

# Architectures Spécialisées pour la Bioinformatique

Dominique Lavenier  
IRISA / CNRS

# Bioinformatique

## Définition :

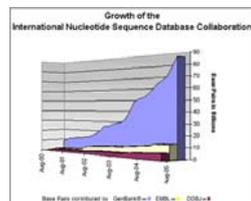
Le terme *bioinformatique* s'applique aux méthodes statistiques et aux algorithmes utilisés pour résoudre les problèmes posés par l'analyse et la gestion des **données génomiques**



# Croissance des banques génomiques

Croissance  
exponentielle

~ X 2 tous les 16 mois



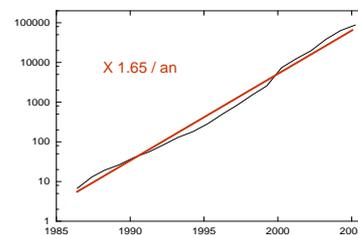
Août 2005 : taille =  $100 \times 10^9$  nucléotides -  $55 \times 10^6$  séquences - 165000 organismes

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

3

# banque de données de séquences d'ADN (GENBANK)

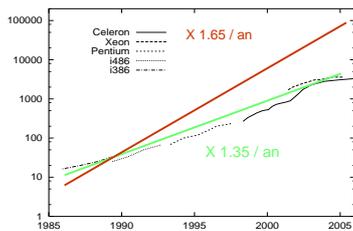


19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

4

## Performances des microprocesseurs (fréquence)



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

5

## Recherche dans les banques

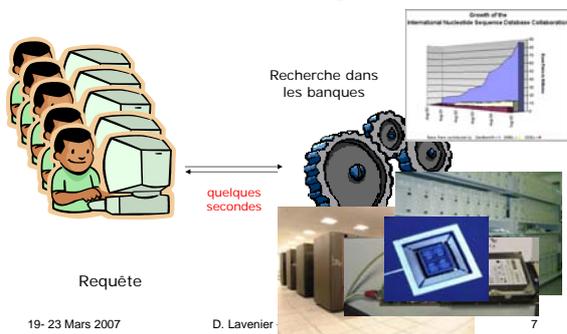
- Opération de base de la génomique
  - représente 70 % des calculs des serveurs bioinformatique
- Recherche par le contenu
  - annotations : peu fiables, pas à jour
- Traitement basé sur la comparaison de séquences
  - concept de similarité, distance

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

6

## Challenge



19- 23 Mars 2007

D. Lavenier

7

## Plan de la présentation

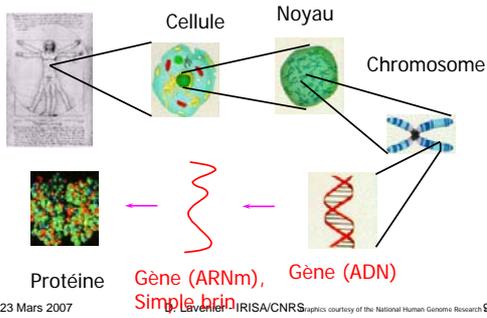
- Contexte
  - Génome, ADN, gènes, comparaison de séquences, ...
- Algorithme de base
  - Principe
  - Parallélisation
  - Accélérateur systolique
- Heuristique
  - Principe
  - Mise en œuvre matérielle
- Indexation des données
  - Principe
  - Le projet ReMIX

19- 23 Mars 2007

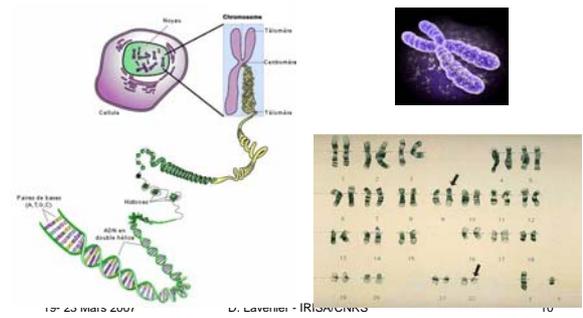
D. Lavenier - IRISA/CNRS

8

## Organisme → molécule

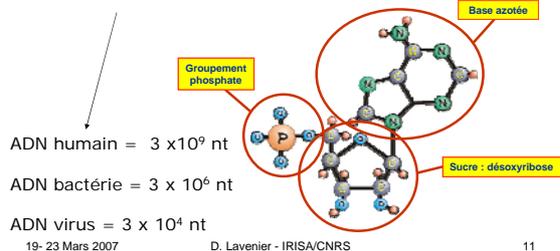


## Cellule - Chromosome

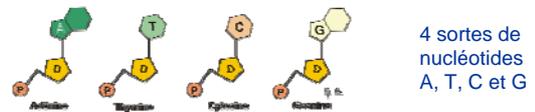


## ADN

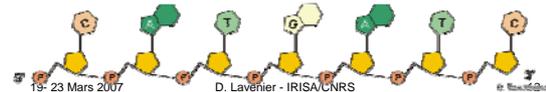
- ADN : acide désoxyribonucléique
- suite de nucléotides



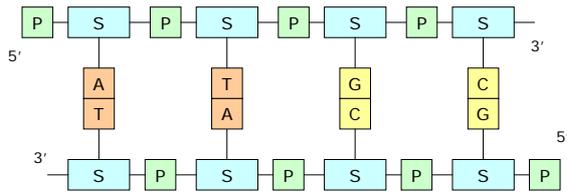
## Nucléotides



Les nucléotides peuvent se lier les uns aux autres par leur sucre (désoxyribose) et leur groupement phosphate.



