Vers une implantation générique validée expérimentalement des domaines abstraits?

Duong NGUYEN, CRI/ENSMP

APRON - Kickoff Meeting

Paris 28 Octobre 2004

Contexte

Analyse statique par abstraction de domaines

Preuve automatique des propriétés de programmes

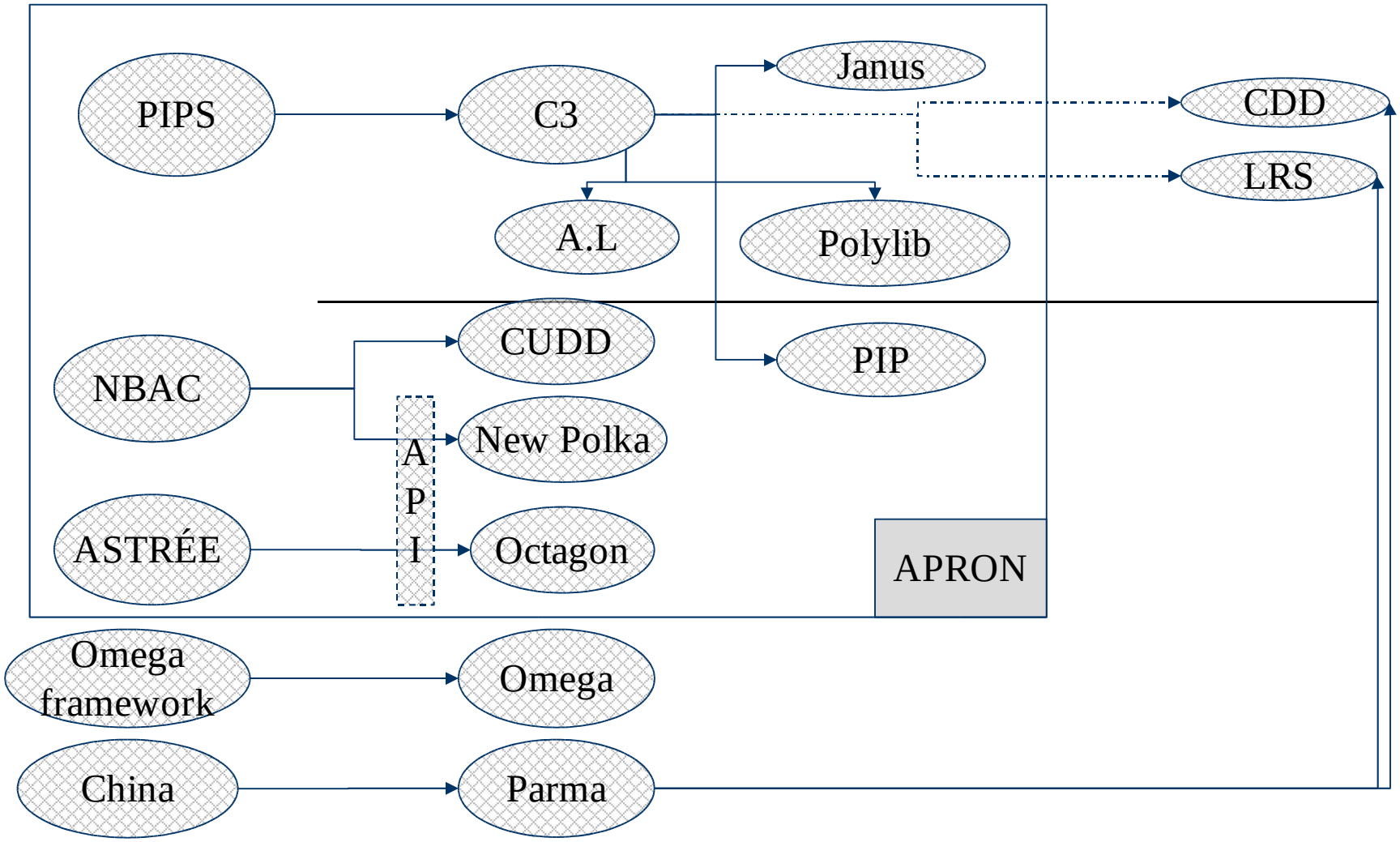
Validité des transformations et de la génération de code de programmes

PIPS

- Paralléliseur Interprocédural de Programmes Scientifiques (http://www.cri.ensmp.fr/pips)
- Exceptions en temps et en magnitude!!!

