

*Une approche expérimentale
pour l'algorithmique parallèle
sur grappes et grilles de
calcul*

Christophe Cérin

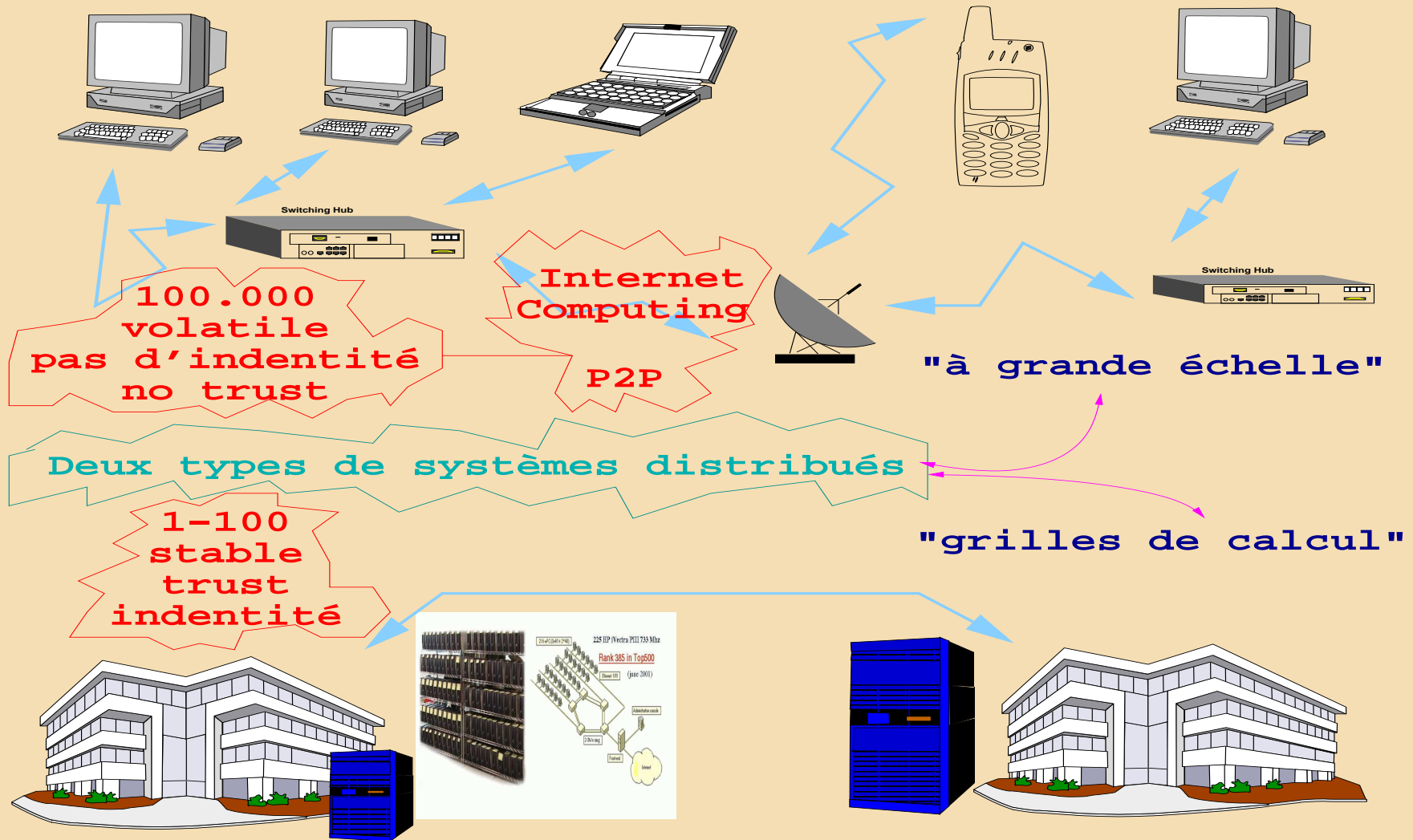
cerin@laria.u-picardie.fr



- x Contexte, démarche et positionnement
- x Archi processeurs et cache
- x Algorithmes mémoire commune
- x Le tri sur clusters
 - homogène et hétérogène
- x I/O hautes performances
 - Représentations sur disques
 - Collecte, analyse d'événements



CLASSIFICATION SYSTEMES DISTRIBUÉS





⇒ Définir les **concepts** et **outils**

✗ résoudre efficacement les problèmes
avec de grandes masses de données

✗ grappes et grilles

⇒ Démarche à **double sens**

✗ expérimentale

✗ garanties algorithmiques prouvées



⇒ Établir un **plan d'intégration**

1) Comparaison

X mise en évidence des conflits

2) Mise en conformité

X résolution des conflits

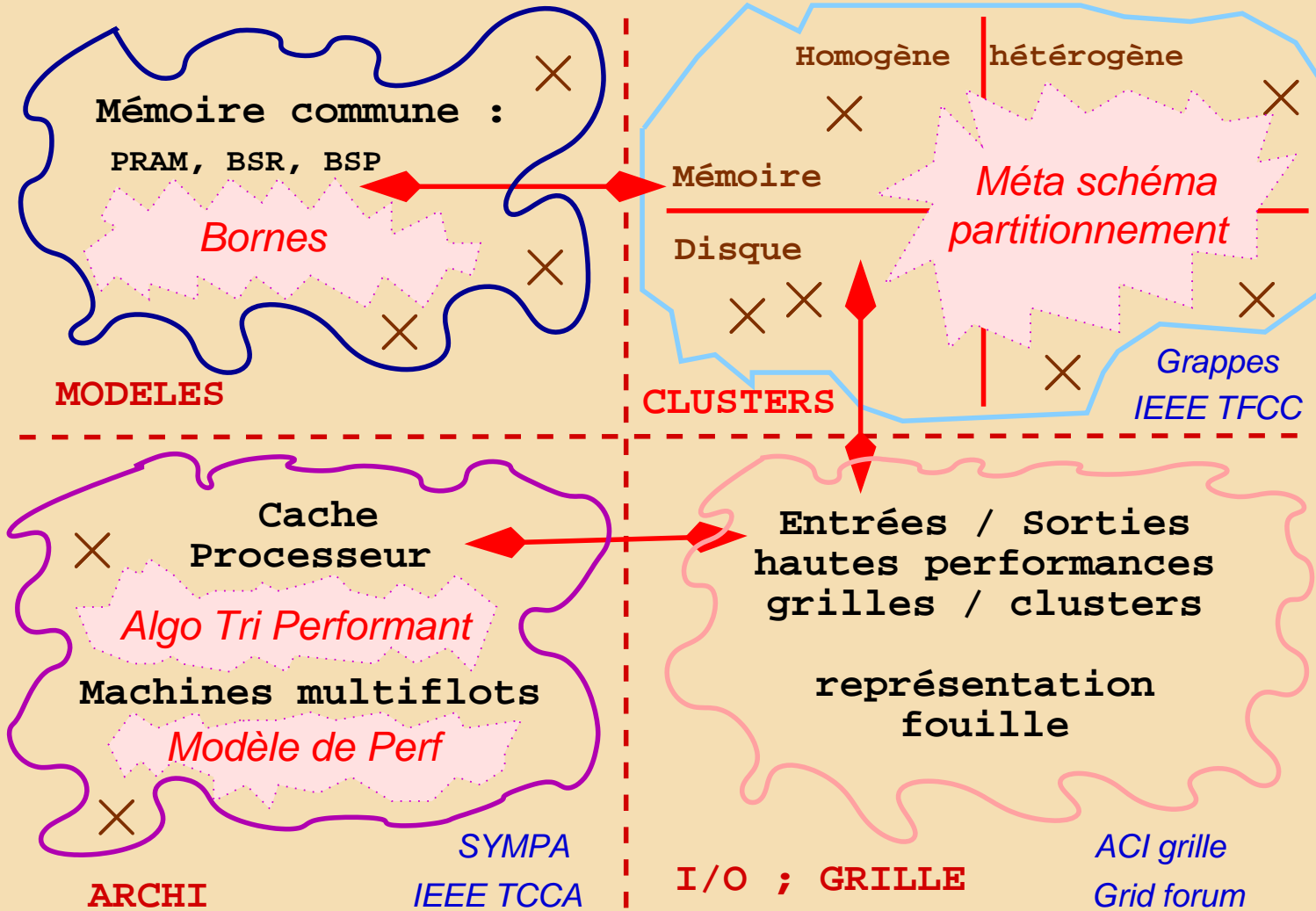
3) Fusion des schémas

4) Restructuration

X amélioration du schéma global



THÉMATIQUES SCIENTIFIQUES



ARCHITECTURE MACHINES MULTIFORMS

1- ARCHITECTURE MACHINES MULTIFLOTS



Cray MTA :

- * 128 threads / proc
- * up to 8 conc. mem ref / proc
- * no cache



Evaluation :

- o déf de mesures : max, min de parallélisme (mot puis langage)
- o Résultat : mesures calculables par induc. hauteur étoile expr. régulière
- o pas de proba. (nouv. approche)