## ADN : représentation symbolique



Par convention, dans les banques : texte du brin 5' → 3'

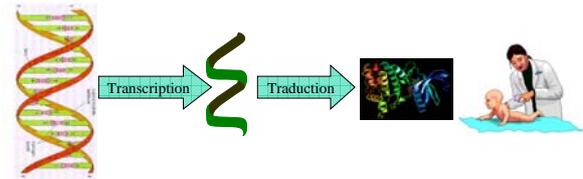
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

13

## Le dogme central de la biologie moléculaire

ADN → ARN → Protéine → Phénotype



Gene (ADN)

ARNm  
(ARN messenger)

Protéine

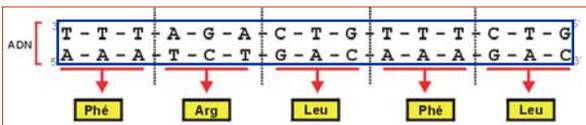
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

14

## Gène

Un segment d'ADN portant toute l'information nécessaire pour la synthèse d'une protéine = **gène**



Gène de de la protéine **Phé-Arg-Leu-Phé-Leu**

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

15

## Génome

Génome humain

→ toute l'information nécessaire pour fabriquer un homme

→ **30 000 à 40 000 gènes**

Chaque cellule (sauf gamètes reproducteurs) contient **deux exemplaires** du génome humain (un qui vient du père et l'autre de la mère).

**DONC** chaque cellule contient 46 chromosomes

Le nombre de chromosomes est variable selon l'espèce

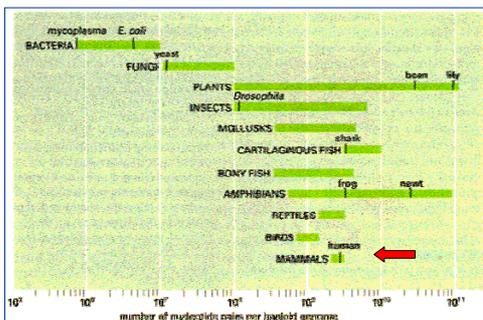
Chien.....78  
Chat.....38  
Rat.....42

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

16

## Taille des génomes



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

17

## Structure des gènes

Chez les **procaryotes**, presque tout l'ADN code pour des protéines.

Chez les **eucaryotes**, seulement 3% de l'ADN code pour des protéines ou des ARN r ou t.

Le 97% restant = "**junk**" DNA (ou "**garbage**" DNA)

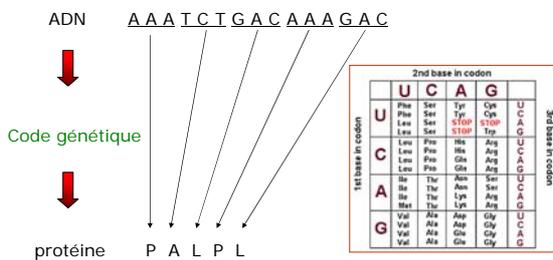
Il serait plus prudent de parler plutôt d'**ADN non codant**

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

18

## ADN → Protéine

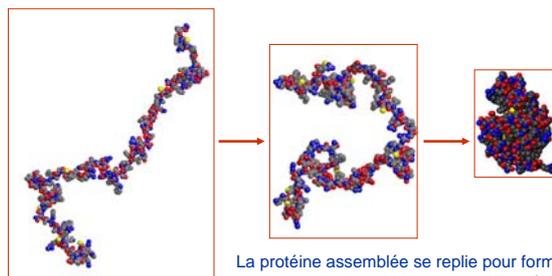


19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

19

## Structure tertiaire

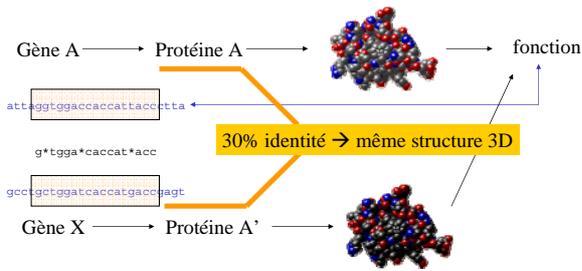


19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

20

## Gène : relation texte/fonction



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

21

## Plan de la présentation

- Contexte
  - Génome, ADN, gènes, comparaison de séquences, ...
- **Algorithme de base**
  - Principe
  - Parallélisation
  - Accélérateur systolique
- Heuristique
  - Principe
  - Mise en œuvre matérielle
- Indexation des données
  - Principe
  - Le projet ReMIX

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

22

## Comparaison de séquences

- Tâche informatique la plus fréquemment exécutée par les biologistes.
  - Il s'agit de déterminer dans quelle mesure deux séquences se ressemblent.
  - Motivation première :
    - inférer des connaissances sur une séquence à partir des connaissances attachées à une autre
  - Autres usages :
    - Construction d'arbres phylogéniques
    - Assemblage
- Recherche dans les banques