Projet APRON

Bibliothèques de domaines abstraits



Plan

Motivation

Spécification et implantation (API)

Évaluation des performances

Conclusion

Vers une implantation générique

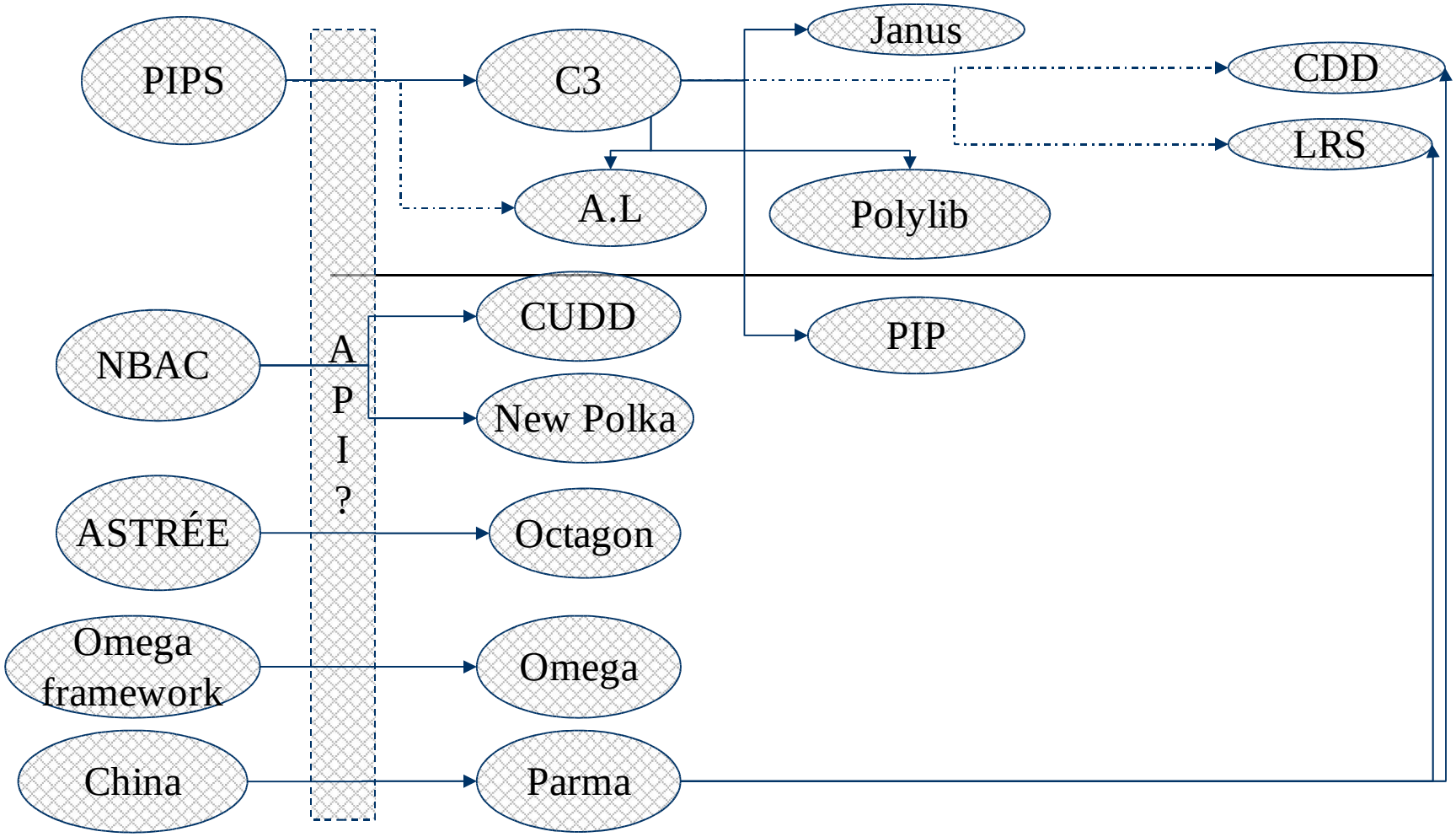
Les domaines abstraits:

- Modèle relationnel (polyèdre, liste de polyèdres, octagone, formule de Presburger)
- Modèle non-relationnel (intervalle)
- BDD (Diagramme de Décision Binaire)
- Chaîne de récurrences

Objectifs, motivation (CRI/APRON)

- Robustesse, rapidité, précision, paramétrisation
- Réutilisation de travaux existants
- Évaluation des résultats
- Interface commune (API)
- Les opérateurs ensemblistes

Interface commune (API)



Exemple d'API

Method Summary	
HQBoolean	**getStatus**(HQSetInterface.HQSetStatus status) Test the status of HQSet: CONSTANT_EMPTY, CONSTANT_UNIVERSE, UNDEFINED, NOT_EMPTY, BOUNDED, BOUNDED_FROM_ABOVE, BOUNDED_FROM_BELOW, CLOSED.
HQSet	**intersectWith**(HQSet set) Intersect current HQSet with given HQSet.
void	**projectDimension**(HQVariable var) Project a HQVariable dimension on the current HQSet.
HQSet	**unionWith**(HQSet set) Union current HQSet with given HQSet.

