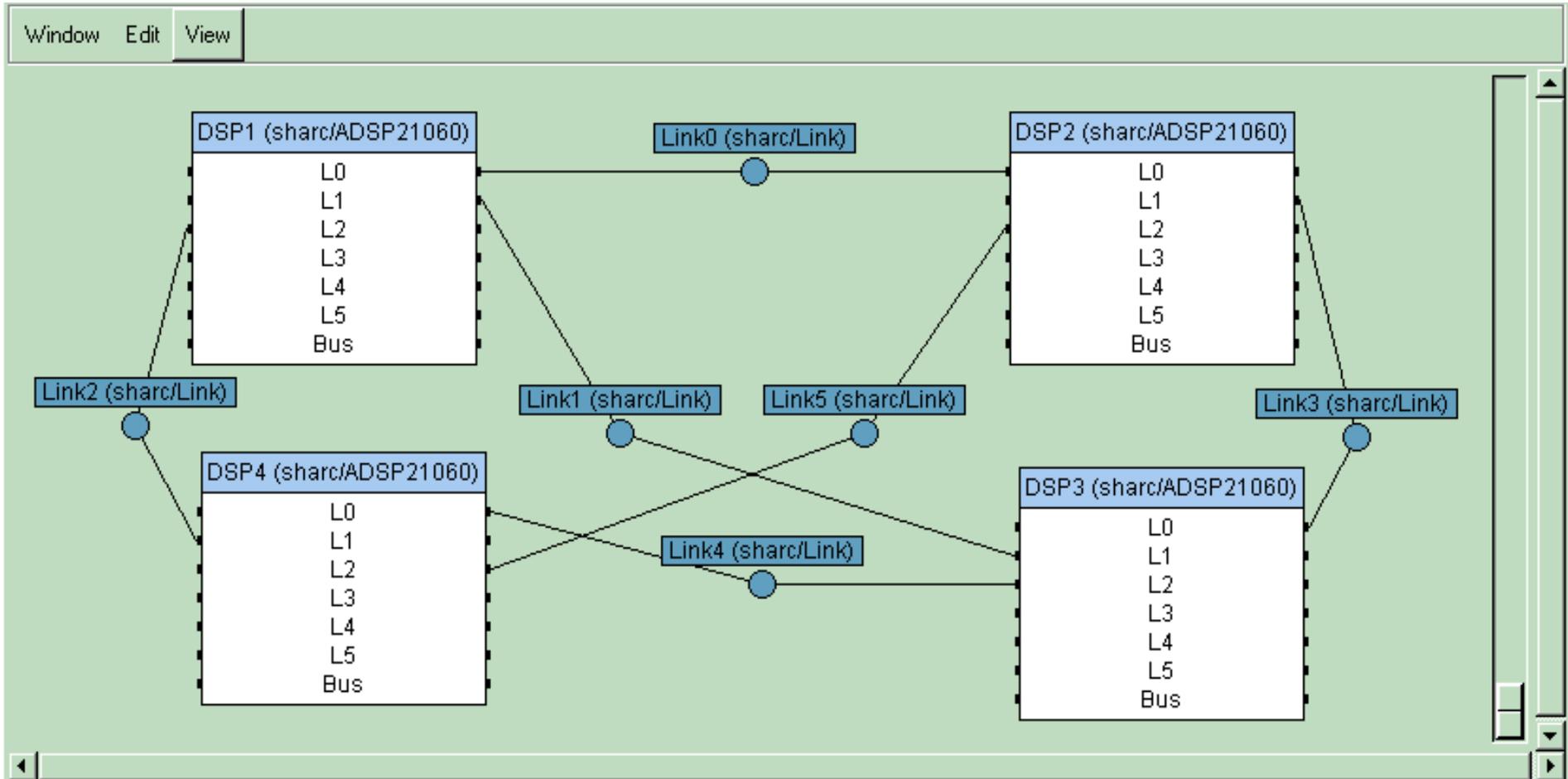
Exploration des implantations de l'algorithme sur différentes architectures

Une **architecture** est définie par son nombre de processeurs et par la topologie d'interconnexion entre ces processeurs