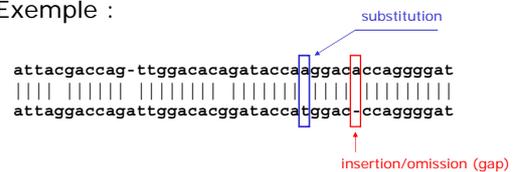
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

23

## Comparer des séquences ?

- Comparer 2 séquences revient à rechercher un alignement
- Exemple :



- 2 opérations élémentaires

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

24



## Algorithme de Smith et Waterman

match M = 2  
mismatch N = -1  
gap = 2

$$S_{i,j} = \max \begin{cases} 0 \\ S_{i,j-1} - \text{Gap} \\ S_{i-1,j-1} + \text{Sub}(X_i, Y_j) \\ S_{i-1,j} - \text{Gap} \end{cases}$$

Complexité :  $N^2$

*taille des séquences*

	A	T	A	C	G	T	A	T	T	G	C
G	0	0	0	0	2	0	0	0	0	2	0
G	0	0	0	0	2	1	0	0	0	0	1
C	0	0	0	2	0	1	0	0	0	0	2
G	0	0	0	0	4	2	0	0	0	2	0
A	2	0	2	0	2	3	4	2	0	0	0
T	0	4	0	1	0	4	2	6	4	2	0
T	0	2	3	0	0	2	3	4	8	3	1
A	2	0	4	2	0	0	4	2	6	7	2
A	2	1	2	3	1	0	2	3	1	5	6
T	0	2	0	0	2	3	1	4	5	3	4

## Implémentation séquentielle

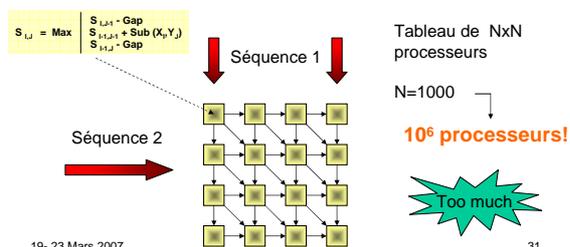
- Banque =  $10 \times 10^9$  nucléotides
- Requête =  $10^3$  nt

$$S_{i,j} = \max \begin{cases} 0 \\ S_{i,j-1} - \text{Gap} \\ S_{i-1,j-1} + \text{Sub}(X_i, Y_j) \\ S_{i-1,j} - \text{Gap} \end{cases} = 10^{-9} \text{ sec}$$

- Temps de calcul =  $10^4$  sec = 3 heures

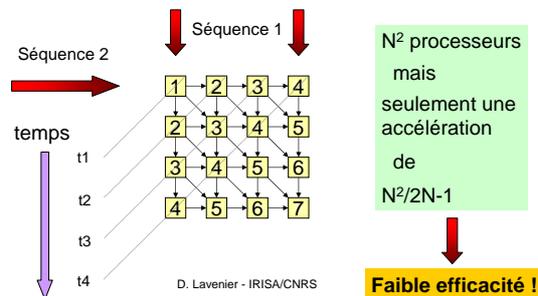
## Architecture spécialisée : idée

- Associer un processeur dédié au calcul de chaque case de la matrice



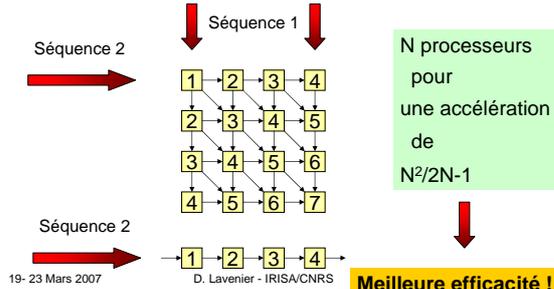
## Dépendance de données

- Seulement 1 diagonale du tableau est active



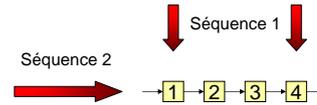
## Réseau 1D

un processeur émule une colonne



## Réseau 1D

- Accélération optimale : parcours des banques



- Séquence 2 = banque

$$\text{Accélération} = \frac{N \times M}{N+M-1} = N$$

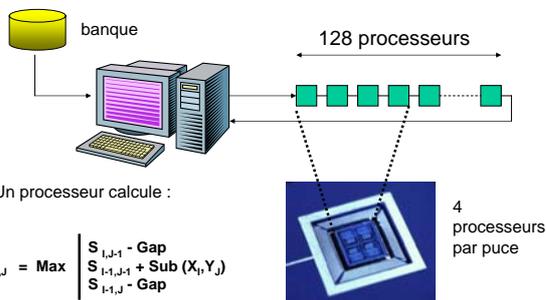
N = longueur de Seq 1  
M = longueur de Seq 2  
M >> N