Problèmes déjà rencontrés

Interfaces existantes
- Manque d'opérateurs
- Niveau d'opérateurs

Domaine fixé pour chaque interface: Niveau de l'abstraction

Gestion de ~~l'environnement~~

Modes différents de gestion d'exceptions
- overflow, timeout
- Espace mémoire -> stop

Différence de précision: 32-bit, 64-bit, 128-bit, gmp

Langages de programmation utilisés: C, C++, Ocaml

En cours de développement
- Les interfaces
- Les opérateurs

Exemple avec PIPS

Problèmes en temps, en espace mémoire et en magnitude

- Configuration de la machine: PC 2.4Ghz, RAM: 2,048 Mo
- Programme ocean.f (4373 LOC), benchmark PERFECT Club
- 11 916 appels en total du test de faisabilité
- 424 overflows
- 3521 appels dépassent 3 secondes chacun
- Système de contraintes de 906530 inegalités

Librairies polyériques:

- Plusieurs implantations existantes (14 recencés)
- Réutilisation, performance, amélioration, heuristique

- Nécessité d'une évaluation

Évaluation de performance

Choix du « meilleur » algorithme et de la « meilleure » implantation

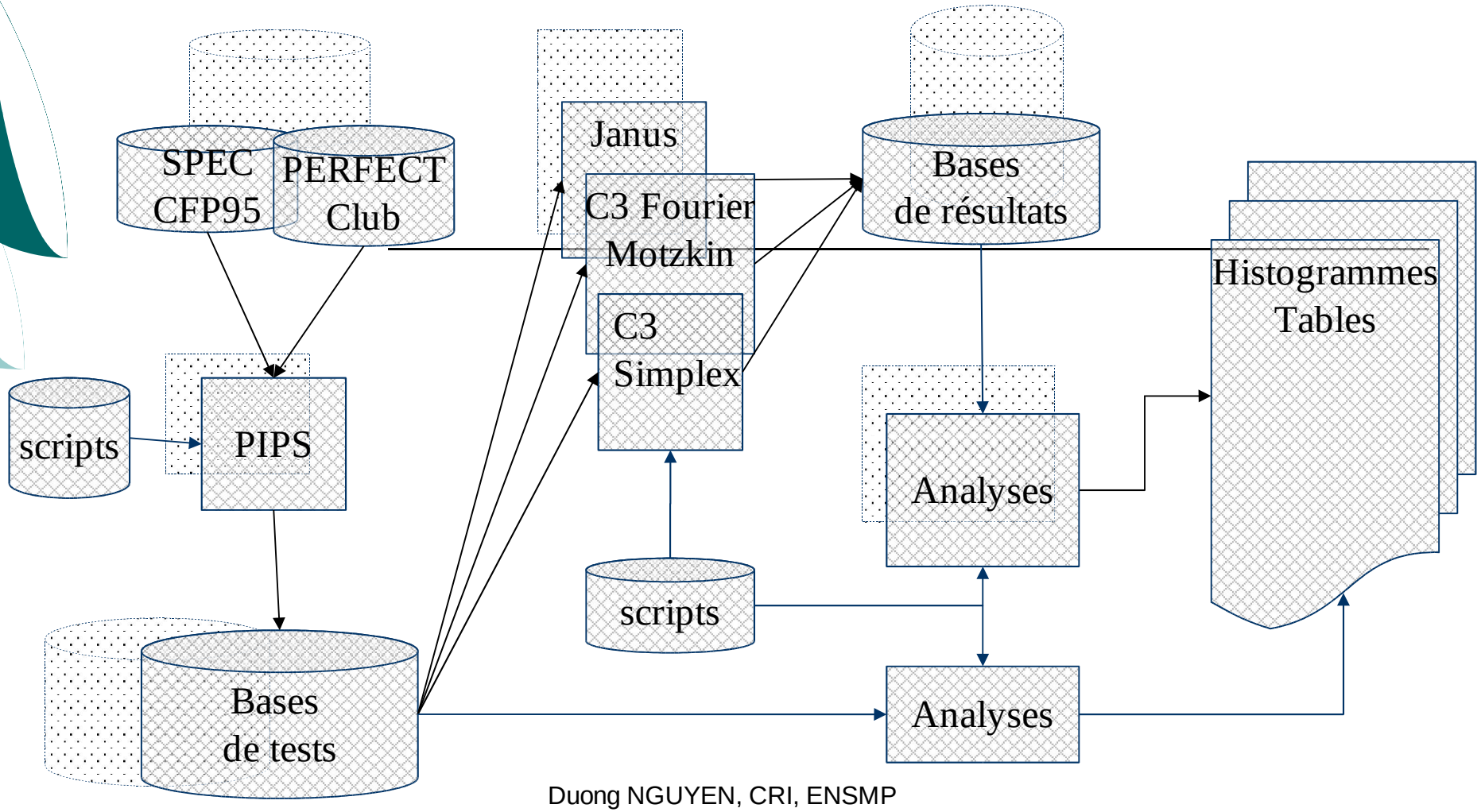
Gestion des heuristiques:

- Quel algorithme, quelle implantation
- Proposition de timeout
- Proposition de critères de choix

Aide à l'amélioration possible (algorithme ou implantation)

Gestion d'exceptions

Constitution des benchmarks



Constitution des bases de tests

CPU Benchmarks analysés avec PIPS:

- PERFECT Club:13 programmes en Fortran
- SPEC CFP 95: 10 programmes en Fortran

Condition d'analyse (PIPS):

- PRECONDITION inter-procedural
- TRANSFORMER inter-procedural
- REGION must inter-procedural

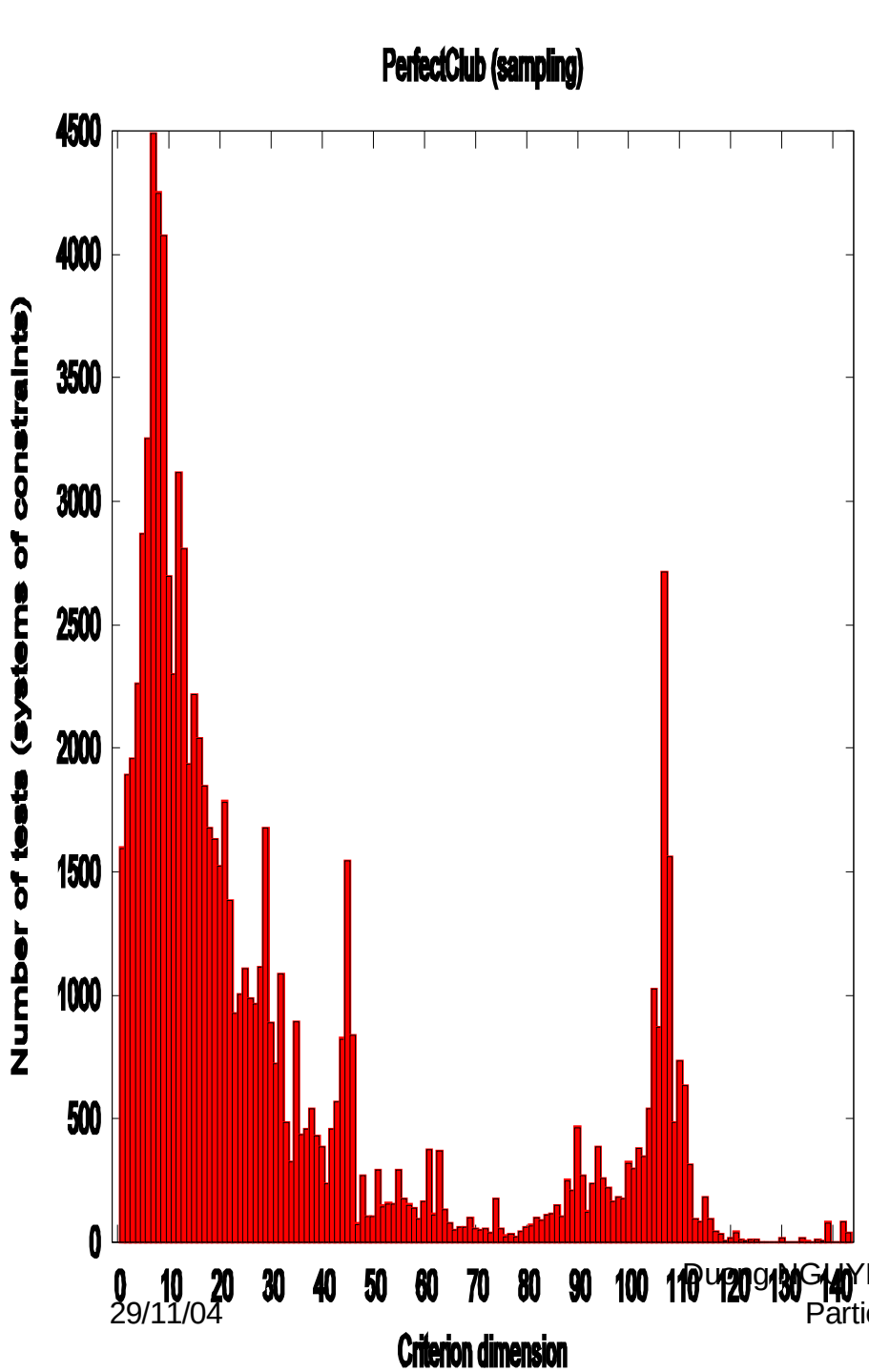
Systèmes d'exploitation:

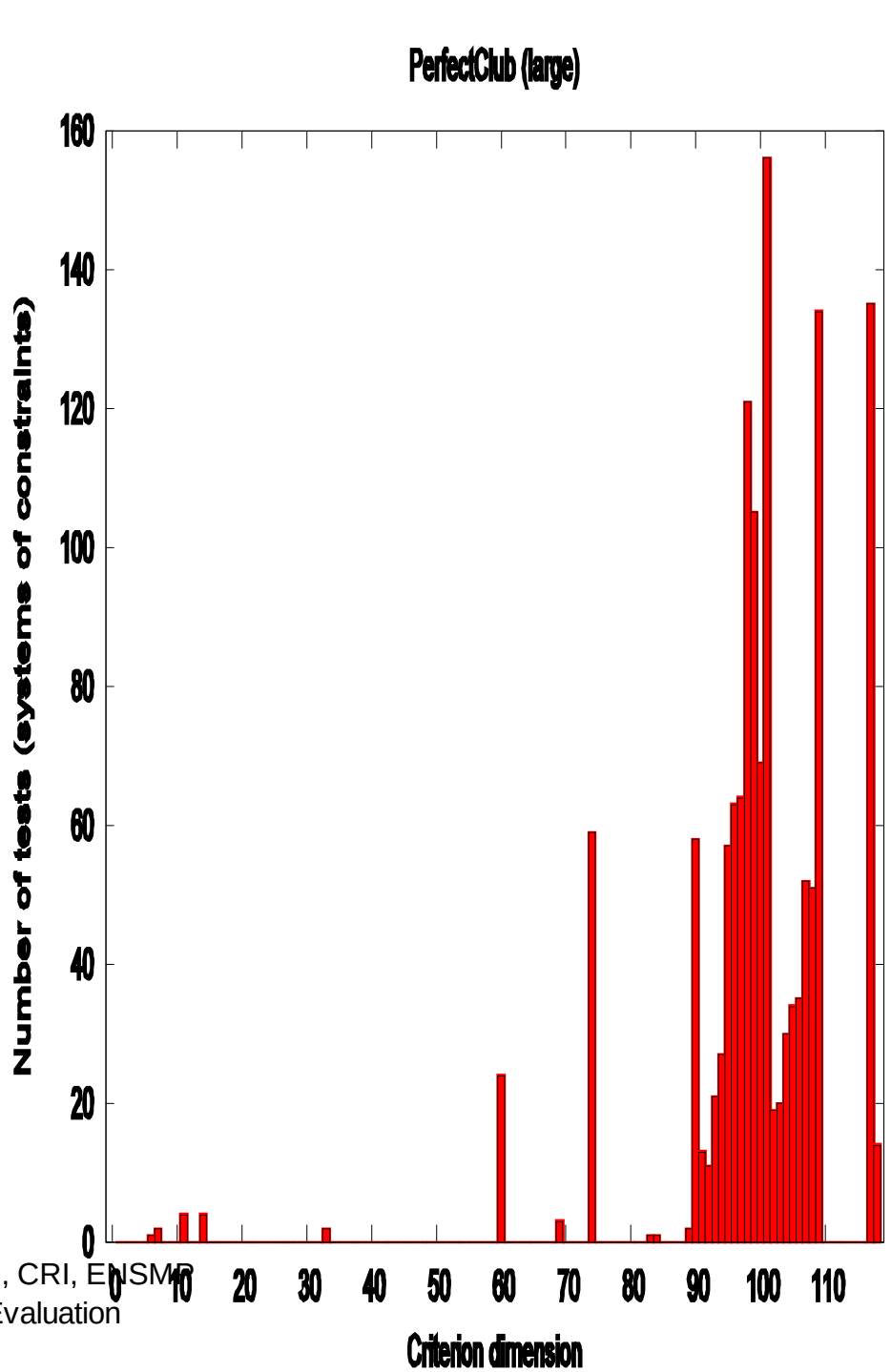
- Linux-gnu (Debian 3.0)
- SunOS (5.8)

Plusieurs distributions (histogrammes) par rapport aux critères (voir exemples)