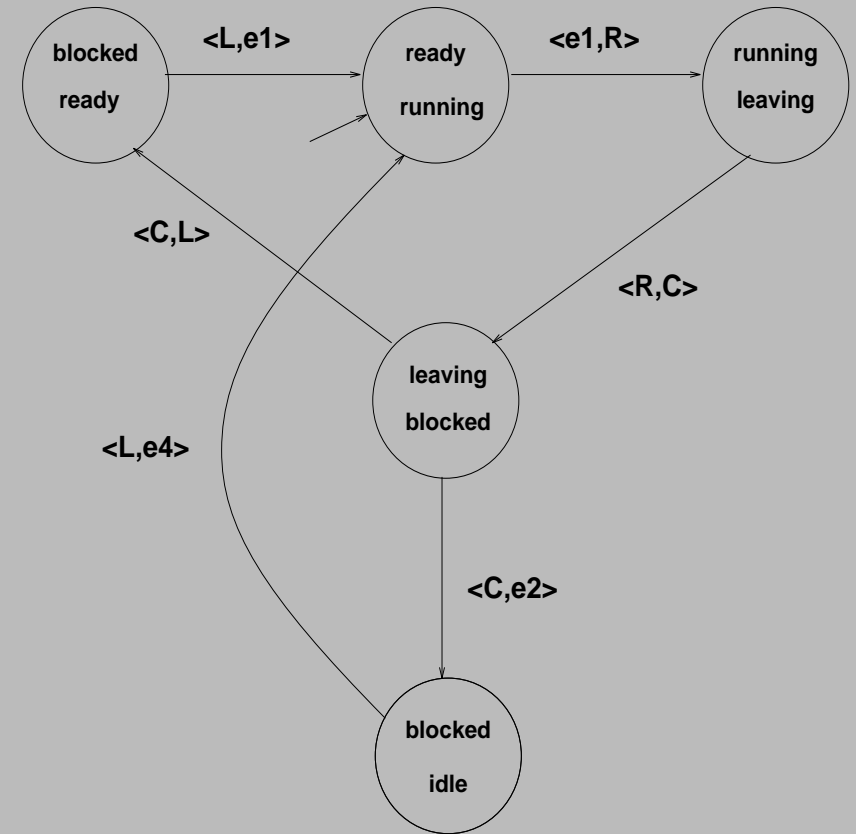
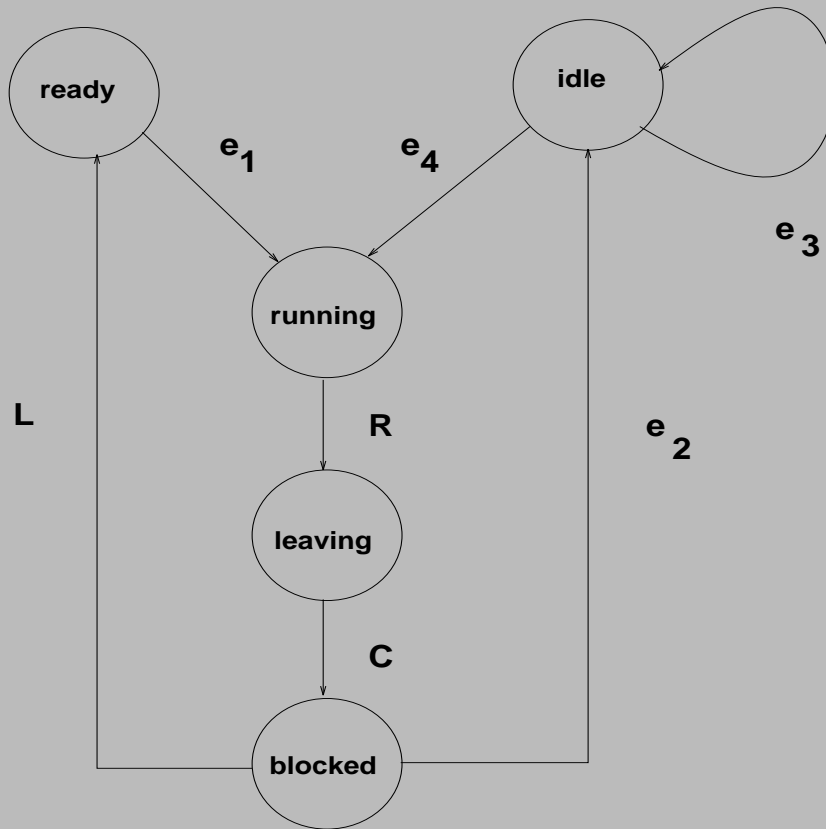


EXEMPLE



2 threads - 1 UC

Prod n-1 fois



ARCHITECTURE

REPORT ON CASE L SW T



- ⇒ **Objectifs** : impacts du cache sur les performances des tris séq.
- ⇒ Dualité cache / I/O - borne en termes d'I/Os pour le tri disque
- ⇒ Quelles techniques de mesure ? (simulation ou compteurs de perf.)
- ⇒ Algorithmes
 - × localité spatiale et temporelle
 - × compteurs de perf



⇒ FastSort ($n \log \log n$), FAME,
3-way-Quicksort, mergesort.

⇒ Élimination de heapsort.

⇒ **Nouvel Algo. ZZZmerge**

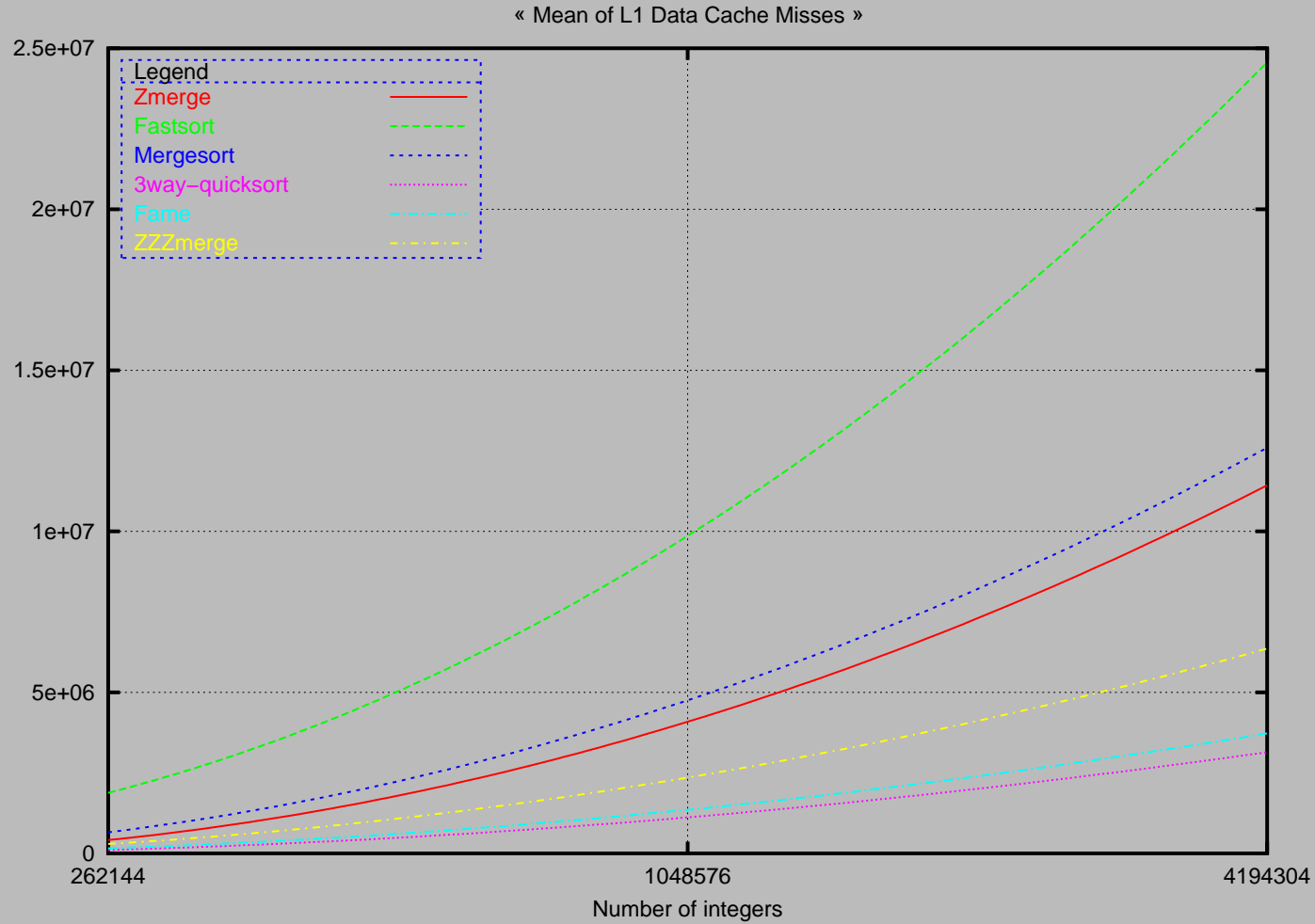
⇒ Principe : mergesort

⇒ **RÉSULTATS**

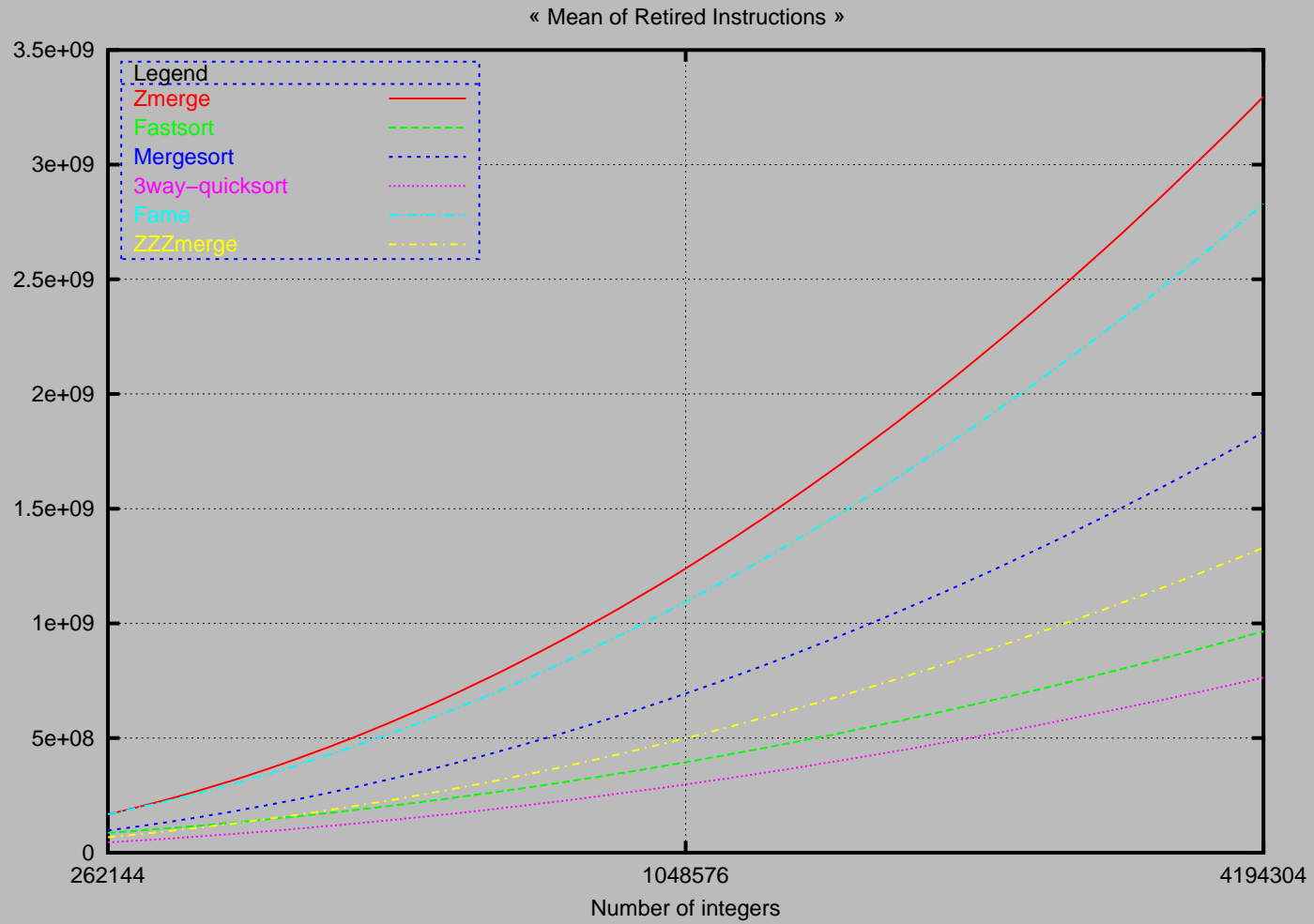
✗ éval. exacte du nb de défauts ZZZmerge