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

34

## Accélérateur Samba (1995)



## Accélérateurs commerciaux

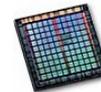
Time Logic : DeCypher accelerator

- Technologie FPGA
- *parallel "reconfigurable" computer*



Paracel : GeneMatcher

- Technologie ASIC
- Réseau 1D de quelques milliers de processeurs



CLCbio : CLC bioinformatics cube

- Technologie FPGA
- performance : x 125 / proc 3 GHz



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

36

## Plan de la présentation

- Contexte
  - Génome, ADN, gènes, comparaison de séquences, ...
- Algorithme de base
  - Principe
  - Parallélisation
  - Accélérateur systolique
- Heuristique
  - Principe
  - Mise en œuvre matérielle
- Indexation des données
  - Principe
  - Le projet ReMIX

## Peut-on faire mieux ?

- Du côté logiciel :
    - début 90 → heuristique « BLAST »
  - Alignement :
    - caractéristique : dans la majorité des cas on observe des mots identiques de  $W$  caractères
- ```

attacgaccag-ttggacacagataccaaggacaccaggggat
|||||  |||||  |||||  |||||  |||||  |||||  |||||  |||||
Attaggaccagattgacacacagataccatggac-ccaggggat
    
```
- Recherche des alignements à partir de ces points d'ancrage
    - diminution importante de l'espace de recherche

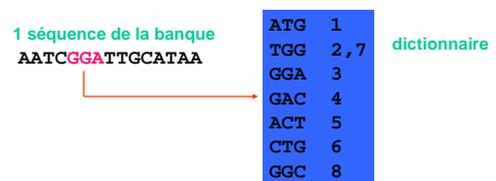
## Principe de l'heuristique (1)

- 1 - la séquence requête est « tronçonnée » en mots de  $W$  caractères que l'on stocke dans un dictionnaire



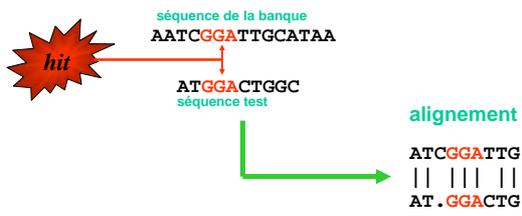
## Principe de l'heuristique (2)

- 2 - pour chaque séquence de la banque on détecte les mots de  $W$  caractères (*hits*) qui appartiennent au dictionnaire



## Principe de l'heuristique (3)

3 - lorsqu'un *hit* est trouvé on « l'étend » pour produire un alignement

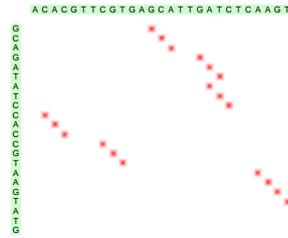


19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

41

## Illustration : étape 1



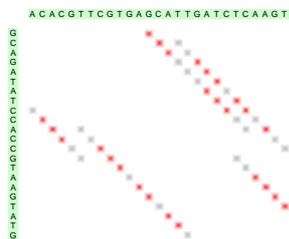
Localisation  
des mots de W characters  
(seeds, hits, ...)

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

42

## Illustration : étape 2



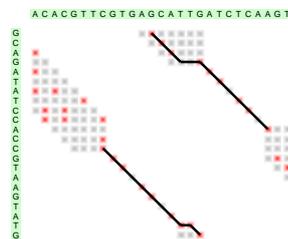
Extension sans gap

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

43

## Illustration : étape 3



Extension avec gap

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

44

## Pourquoi c'est plus rapide que la programmation dynamique

Exemple : 2 séquences d'ADN de taille 1000 - W=10

Taille max du dictionnaire 1000

Nombre de mots possibles de taille 10  $4^{10}$

Nombre moyen d'apparitions d'un mot de taille 10 dans une séquence de longueur 1000  $10^3 / 4^{10}$

Nombre moyen d'apparitions de 1000 mots de taille 10 dans une séquence de longueur 1000  $10^6 / 4^{10} = 1$

Nombre moyen de hits entre 2 séquences de longueur 1000



## Pourquoi c'est plus rapide que la programmation dynamique

Algorithme :

- pour chaque mot de W caractères dans la séquence 1000

- recherche dans le dictionnaire 1

- si le mot existe  $10^3 / 4^{10}$

- étendre le hit 1000

# d'opérations :  $1000 \times (1 + (10^3 / 4^{10}) \times 10^3) = 2 \times 10^3$

À comparer aux  $10^6$  opérations de la programmation dynamique (gain=500)

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

46

## BLAST

- Le programme le plus utilisé en bioinformatique
- **Avantages**
  - Donne des résultats satisfaisants
  - Rapide
- **Inconvénients**
  - Sensibilité dépend du paramètre W
  - Parcours systématique de la banque

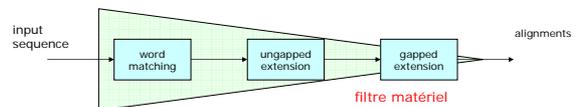
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