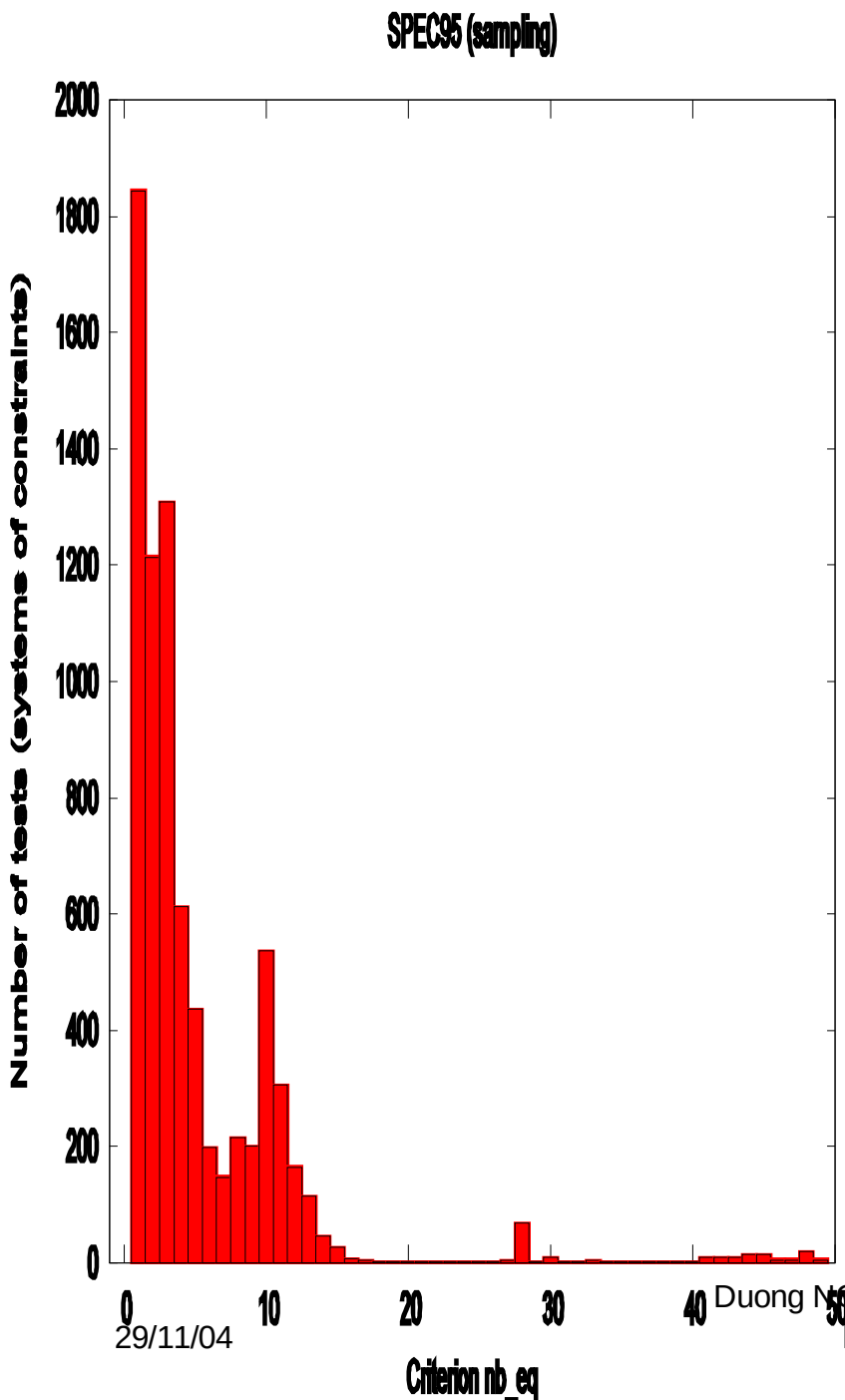
Problème:

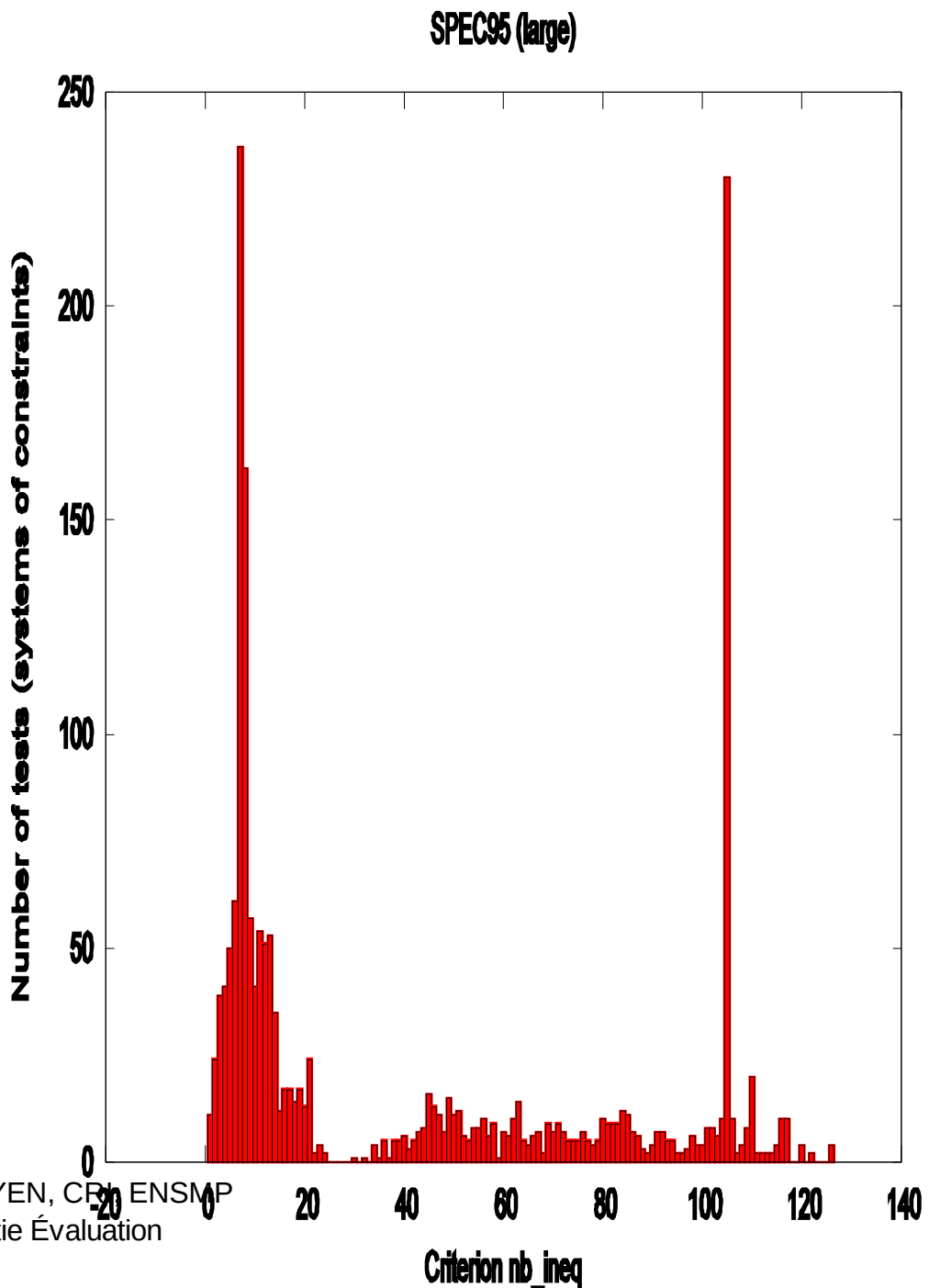
- Taille des bases de tests
- Deux bases de tests: échantillonné, filtré (problèmatique)