- Topologie **point à point en anneau**
(test sur 1 à 8 processeurs)
- Topologie **complètement interconnecté**
(test sur 1 à 6 processeurs)
- Topologie **point à point en anneau, et un bus global partagé** (test sur 1 à 8 processeurs)

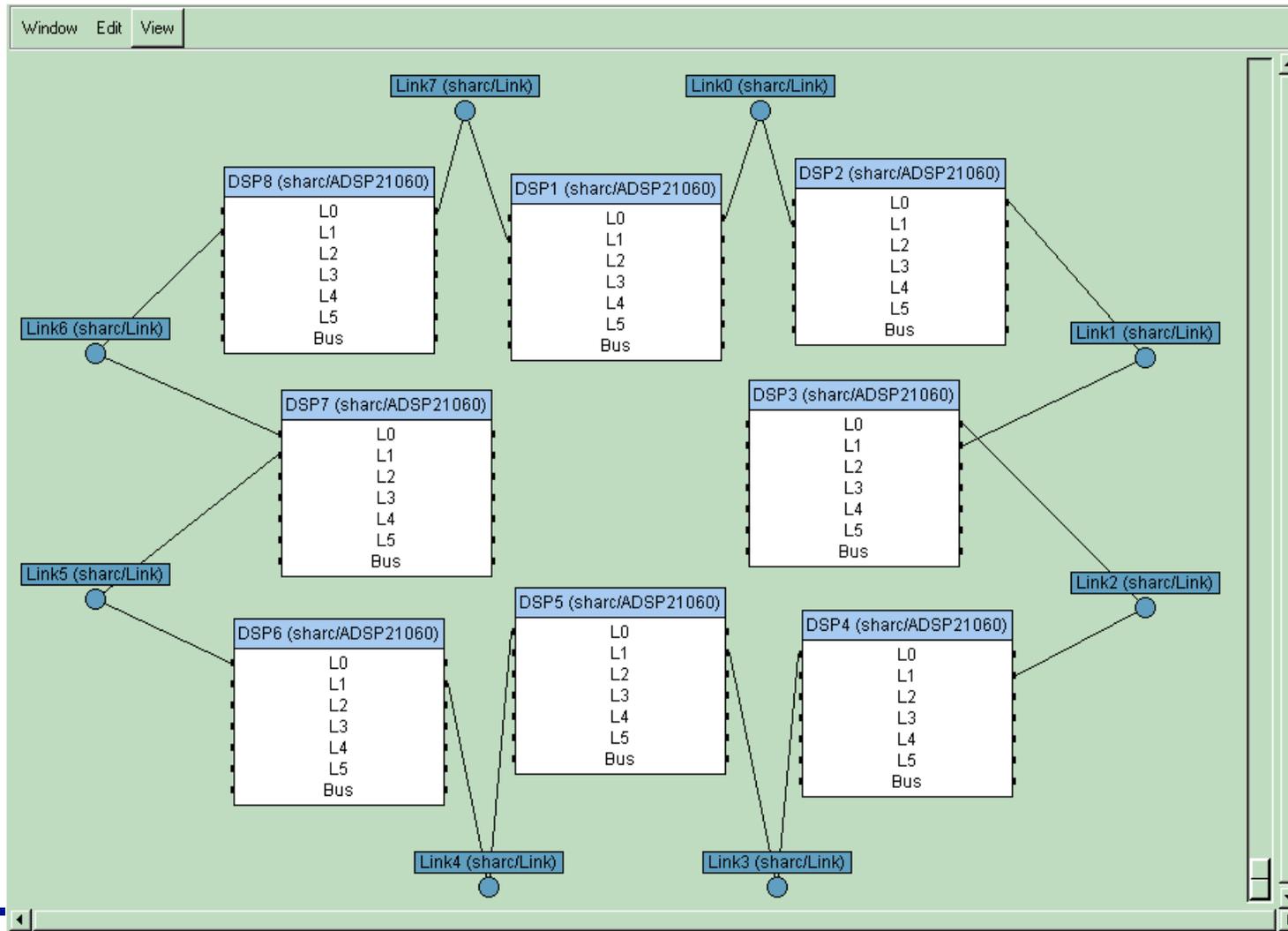
IMPLANTATION AVEC SynDEx V6

Architecture à 4 processeurs, topologie point à point complètement interconnecté

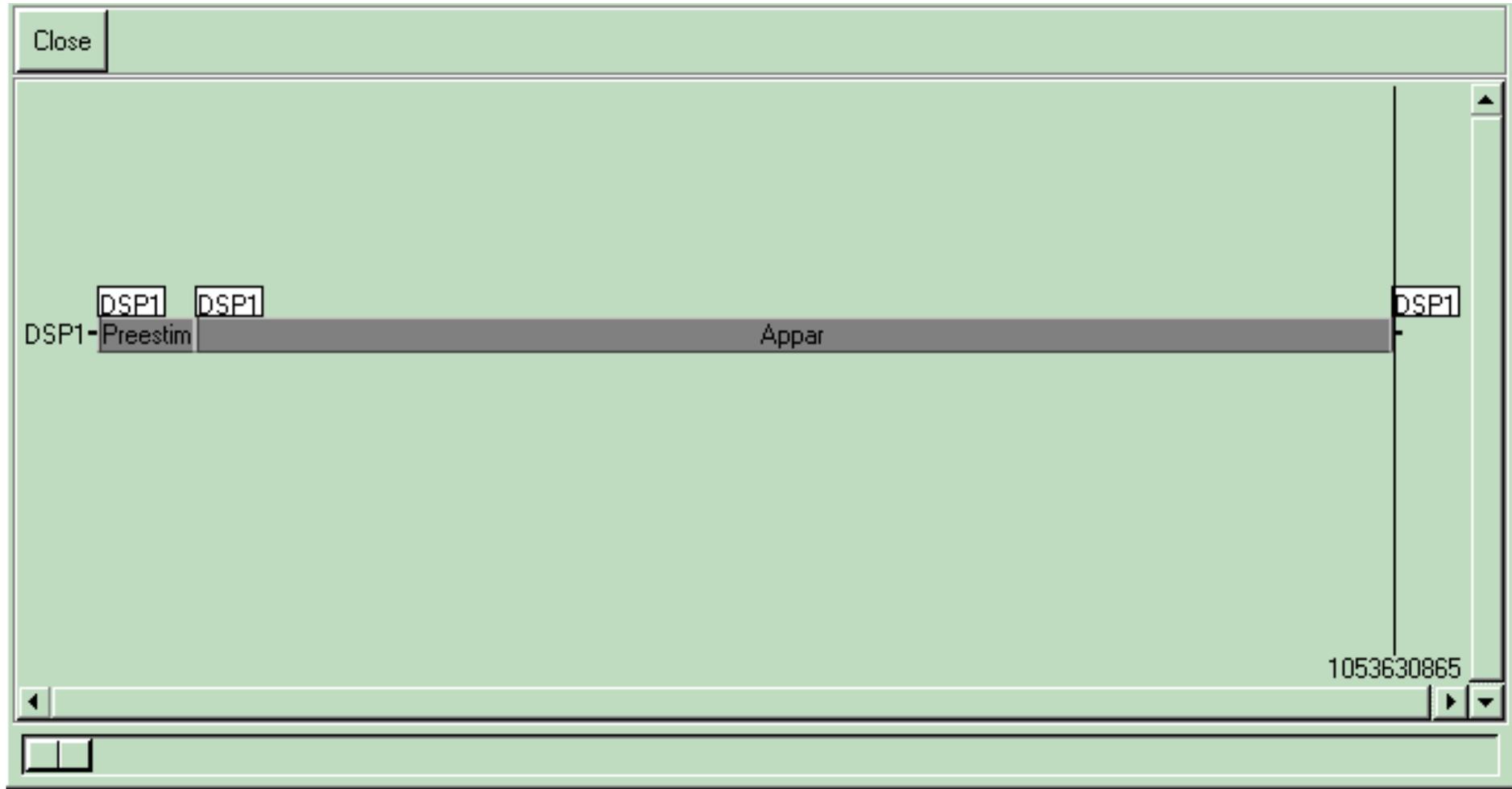