47

## BLAST profiling

P. Krishnamurthy et al. 15<sup>th</sup> IEEE ASAP conference, 2004

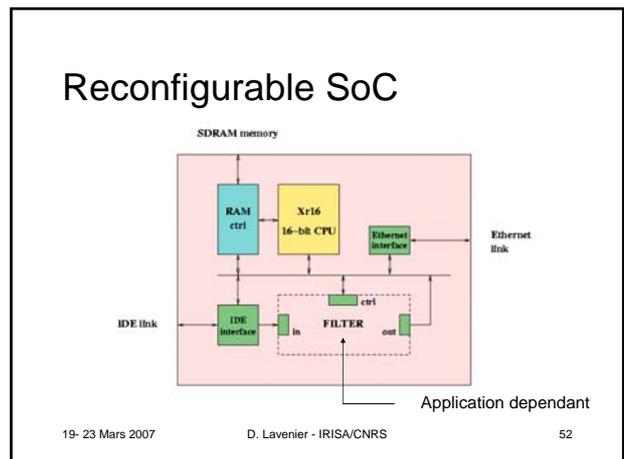
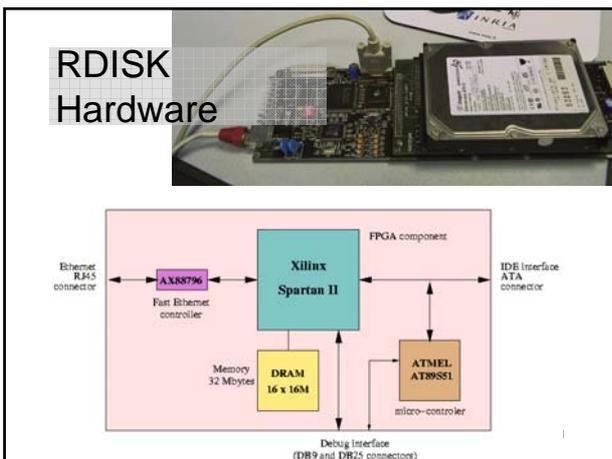
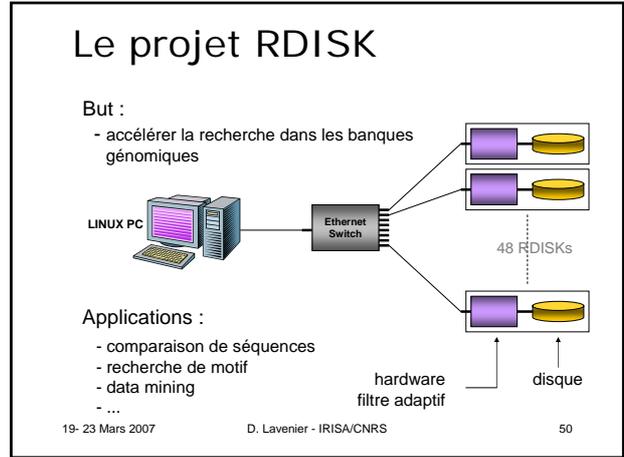
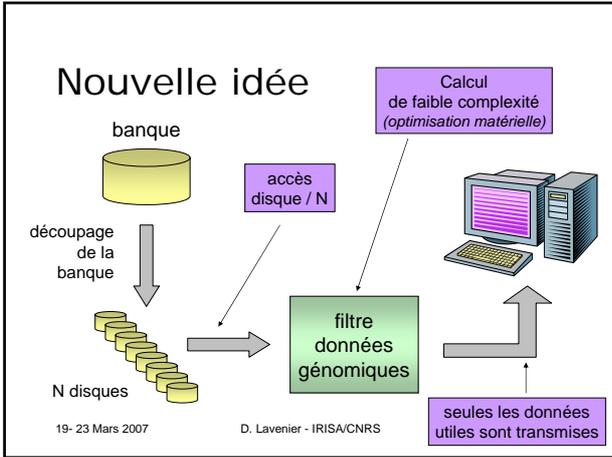


| Query size | word matching | Ungapped extension | Gapped extension |
|------------|---------------|--------------------|------------------|
| 10K        | 86.50 %       | 13.30 %            | 0.20 %           |
| 100K       | 83.30 %       | 16.60 %            | 0.10 %           |
| 1M         | 85.30 %       | 14.65 %            | 0.05%            |

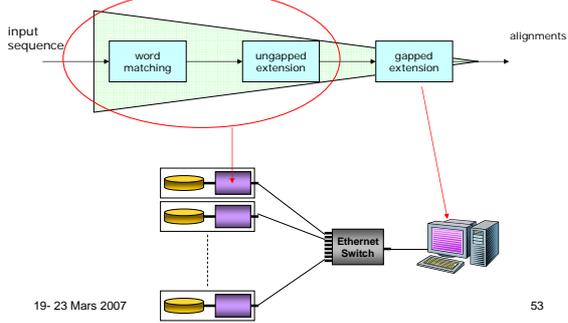
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

48



## Implémentation de BLAST sur RDISK



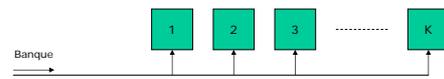
## Implémentation de BLAST

Séquence requête

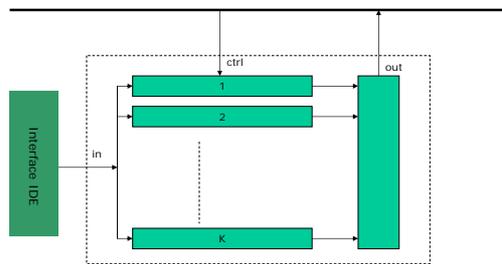
AGGACCATTAGGGACACATTAGGAGGAGGTGGACAGTGTAGGACCAAA

- 1 AGGACCATTAGGGA
  - 2 GACCATTAGGACCA
  - 3 CCATTAGGACATT
  - 4 ATTAGGACATTAG
- K petites sous-séquence