✗ Gains entre 9% et 15% de ZZZmerge

OBSERVATIONS DU NB DE DÉFAUTS L1

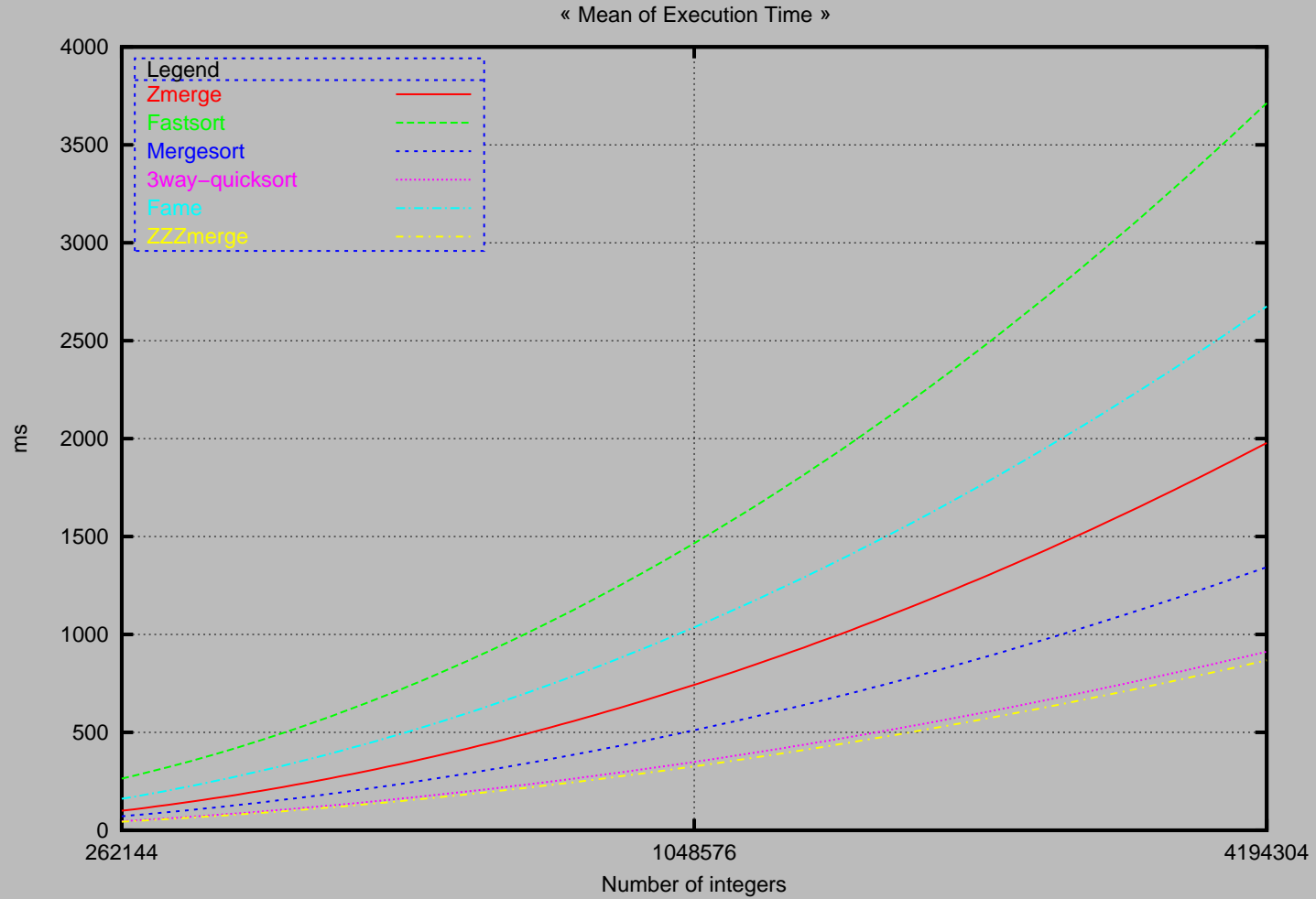


OBSERVATIONS RETIRED INSTRUCTIONS

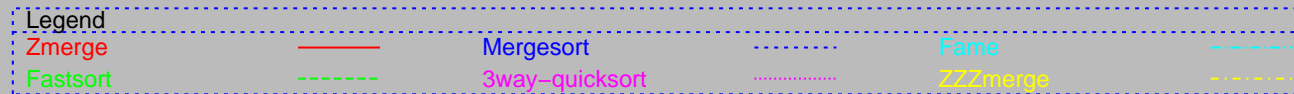
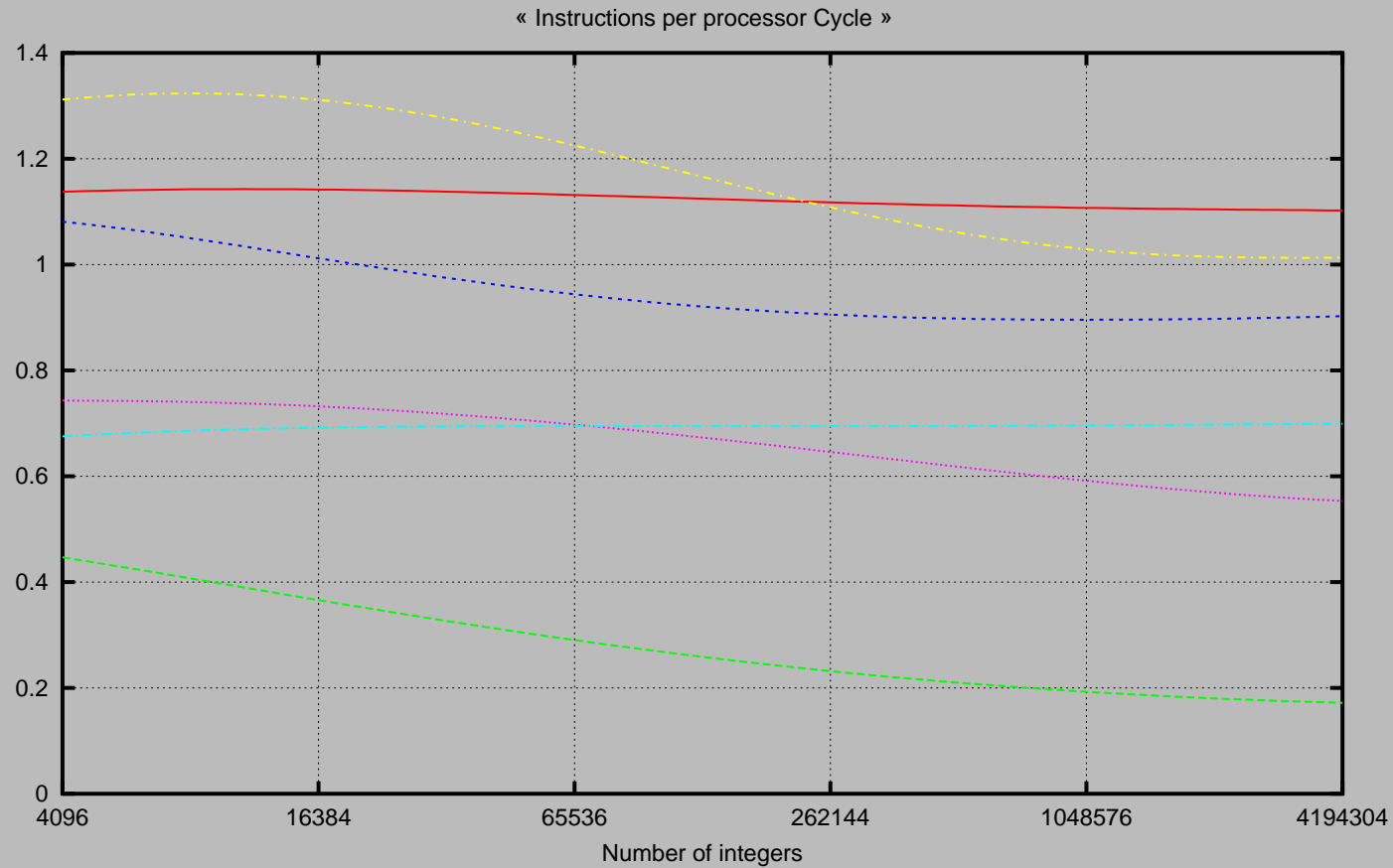




OBSERVATIONS TEMPS D'EXÉCUTION



OBSERVATIONS IPC



- ⇒ *"Si X a une IPC au moins 2 fois plus grand que Y mais qu'il a un nb de défauts de cache L1 et un nb d'inst. exécutées 2 fois plus important que Y, alors X a quand même une exécution en temps meilleure que Y."*
- ⇒ Perspectives : Paral. dans les circuits
- ✓ repenser les algos et les compilateurs
 - ✓ tri : gain encore possible



ALGORITHMOLOGIE MEMOIRE COMMUNE

⇒ **PRAM** : Parenthesis matching -> éval expr. arith.

✗ Atteinte de la borne + présentation propice à une impl. message passing.

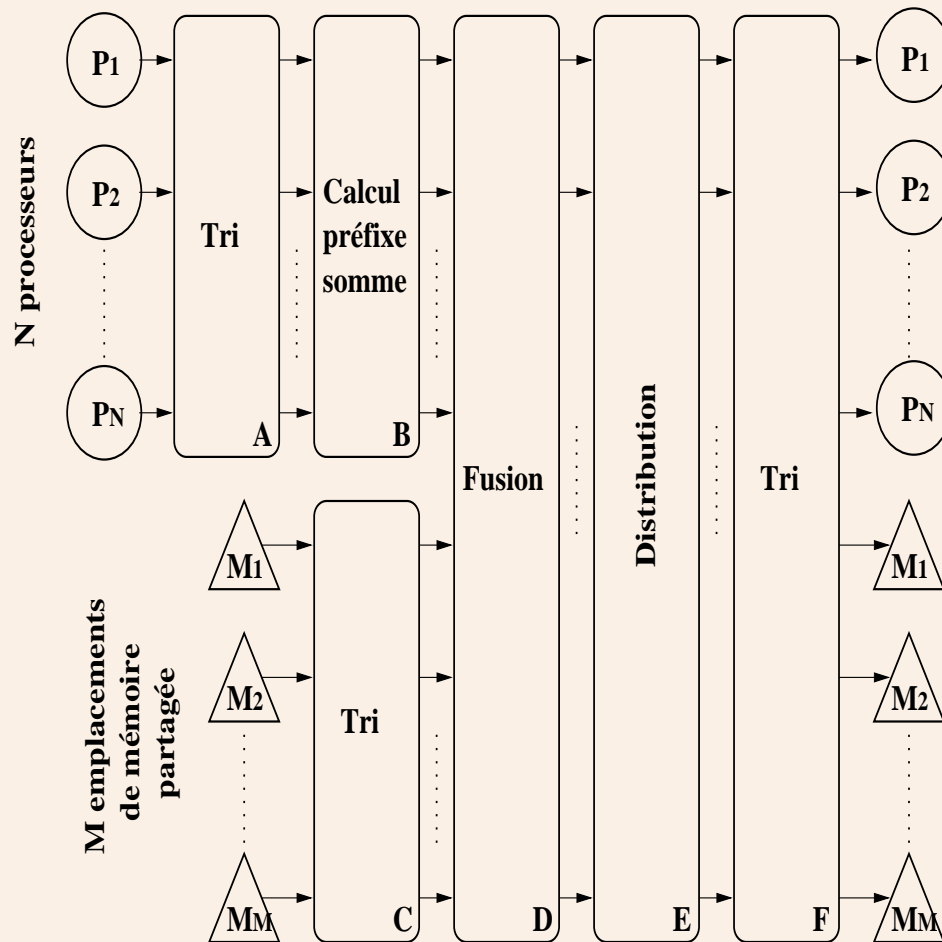
⇒ **BSR** : segment somme maximale (1-D, 2-D).