IMPLANTATION AVEC SynDEx V6

Architecture à 8 processeurs, topologie point à point en anneau



Ordonnancement de l'algorithme sur l'architecture mono-processeur



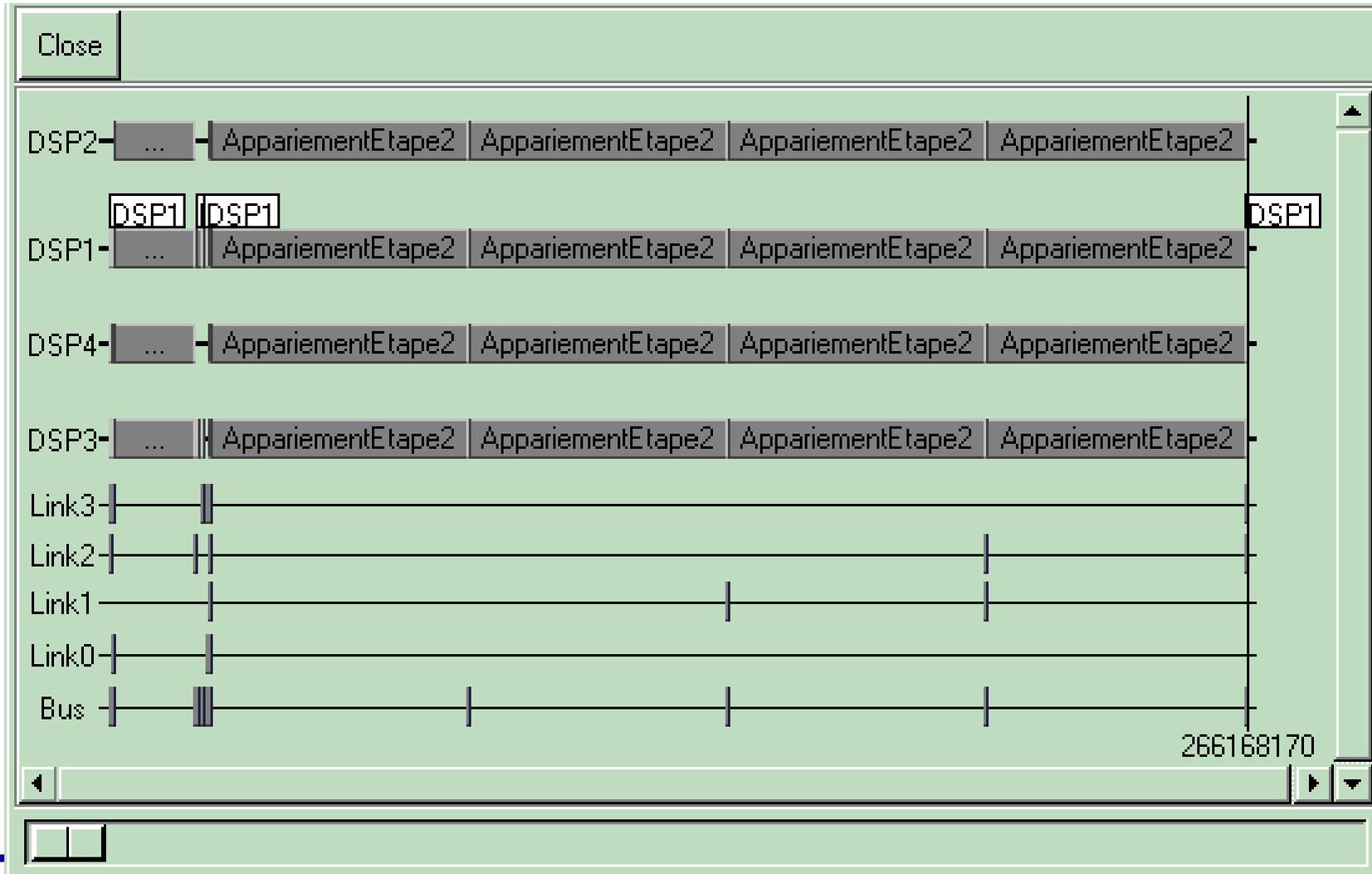
IMPLANTATION AVEC SynDEx V6

Distribution/ordonnancement de l'algorithme sur l'architecture bi-processeur



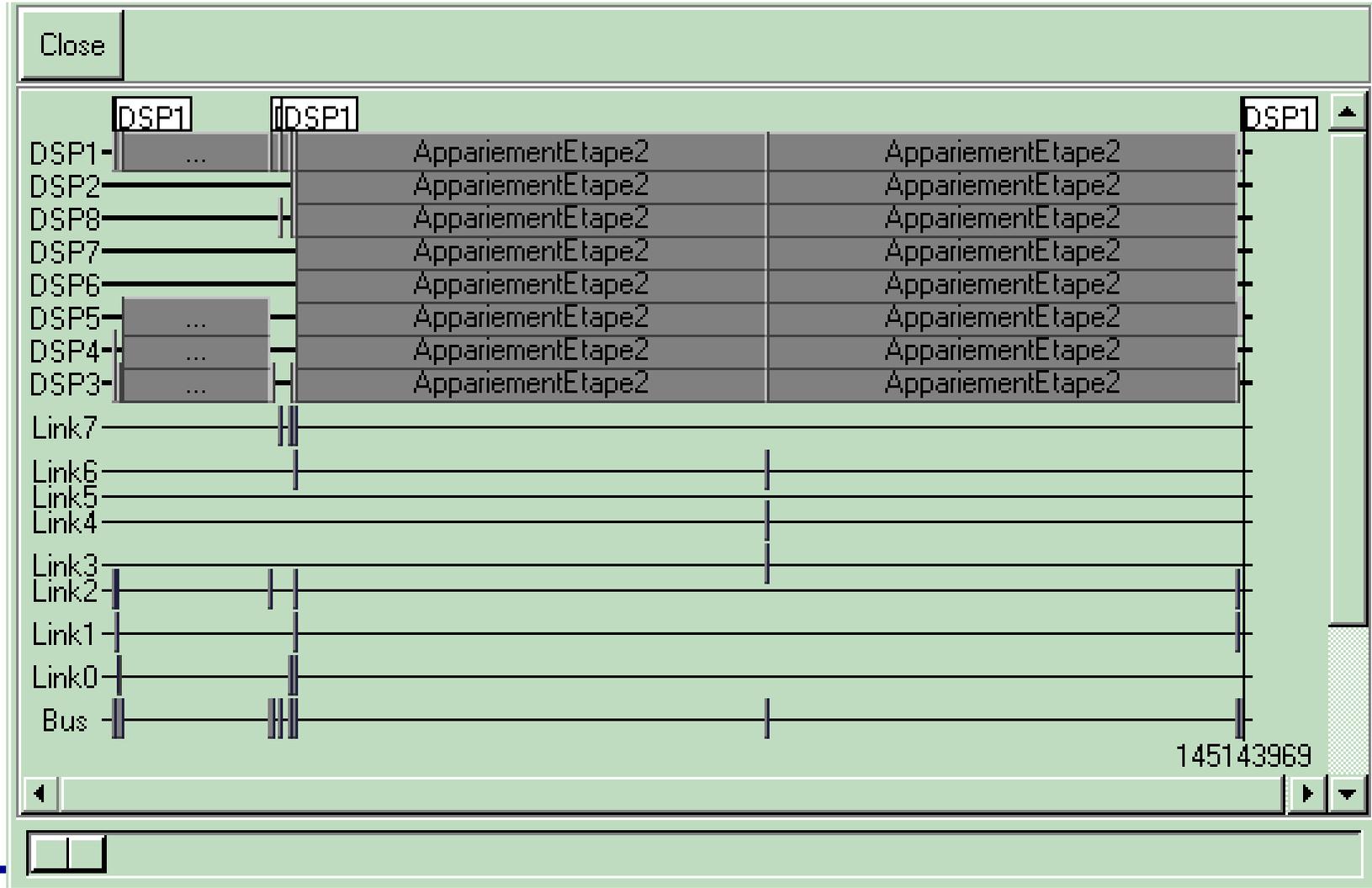
IMPLANTATION AVEC SynDEx V6

D/o de l'algorithme sur une architecture à 4 processeurs