K opérateurs : détection de similarité locale

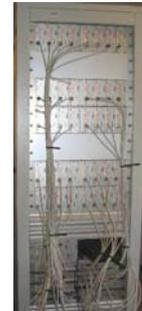


## Implémentation de BLAST



## RDISK system

- 48 node system operational since 2004
- West Genopole server → motif search



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

56

## Performances

- Parcours des bases de données
  - Génome humain : ~3s
  - Recherche par le contenu (Blast)
  - Recherche de motifs complexes
- 1 système RDISK-48
  - Equivalent à un Cluster de 192 PCs
  - 10 fois moins cher (1 carte ~ 250 Euros)

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

57

## Plan de la présentation

- Contexte
  - Génome, ADN, gènes, comparaison de séquences, ...
- Algorithme de base
  - Principe
  - Parallélisation
  - Accélérateur systolique
- Heuristique
  - Principe
  - Mise en œuvre matérielle
- **Indexation des données**
  - Principe
  - Le projet ReMIX

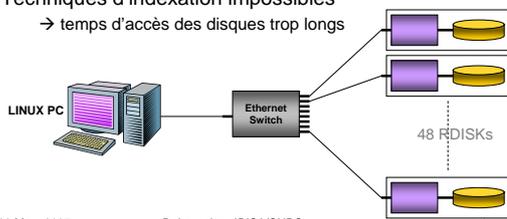
19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

58

## Les faiblesses de RDISK

- Besoin de parcourir toute la banque
  - les calculs sont encore limités par le débit des disques
- Techniques d'indexation impossibles
  - temps d'accès des disques trop longs



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

59

## Indexation des données

- Idée :
  - Plutôt que d'indexer la requête, on indexe les banques!
- Avantage
  - Parcours de la requête ☺
- Inconvénient
  - Taille de l'index ☹
  - Disques trop lents ☹

19- 23 Mars 2007

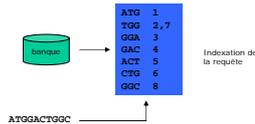
D. Lavenier - IRISA/CNRS

60

## Indexation des données génomiques

### BLAST

- Indexation de la séquence requête
- Parcours de la banque



### iBLAST

- Indexation de la banque
- Parcours de la séquence requête



19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

61

## Taille de l'index

### Exemple :

génomme humain =  $3 \times 10^9$  caractères  
 → 12 Go (chaque position = 1 entier)

Ne tient pas en mémoire centrale

banque EST =  $50 \times 10^9$  caractères  
 → 200 Go



Concevoir une mémoire spécialisée  
 → de grande taille  
 → adaptée à l'indexation

↳ Accès rapide aux données

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

62

## Projet ReMIX

mémoire reconfigurable pour l'indexation

### • Objectifs

- Capacité : plusieurs centaines de Go
- Stockage spécifique de « gros » index
- Pas une simple extension mémoire
  - accès à n'importe quelle donnée à coût fixe : pas de cache
  - accès indirect : une requête en lecture déclenche des traitements (simples, mais coûteux) sur les données

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

63

## Technologie mémoire

Estimation pour une mémoire de 64 Go

| technologie | Coût en Euro (2005) | Temps d'accès (ns) | Bande Passante (Mo/sec) | Consommation (W) |
|-------------|---------------------|--------------------|-------------------------|------------------|
| SRAM        | 125 000             | 5                  | 800                     | 5230             |
| SDRAM       | 4000                | 10                 | 2000                    | 35               |
| FLASH NOR   | 20000               | 100                | 320                     | 2                |
| FLASH NAND  | 2000                | 25000              | 160                     | 2                |

19- 23 Mars 2007

D. Lavenier - IRISA/CNRS

64

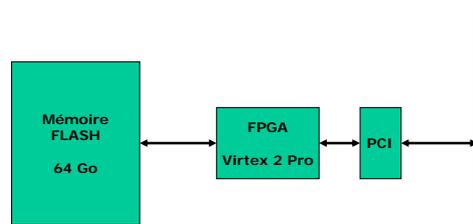
## Traitement des données

- Essentiellement du filtrage (cf RDISK)
- On accède – via l'index – à un paquet de données
- On filtre ces données pour ne renvoyer que l'information pertinente