✗ Atteinte borne avec moins d'inst BSR.

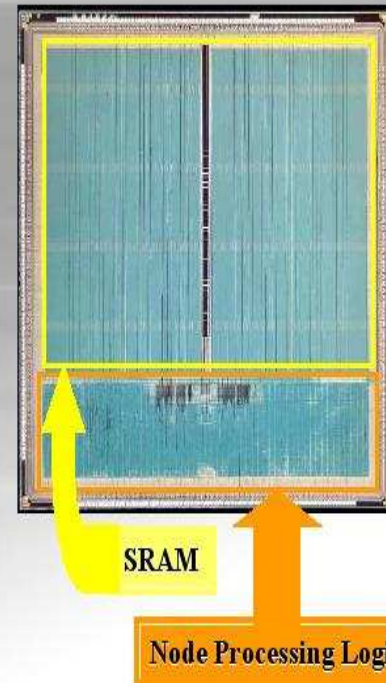
✗ Nouvel Inst. BSR

✗ Règles BSR -> BSP (compilateur).

BSR / PROCESSOR IN MEMORY (PIM)



1st DIVA PIM Chip



- ◆ Purpose
 - Demonstrate bandwidth advantages of PIM technology
- ◆ Key architectural components
 - High memory bandwidth
 - 256-bit WideWord processing
 - PIM routing component
- ◆ Chip statistics (55 million transistors)
 - 9.8mm X 9.8mm in TSMC 0.18 μ m
 - ~200K logic cells plus 8Mbit SRAM
 - Running Cornerturn application at 160MHz while dissipating 0.8W
- ◆ Package
 - 35mm, 352 BGA
 - 241 signal I/O, 111 Vdd or Gnd



Sort : $r_j = \sum 1 | x_i \leq x_j$



ALGORITHM MODE TRA SUR CLUSTERS



⇨ Problème difficile

✗ Accès mémoire imprédictible

✗ Absence garanties : NowSort, NAS 2.3

✗ #RAM \subseteq Taille : Minute Sort

⇨ Objectifs multiples

▶ Schéma générique

- mémoire + disque

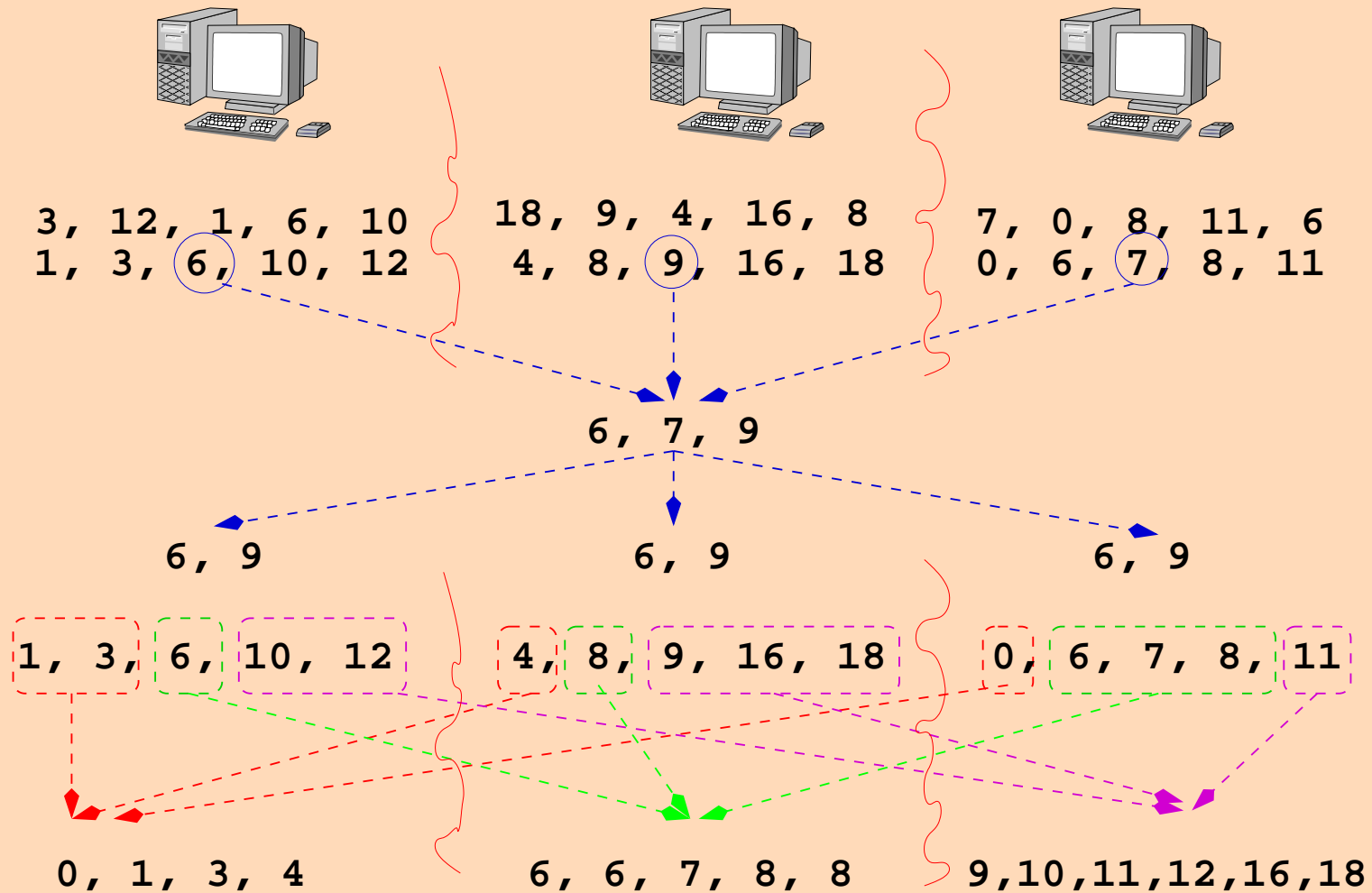
- Hétérogénéité (jamais étudié)

▶ Garanties sur Time et Work ($\forall n$)

▶ Utilisant le moins de RAM -> non dédié.

▶ Reproductibilité des expériences.

MÉTA-SCHÉMA : ÉCHANTILLONAGE





▶ Hétérogénéité

X Procs à des vitesses différentes

X Paramétrage du code

▶ Reproductibilité des expériences

X 8 tests possibles ; Codes source ;

▶ RÉSULTATS

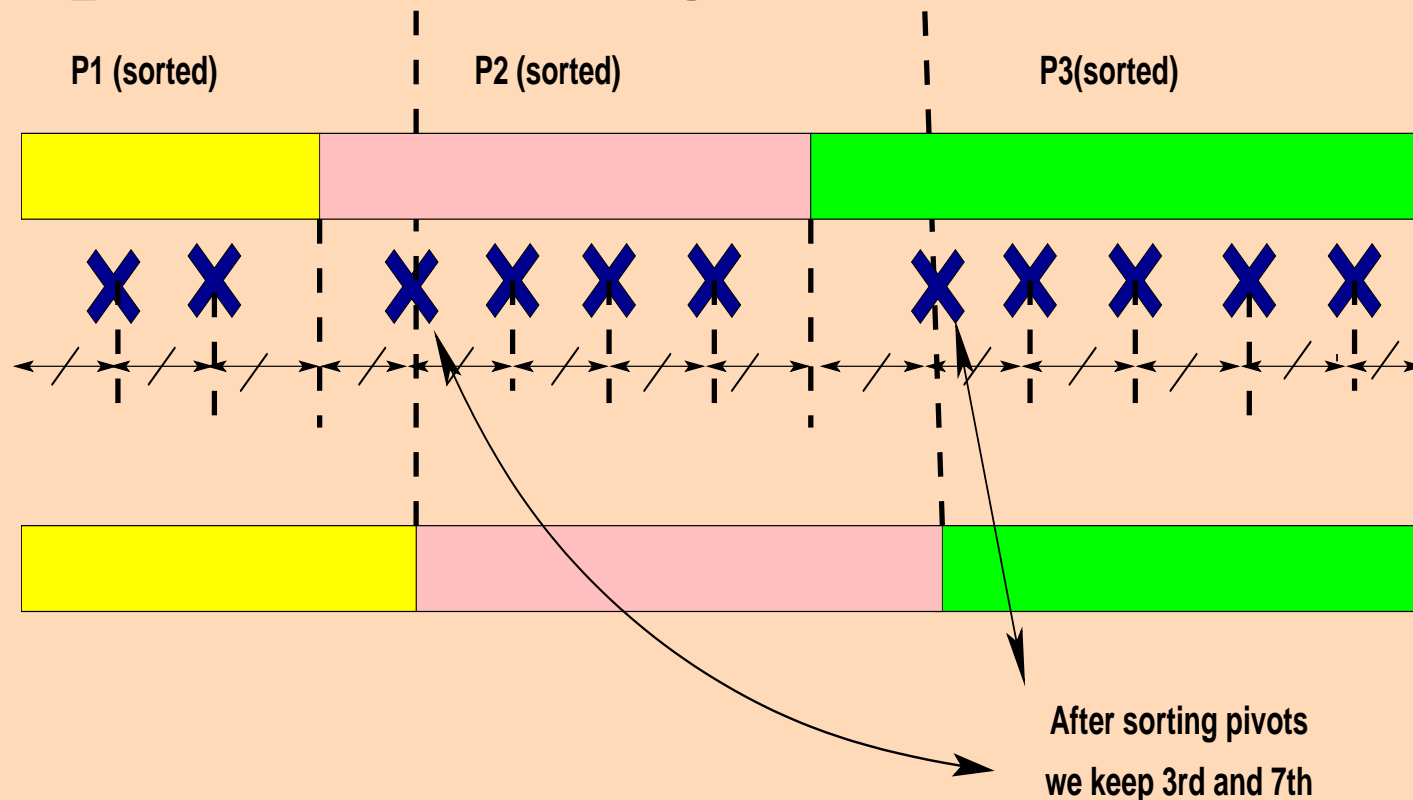
X hétérogène ou pas,

X mémoire ou disques

X garanties formelles : charge, T d'exé.

▶ Avec tri initial : aucun proc ne traite plus de 2 fois ce qu'il avait au départ.

▶ Exemple en hétérogène :





► Sans tri initial et cas homogène

$(p.k-1)$ pivots partitionnent l'entrée en $p.k$ morceaux telle que la taille du plus grand morceau est plus petite ou égale à n/p avec la probabilité au moins égale à

$$1 - 2p \left(1 - \frac{1}{2p}\right)^{p.k}$$

► Amélioration de l'équili. pour cas hétérogène : simulation de + de processeurs ($p \rightarrow p' = \sum_{i=1}^p perf[i]$) et réutilisation du TH.