Duong NGUYEN, CRI, ENSMP


SPEC95 (sampling)
Number of tests (systems of constraints)
Criterion nb_eq
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
10
20
30
40
50


SPEC95 (large)
Number of tests (systems of constraints)
Criterion nb_ineq
0
50
100
150
200
250
-20
0
20
40
60
80
100
120
140

Faisabilité: comparaison

Janus 64-bit (INRIA) vs C3 Simplex (CRI)

Janus 64-bit (INRIA) vs C3 Fourier-Motzkin (CRI)

Nombre de lignes de codes (en langage C):

- Janus 64-bit: ~ 3403 LOC
- C3 Simplex: ~ 1488 LOC
- C3 FM: ~ 1000 LOC

Solution en entier

Critères des comparaisons:

- Temps d'exécution
- Nombre d'exceptions

Temps de conversion de format (interne) réduit

- Exécution = (conversion + 100 fois opérateur)/100
- Résolusion de l'horloge: milisecond

Accélération en temps d'exécution en total		
Perfect Club	Base échantillonnée à 10% 91825 tests	Base filtrée 1556 tests
Janus vs C3 Simplex	21 (fois)	17 (fois)
Janus vs C3 Fourrier-Motzkin	67 (fois)	41 (fois)

Nombre d'exceptions (64-bit)		
Perfect Club	Base échantillonnée à 10% 91825 tests	Base filtrée 1556 tests
Janus	62 (620)	164
C3 Simplex	602 (6020)	168
C3 Fourier-Motzkin	317 (3170)	164

Passage au dual: comparaison

Pourquoi étudier le passage au dual:

- Enveloppe convexe = 3 fois le passage au dual
- Problème avec deux arguments -> critères

Polylib - C3 (CRI) vs CDD (Institute for Operations Research, Zürich, Switzerland)

Nombre des lignes de codes (en langage C):