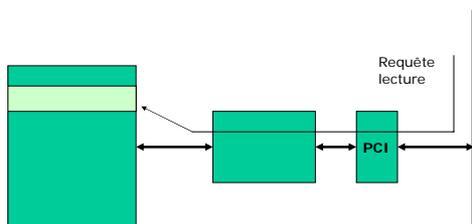


Opérateur Hardware Reconfigurable

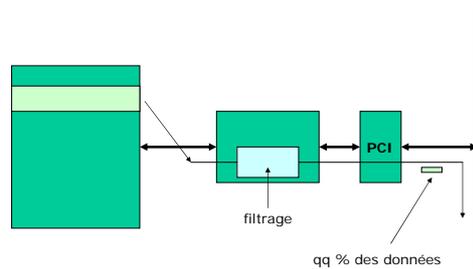
## Architecture

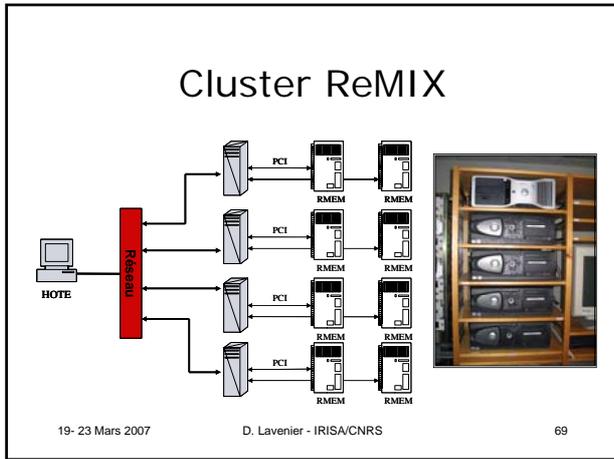


## Fonctionnement



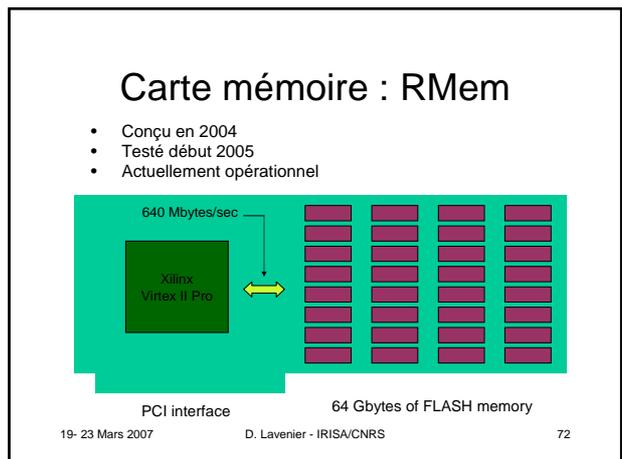
## Fonctionnement



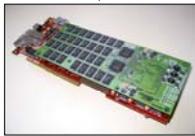
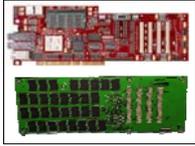


- ## Cluster ReMIX (suite)
- 1 Nœud frontal (Dell Precision 380) :
    - Pentium IV 3.6GHz, 4Go RAM, DD 500Go, Réseau Gbit
  - 4 Nœuds de calcul (Dell Precision 370) :
    - Pentium IV 3GHz, 2Go RAM, DD 320Go, Réseau Gbit
  - 400Go sur le nœud frontal, et 130Go sur les nœuds de calcul pour stocker de bases de données
  - Toutes les machines sont connectées sur le réseau de l'IRISA
  - A terme : création d'un réseau local dédié
- 19- 23 Mars 2007      D. Lavenier - IRISA/CNRS      70

- ## Fonctionnement du cluster
- Application principale tournant sur le nœud frontal
  - Bases de données réparties/dupliquées sur les cartes RMEM
  - Requête(s) de recherche spécifiée(s) par l'application principale
  - Découpage en sous requêtes sur la base indexée
  - Envoi des sous-requêtes aux RMEM concernées
  - Traitement local des sous requêtes
  - Résultats partiels renvoyés par les RMEM
  - Fusion et affichage sur le nœud frontal
  - Fonctionnement géré par un framework JAVA (F. Raimbault)
- 19- 23 Mars 2007      D. Lavenier - IRISA/CNRS      71



## Carte RMEM



### Achat d'une carte FPGA

- FPGA Xilinx V2PRO
- 128Mo DDR SDRAM
- 32Mo SDRAM
- 16Mo Flash
- 2Mo SRAM
- PCI 32b/33MHz
- Ethernet Gb

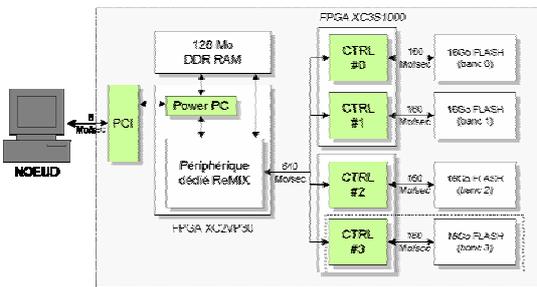
### Développement d'une carte mémoire

- 64Go Nand Flash  
(4 bancs de 16Go)
- 2 FPGAs

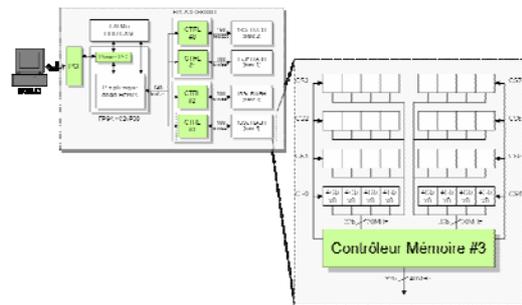
## Technologie mémoire Nand Flash

- Mémoire série, **organisation en blocs et pages**
  - Lecture et écriture de pages (2ko)
  - Effacement de blocs (64 pages)
- **Accès aux pages non immédiat** : Passage par un buffer
- **Inconvénients** :
  - **Latence d'accès aléatoire (25µs)**
  - Existence de **blocs défectueux**
  - Nombre de **cycles d'effacement limité (100k)**