RÉSULTATS TRIS DISQUES HÉTÉROGÈNES



Critère	H-PSS	H-PSRS	H-PSOP
Tri initial	Non	Oui	Non
Tri final	Oui	Merge	Oui
Nb Candidats	$6p' \log p'$	$p' (p-1)$	$4p'p' \log p'$
Nb pivots	$P-1$	$P-1$	$4p' \log p' -1$
Equilibrage théorique	Proba	≤ 2	Proba
Equilibrage mesuré	+,- 15%	+,-0.1%	+,-0.01%
Texe	1	3	2



▷ Hétérogénéité des processeurs

▷ Méta-schéma (échantillonnage)

✗ mémoire / disques

✗ codes

✗ tests : utilisations multiples

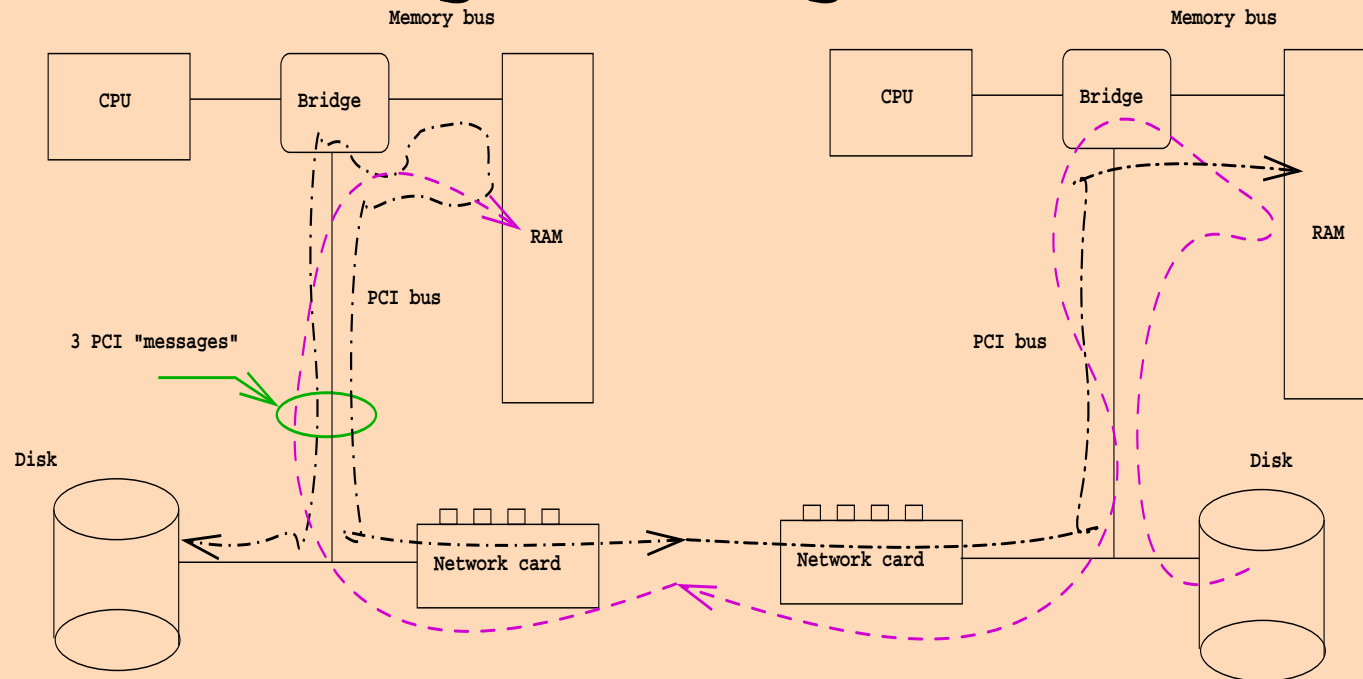
GARANTIE :

✗ équilibrage des charges

✗ temps d'exécution

▶ RÉUTILISATIONS IMMÉDIATES

- ▶ Bases de données (cf jointure)
- ▶ Validation Read² (disk to disk efficace)
- ▶ Benchmarking hétérogène





▶ **PORTÉE GÉNÉRALE DE NOS APPROCHES**

▶ **PARTITIONNER DES DONNÉES**

lorsque la machine est hétérogène

▶ **GÉNÉRALISATIONS HÉTÉ D'ALGOS**

▶ **Fouille (regroupement).**

1- prendre un sous-ensemble de données s

2- partitionner s en p partitions de
taille s/p

3- dans chaque partition, créer s/pq clusters

▶ **Partitionnement de graphes**

10 PARALLELS
WHITE PERFORMANCE
SUN GALLS

► Motivations stockage disques

✕ Vidéo à la demande

✕ Bases documentaires, web

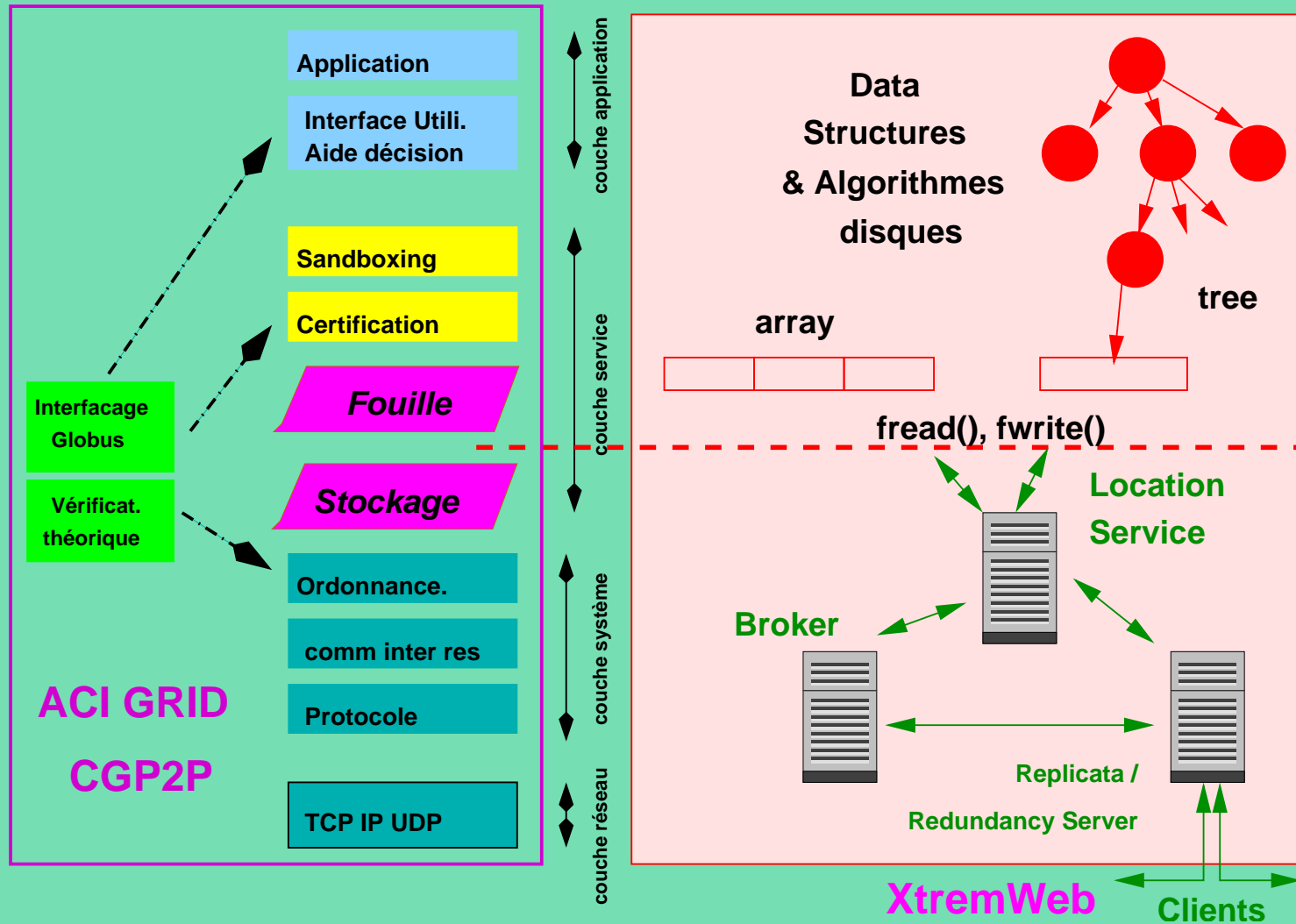
✕ Applications scientifiques

► Densité des disques ▼ de 60 à 80% par an
MAIS les temps d'accès ▼ de 10% par an.

► Problématique sous-étudiée



STOCKAGE DANS "GRILLE"



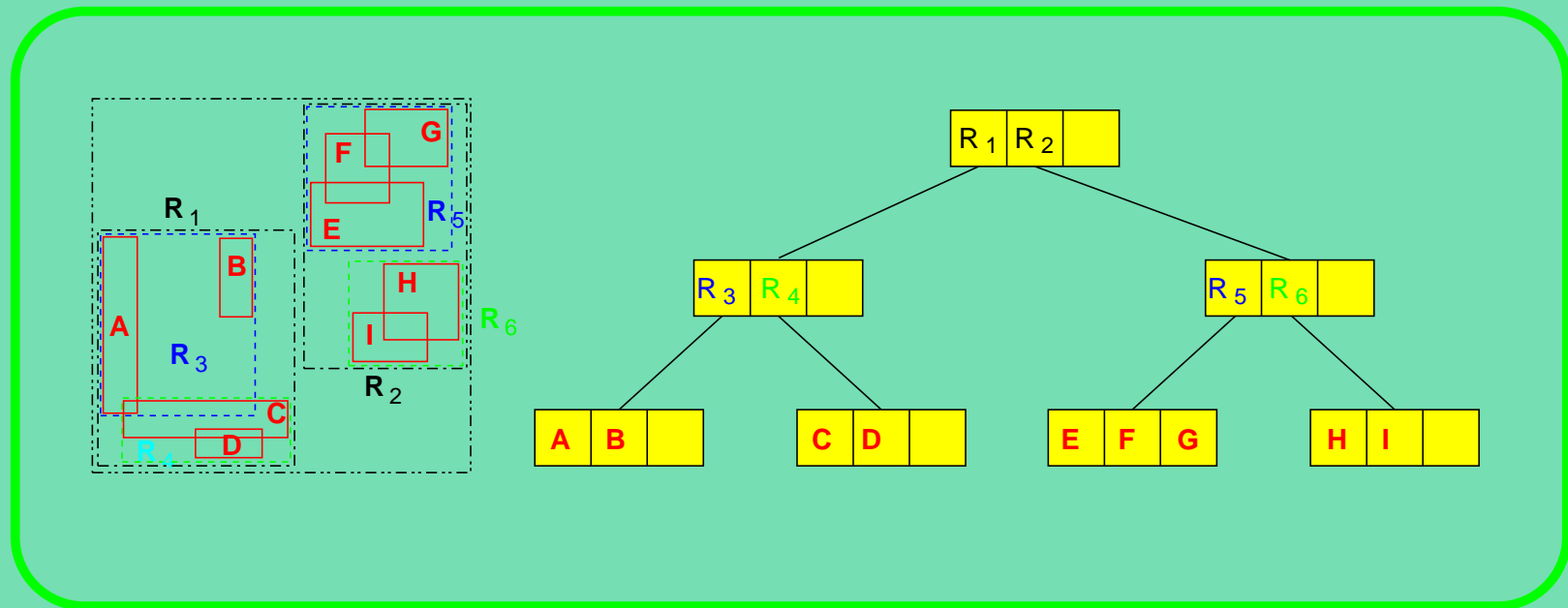


- ❑ Placement de taches : difficile
- ❑ Nécessité de modèles
- ❑ Il existe peu d'heuristiques
(Pb retour comportement des grilles)
- ❑ Trace utilisée
 - ✗ 110 PC de bureau (LAL à Orsay)
 - ✗ 15 jours de mesures
 - ✗ 15 events et infos RARES
- ❑ Difficulté monter expériences
- ❑ Structures, stockage des traces

PLACER ARBRE 1TO SUR LA GRILLE



...composée de 1K machines



Requêtes: Contient, Est Contenu, Est Egal.