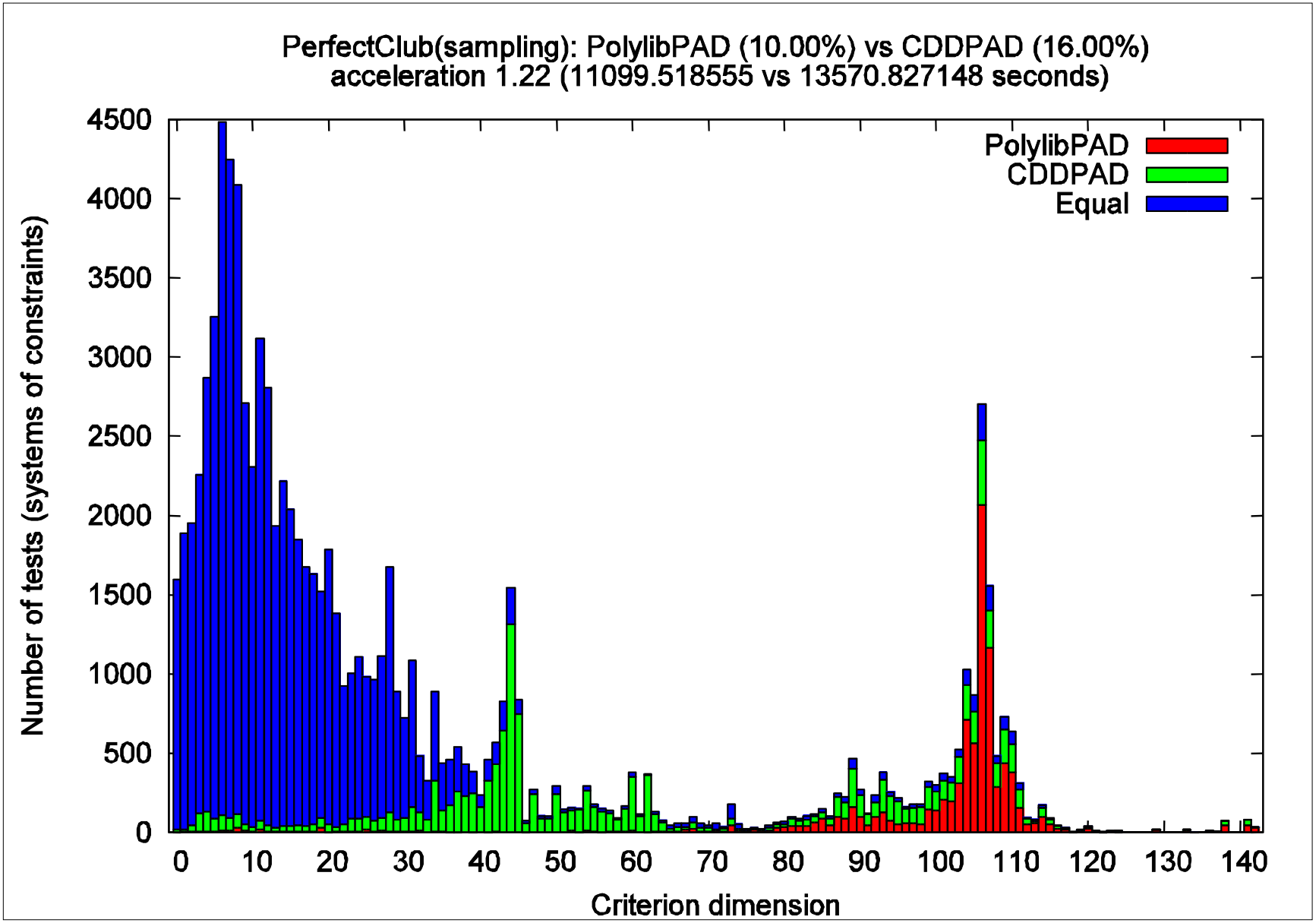
- CDD: > 4000 LOC
- Polylib: > 4000 LOC

Critères des comparaisons:

- Temps d'exécution
- Nombre d' exceptions

Temps de conversion de format (interne) réduit

- Exécution = (conversion + 100 fois opérateur)/100
- Résolusion de l'horloge: milisecond


PerfectClub(sampling): PolylibPAD (10.00%) vs CDDPAD (16.00%)
acceleration 1.22 (11099.518555 vs 13570.827148 seconds)
PolylibPAD
CDDPAD
Equal
Number of tests (systems of constraints)
Criterion dimension
0
500
1000
1500
2000
2500
3000
3500
4000
4500
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140

Accélération en temps d'exécution en total		
PerfectClub	Base échantillonnée à 10% 91825 tests	Base filtrée 1556 tests
Polylib vs CDD	1.22 (fois)	0.09 (fois)

Nombre d'exceptions (64-bit)		
PerfectClub	Base échantillonnée à 10% 91825 tests	Base filtrée 1556 tests
CDD	15 (150)	164
Polylib	384 (3840)	366

Résultats de l'évaluation (2002-2004)

Démarche fixée:
- Quantitatif
- Graphique
- Automatique
- Conversion de format interne

Résultats utilisables
- Janus dans PIPS
- CDD encourageant
- Gestion des heuristiques

À faire:
- Timeout
- Heuristiques
- En parallèle
- Décomposition en produits de polyèdres

Opérations déjà évaluées
- Faisabilité (C3, Janus, PIP)
- Passage au dual (C3, Polylib, CDD,LRS)
- Enveloppe convexe (C3 – Factorisation partielle, Polylib, New Polka)

Opérateurs en cours
- Projection
- Normalisation

Outils à considérer:
- LRS, Parma, Qhull, Porta, Omega

Conclusion

APRON:

- Coopération concernant l'API
- Benchmarks d'opérateurs polyédriques
- ~~Décomposition de polyèdres (David Merchat)~~
- API générique (PIPS)
- Changement de domaine automatique?

PIPS:

- Janus (CDD, LRS)
- Utilisation de l'API

Travaux prévus hors APRON:

- Parma
- Omega