## Organisation d'une carte RMEM



## Organisation d'une carte RMEM





## Conception de l'opérateur

- Conception dans le **respect** des **protocoles** et de l'**interface**
- Fourniture d'un **environnement de simulation**
- Utilisation d'un **makefile** fourni pour :
  - **Synthèse** de l'opérateur seul
  - **Intégration** dans le design complet
  - **Implémentation** du design complet

=> **Bitstream** prêt pour la **configuration** du FPGA **V2PRO**

## Spécialisation du software

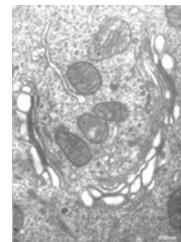
- **Deux modes** de fonctionnement :
  - Mode « Filtrage unique »
  - Mode « Requêtes multiples », plus performant
- **Filtrage unique** :
  - Commande directe de l'opérateur depuis l'hôte
  - Pas de spécialisation
- **Requêtes multiples** :
  - Envoie au PC d'un ensemble de requêtes de filtrage
  - Requête = Paramètres opérateur + adresses taille données
  - Description des requêtes (contenu d'une structure)
  - Écriture des routines de traitement associées

## Environnement

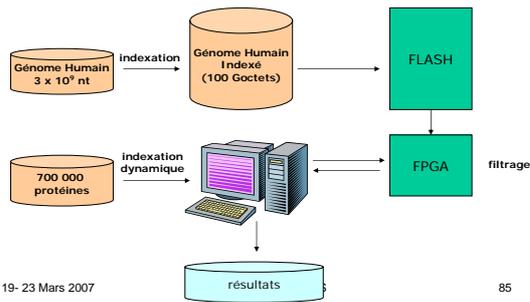
- **Librairie dynamique ReMIX** fournissant des fonctions de haut niveau pour :
  - **Gérer la carte** (configuration du FPGA, reset ...)
  - **Gérer la mémoire Flash** (lecture, écriture, effacement)
  - **Utiliser un opérateur spécialisé** (chargement, paramétrage, démarrage, arrêt, récupération de résultats ...)
- **Gestion des données** :
  - **Vision mémoire** : Adressage direct des bancs de mémoire Flash
  - **Vision fichier** : Système de fichiers dédié ReMIX
- Librairie exploitable par une application locale ou le framework

## Exemple d'utilisation de ReMIX

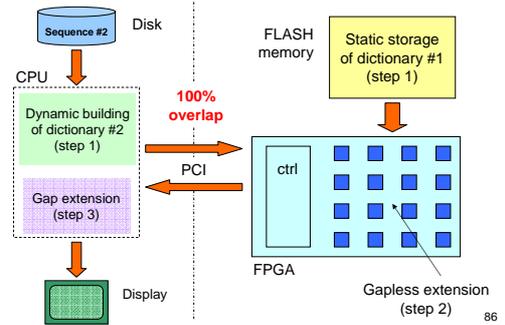
- Découverte de nouveaux gènes mitochondriaux
- Réalisé en partenariat avec l'équipe INSERM U694 du CHU d'Angers
- Comparaison du génome Humain contre 700000 protéines de toutes les bactéries séquencées
- En 2004 → 2 ans de calcul sur un serveur bi-pro (400000 protéines) avec tblastn



## Mise en Œuvre

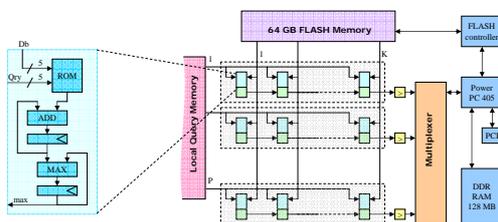


## Fonctionnement



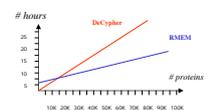
## Opérateur de Filtrage

*tableau de 8 x 20 processeurs*



## Résultats

- Performances
  - Opérateur
    - 160 x 10<sup>6</sup> comparaisons/sec  
(équivalent à 30 G opérations/sec)



- Calcul sur une seule carte RMEM
  - Chromosome par chromosome
  - rechargement de la FLASH
  - Temps de calcul : ~2 semaines

## Conclusion

- Solutions principalement investiguées
  - ASIC
  - FPGA - FPGA / FLASH
- Avantages
  - Performant : speed up 10 → 100
  - Economique (vs cluster)
- Inconvénients
  - Temps de développement
  - Portabilité limitée
  - Marché restreint

## Perspectives

- Autres alternatives
  - Multi-cœurs
    - demain : des puces à 16, 32, 64, ... cœurs
  - Tableau de processeurs spécialisés
    - CELL
    - CLEARSTREAM
  - GPU  Étude en cours (calcul intensif)