- ✗ On veut **prédire** avant de placer
- ✗ Fouille : produire des règles qui décrivent les relations entre différentes séquences



Ex: si (BF) arrive 4 fois tandis que -
(ABF) arrive 3 fois le compilateur
produit la règle :
"si BF arrive alors il y a 75%
de chance de voir arriver A"



⇒ **Explosion combinatoire** du nombre d'états

Ex: $m=k=15$: 7174453500000000000000
séquences !

⇒ **ÉLAGAGE**

- seuil (WINEPI, MINEPI)
- intégration contraintes sur les séquences d'entrées (cSPACE, SPIRIT)

- ⇒ Versions améliorées de grep ; "approximate string matching" : insert, del, substitution.

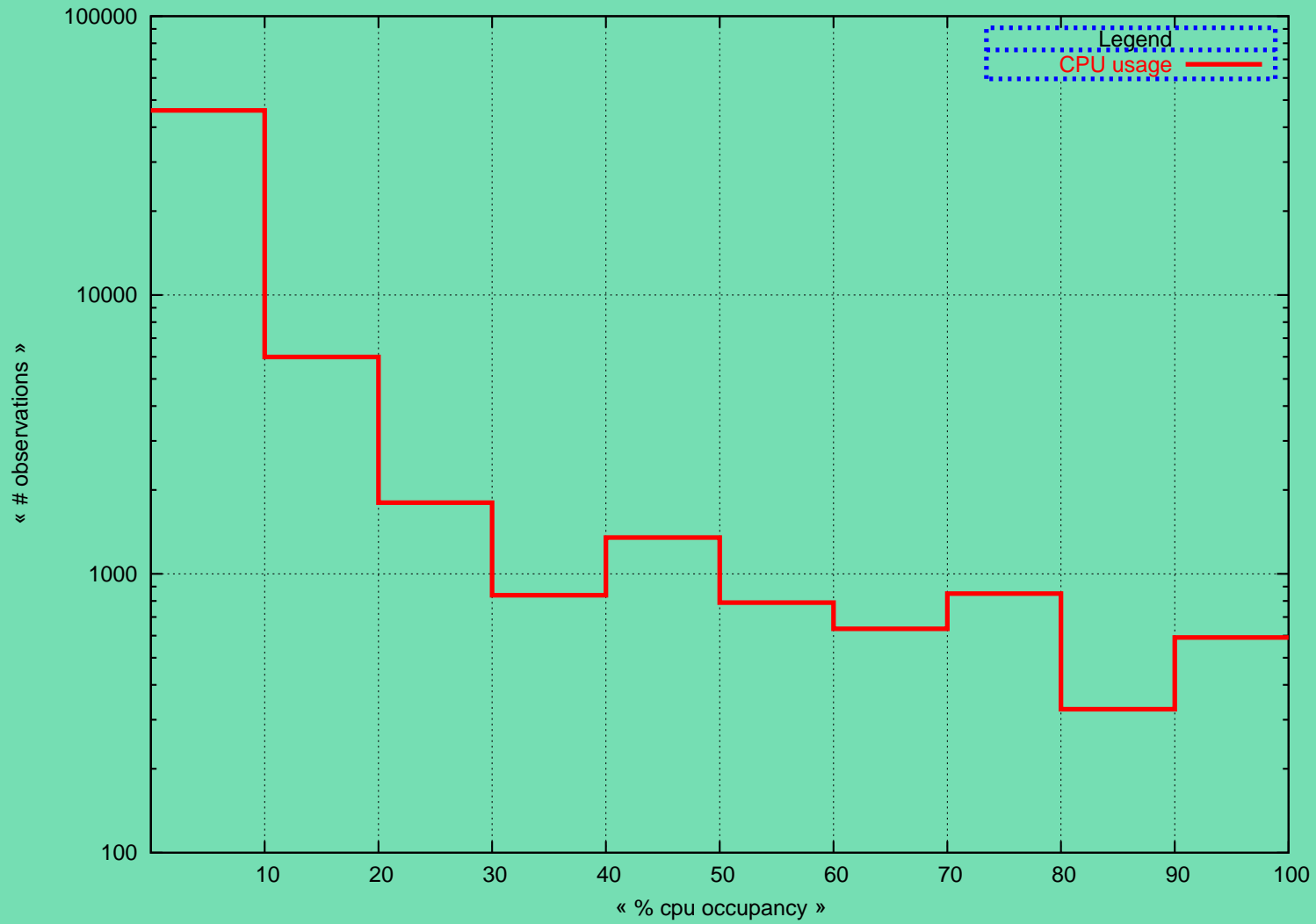


Candidats potentiels pour la fouille.

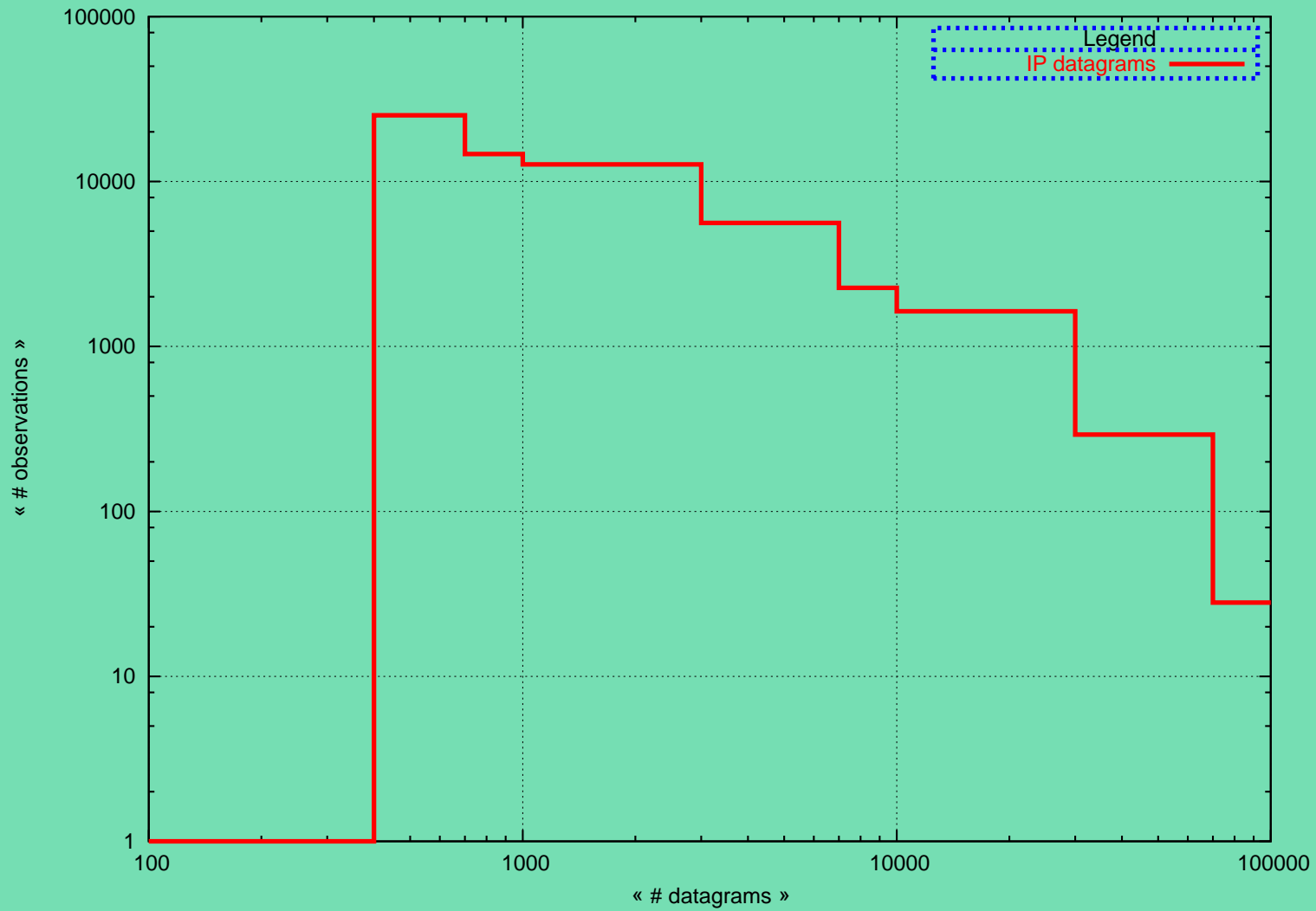
- ⇒ nrgrep-1.1.1/ (Gonzalo Navarro at the University of Chile)
- ⇒ Améliorations envisagées
 - corrections de **Bugs**



EXPLORATION DE LA CHARGE CPU



DÉNOMBREMENT DES DATAGRAMMES IP





- X Étude statistique disponible
- X Les choix des outils de fouille et de représentation des données se précisent
- X Défi : fouille multi-dimensionnelle
- X **Problématique** de la fouille est sous-étudiée dans le Grid-Forum



CONCLUSIONS - GENERAL RESULTS - PERSPECTIVES



► Architecture

- ✗ Méthode novatrice d'éval perf
- ✗ Cache : nouvel algo de tri (9% - 15%)

► Modèles

- ✗ PRAM, BSR : bornes
- ✗ BSP : comparatif PUB7 / BSPLIB

► TRI SUR CLUSTERS

- ✗ Hétérogène, disque, mémoire: GARANTIES
Expérimentations - Benchmarking hété.
- ✗ Méta-schéma de partitionnement
en hétérogène réutilisable.