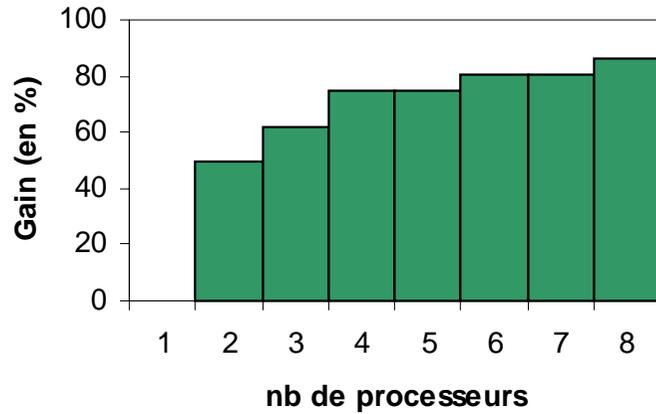


IMPLANTATION AVEC SynDEx V6

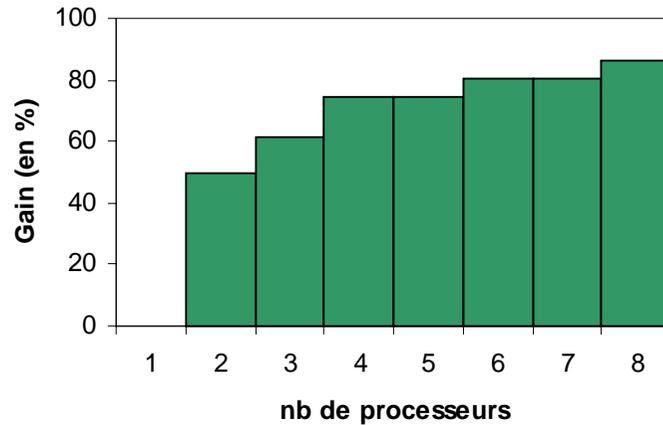
D/o de l'algorithme sur une architecture à 8 processeurs



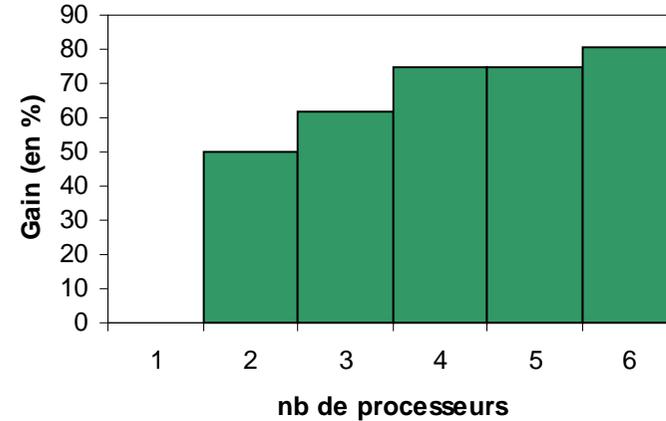
Analyse des résultats



Topologie point à point en anneau



**Topologie point à point en anneau
et bus global partagé**



**Topologie point à point
complètement interconnecté**

Stage EADS & INRIA : Kamel MEZIANE

SynDEx permet de:

- prédire les performances d'une implantation d'algorithme sur différentes architectures même si nous ne les possédons pas réellement
- simuler différentes implantations des variantes d'un algorithme sur différentes architectures afin de rechercher la plus adaptée
- générer du code pour une architecture choisie
- prototyper rapidement et effectuer une implantation temps réel optimisée, des applications de traitement d'images

SynDEx repose sur une méthodologie de développement :

- Spécification de l'application dans le formalisme SynDEx
- Simulation/exécution en mono-processeur sur station de travail
- Chronométrage des durées d'exécution des opérations
- Simulation en multi-processeur



- Raffinement de la spécification
- Chronométrage des durées d'exécution des opérations
- Simulation en multi-processeur



AAA optimisée