○ Fouille, représentation données,
stockage sur grille

× Sous étudiés

× Premiers résultats : statistiques

× Choix structures données

× Choix des algos. fouille

× Partitionnement à large échelle

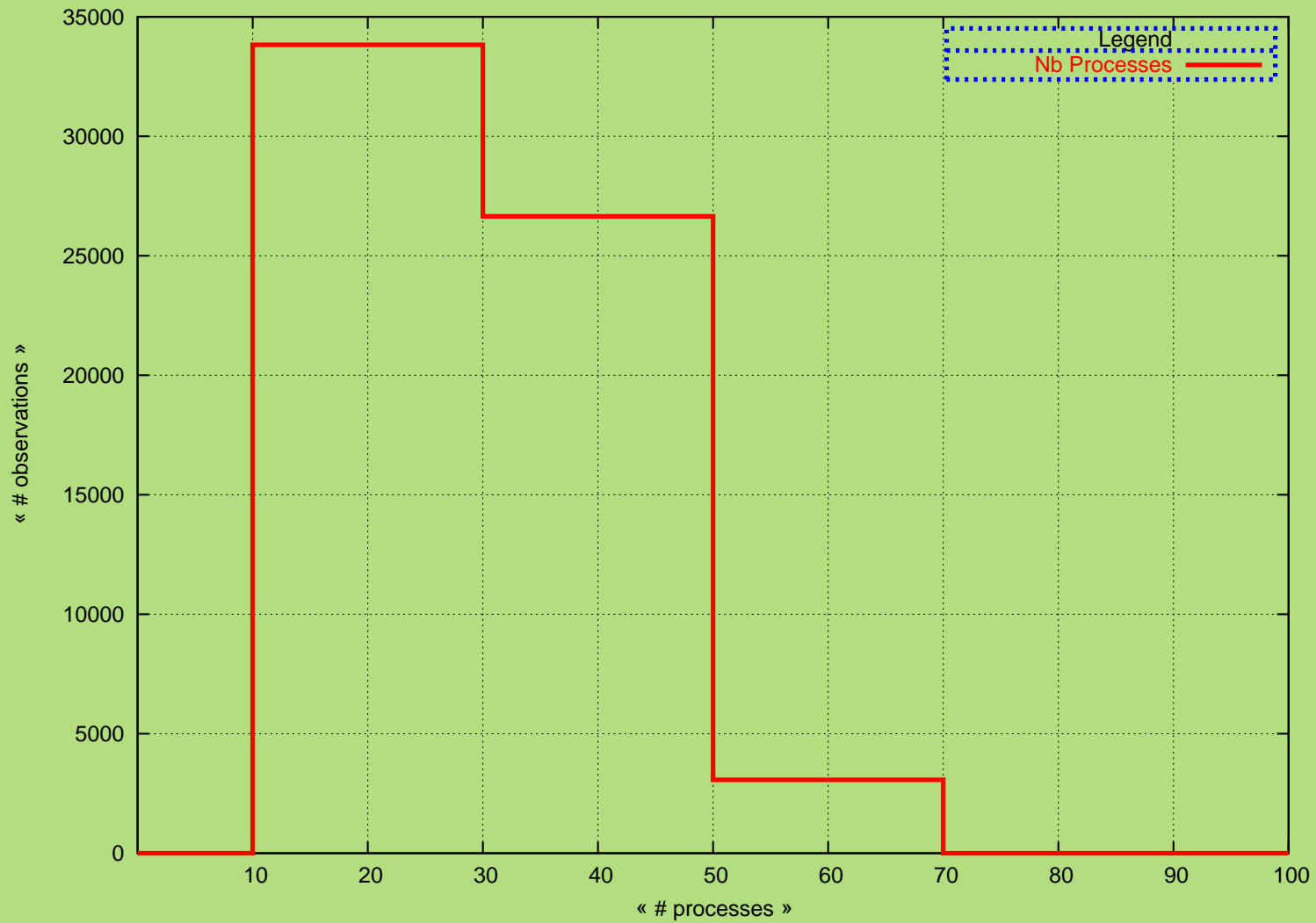
× Positionnement dans le Grid-Forum



QUESTIONS

REPORTS

NOMBRE DE PROCESSUS

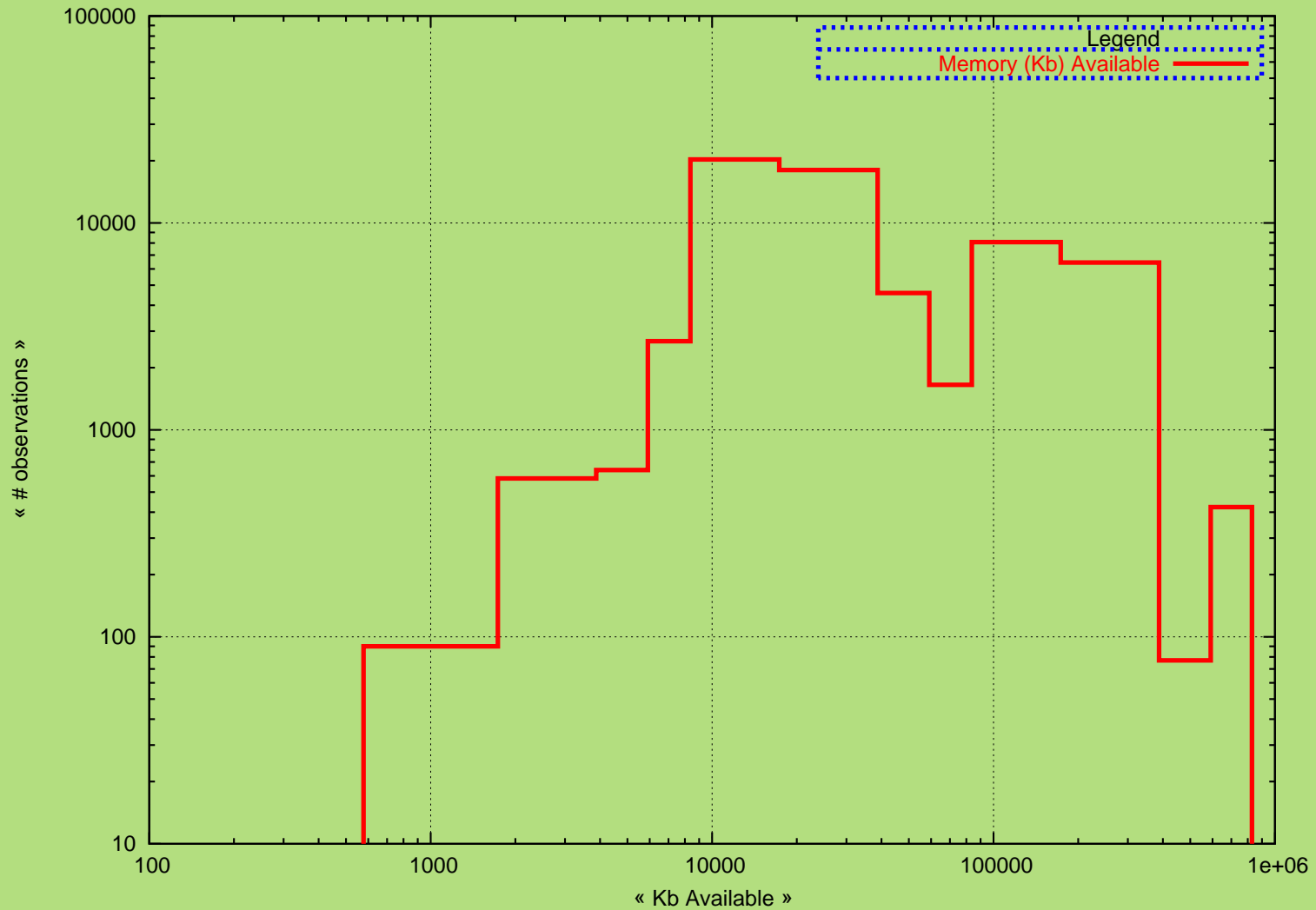


TRANSFERTS DISQUES





MÉMOIRE DISPONIBLE



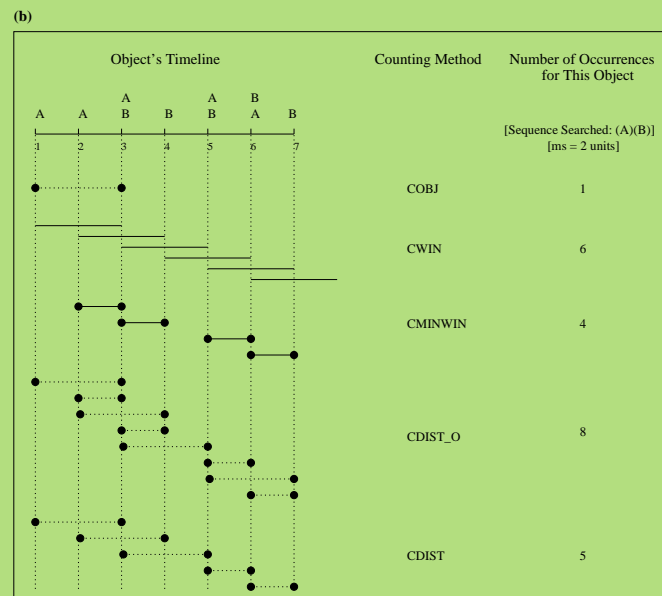
FOUILLE : CRITÈRE DE SÉLECTION



- Une séquence est intéressante si elle apparaît suffisamment de fois ;
- Différentes méthodes de comptage (>8) ;

(a)

	Count All	Count Minimal
Count Windows	CWIN	CMINWIN
Count Occurrences	CDIST_O	CDIST





⇒ **Explosion combinatoire** du nombre d'états

Ex: $m=k=15$: 7174453500000000000000
séquences !

⇒ **ÉLAGAGE**

- seuil (WINEPI, MINEPI)
- intégration contraintes sur les séquences d'entrées (cSPACE, SPIRIT)



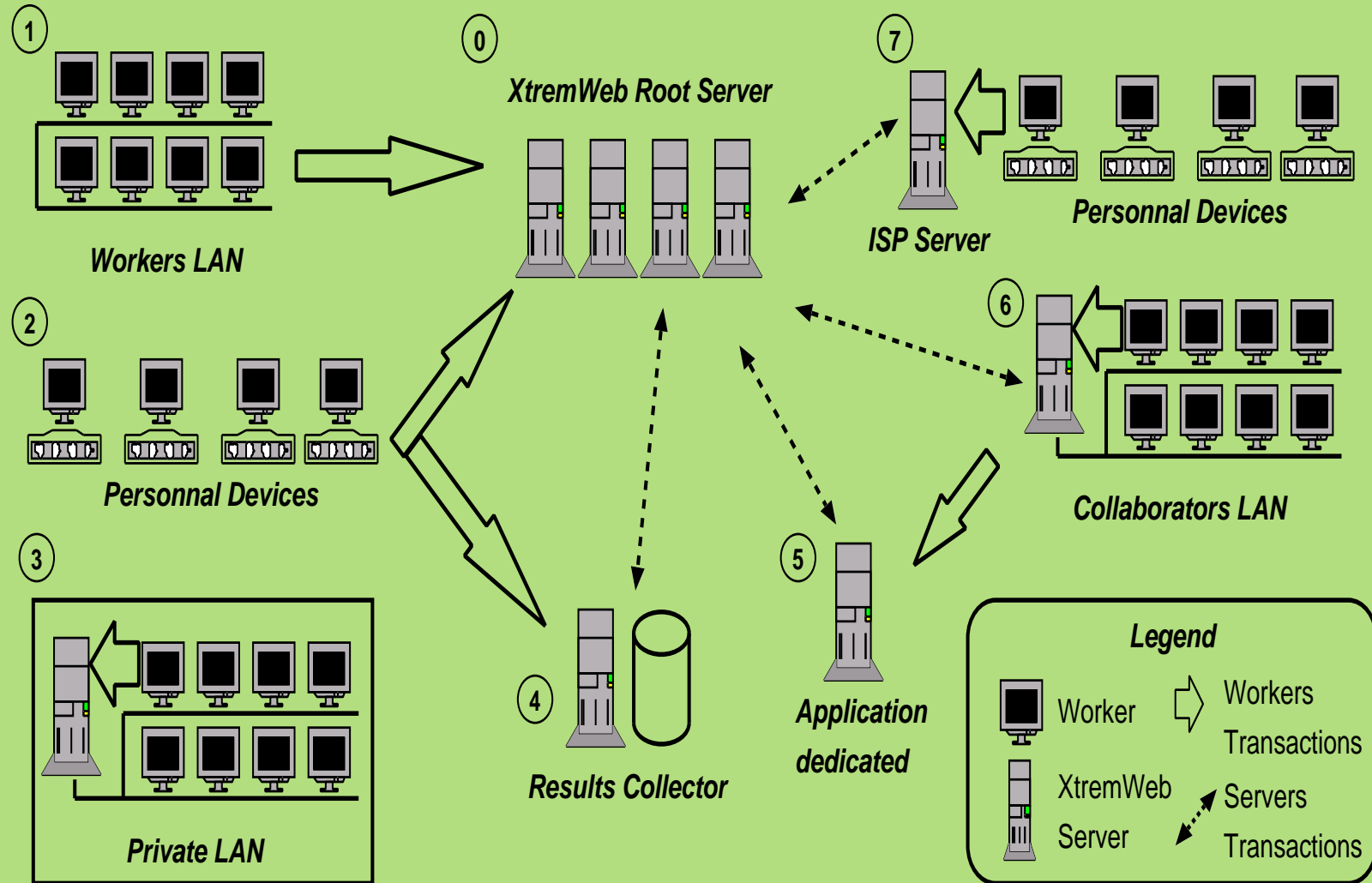
- ⇒ Règle d'antimonotonie : pour qu'une séquence soit fréquente il faut que toutes les sous-séquences le soient
- Utilisation du seuil
- ⇒ SPIRIT [VLDB 1999] : introduit un langage souple de spécification des contraintes ;
- ⇒ cSPADE [Zaki, 2001] : reconnu comme le plus performant...mais problème avec le format d'entrée ;



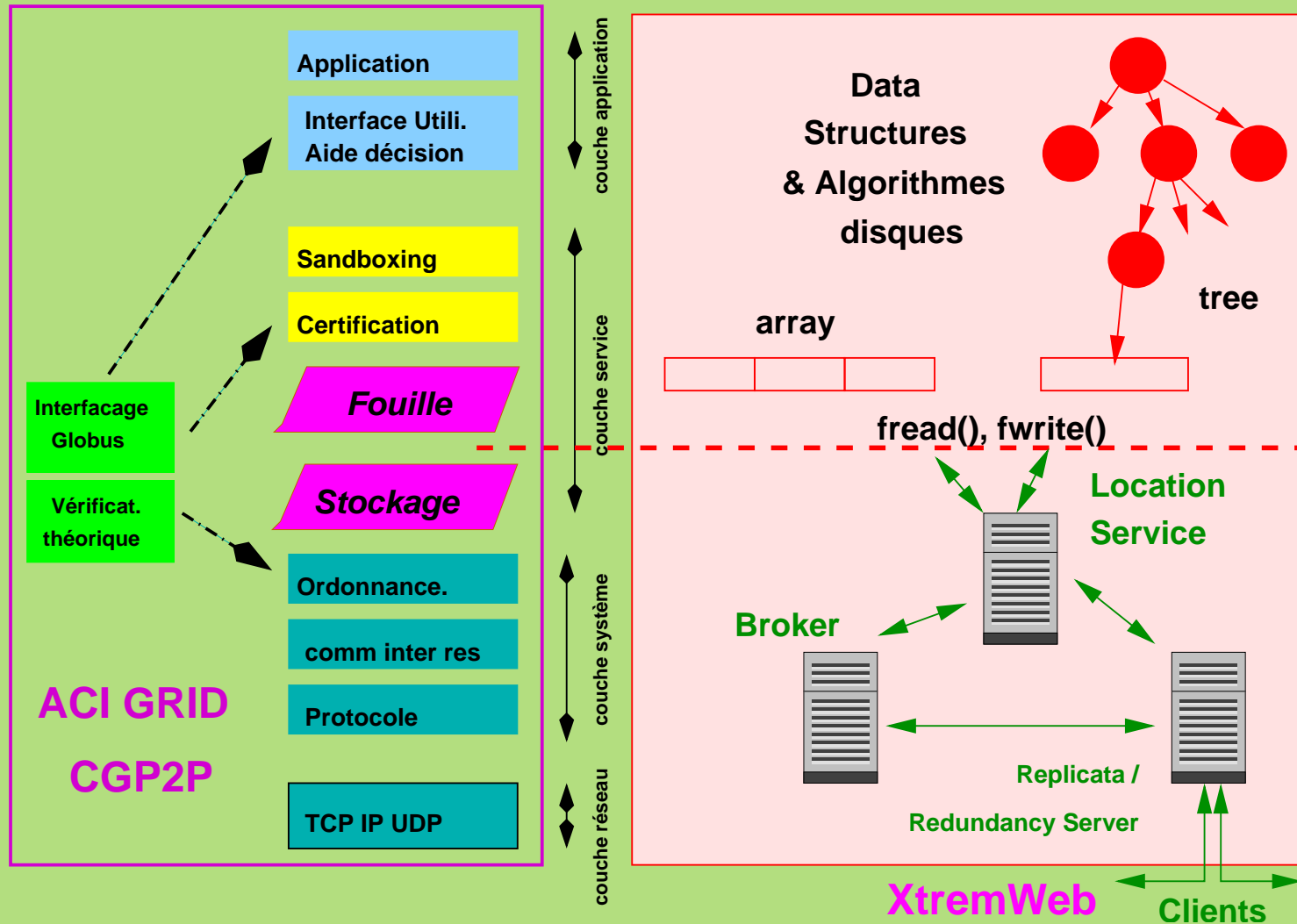
- ⇒ Exprimer des contraintes au niveau des motifs ;
- ⇒ SPIRIT [VLDB 1999] : introduit un langage souple de spécification des contraintes ;
- ⇒ cSPADE [Zaki, 2001] : reconnu comme le plus performant...mais problème avec le format d'entrée ;



INTERNET COMP : XtremWeb



STOCKAGE DANS "GRILLE"

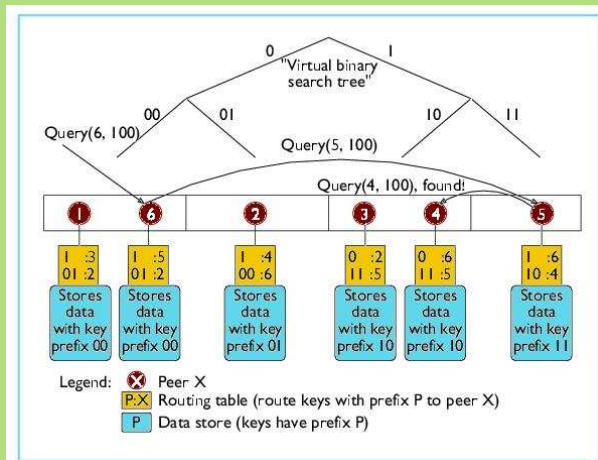


DATA ACCESS: GRIDELLA



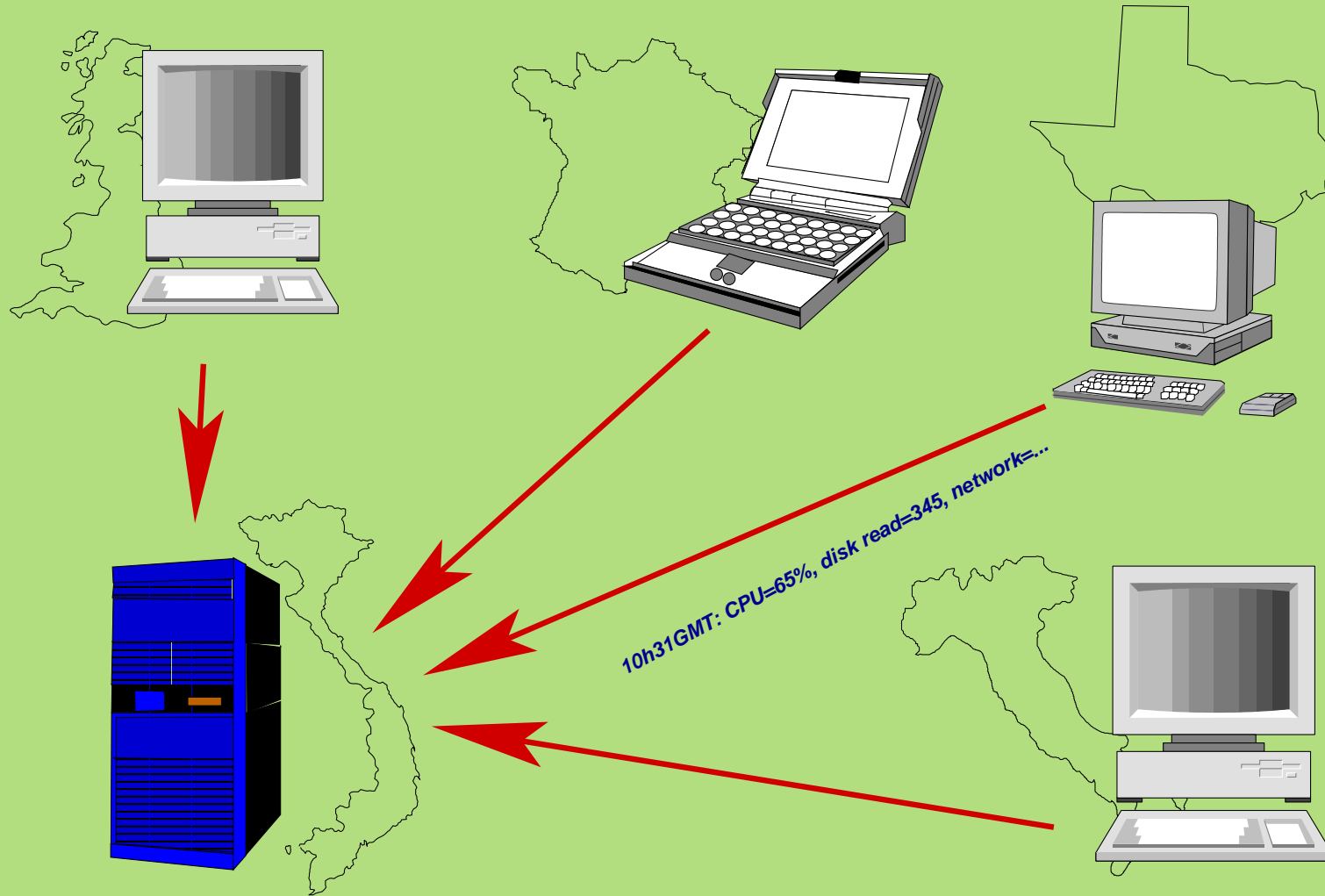
➡ <http://www.p-grid.org>

➡ Architecture : pas de contrôle global.





FOUILLE : CONTEXTE ARCHITECTURAL



X Network Time Protocol (NTP) synchronise le temps d'un client ;

X Organisation hiérarchique ;

```
[root@m24 etc]# crontab -e
```

```
5 * * * * /usr/local/bin/ntpdate ntp.obspm.fr
```

Serveurs secondaire de temps de l'Observatoire de Paris.

X NTP assure une précision de 10ms ;

X <http://www.ntp.org/>

