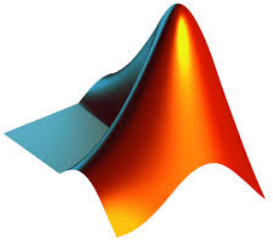
Le HPC appliqué aux données de séquençage

Frédéric Jarlier



2000-2007: The Mathworks SAS (Ingénieur)

2007-2013: CNRS (IR2)

2014: Institut Curie (IR2)

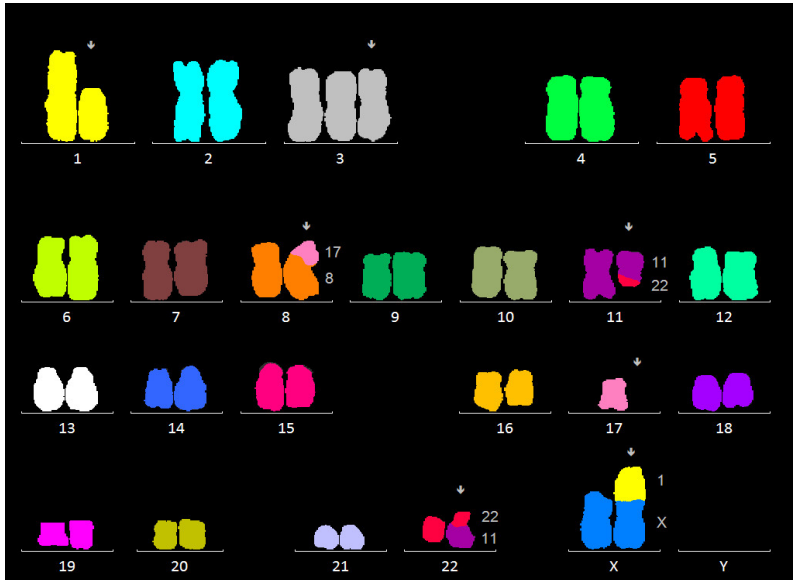
Le HPC appliqué aux données de séquençage

- **Plan**
  - Contexte
  - Optimisations
  - Résultats
  - Conclusion
  - Annexes

Contexte

- Comprendre le cancer c'est comprendre la structure du génome

- Le cancer est une maladie des gènes
  - Ré-arrangement chromosomiques et mutations



CK Rocha et al., Molecular Cytogenetics 2011

Contexte

- Le séquencage du génome, un exemple de pipeline

Illumina HiSeq

1 WGS 30X

Alignement

20h avec 8 threads

Tri

6h sur 32 threads

Variant calling

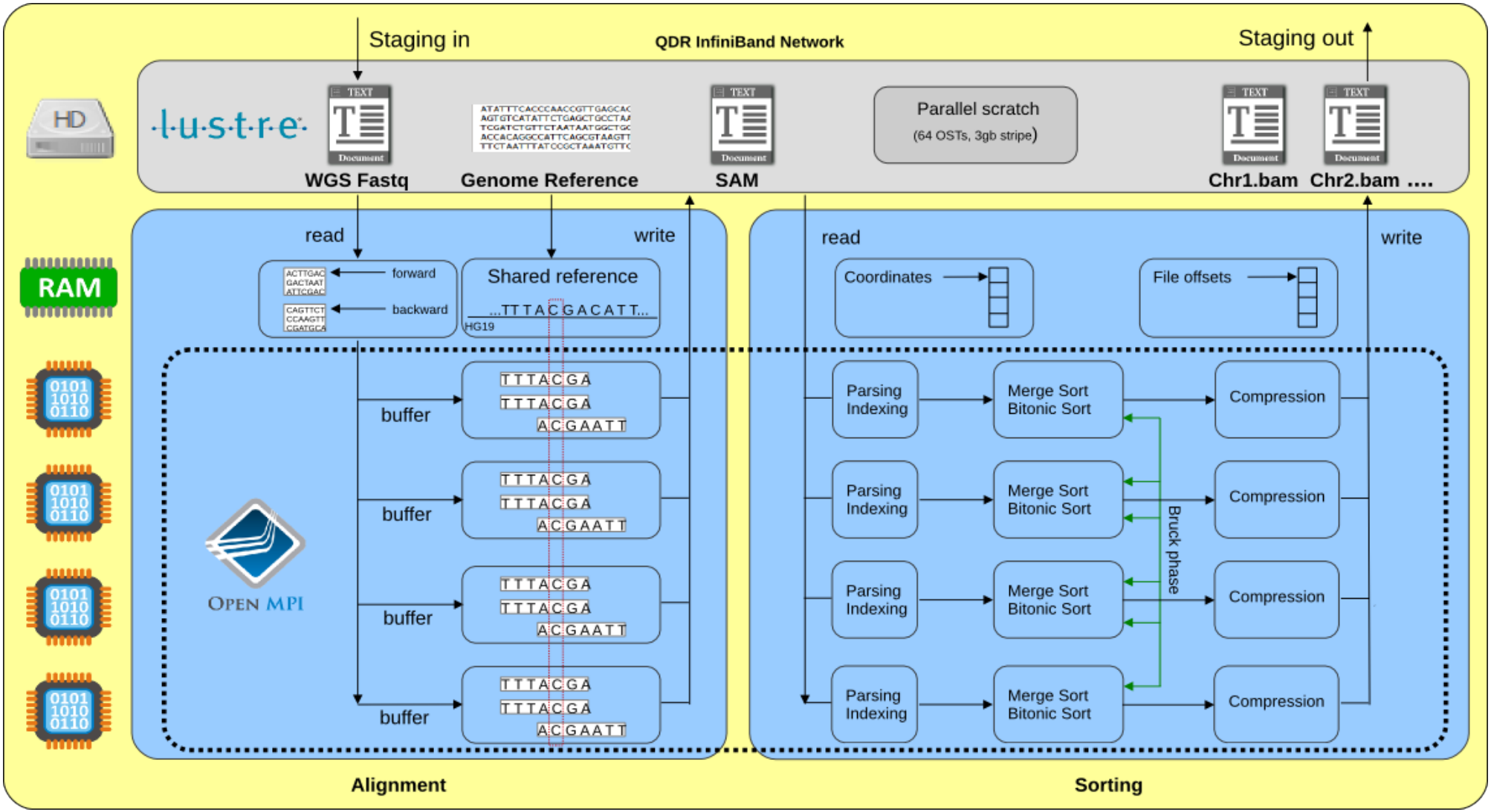
>5 jour

10 séquenceurs 1 WG (30X) / heure => 300 Go / heure

BGI 181 séquenceurs => 6 To par jour

La question : comment analyser des volumes importants de données

- **Le pipeline QUASART**



- **Exemple d'architecture dédiée au calcul parallele**

180 dual processor nodes (Intel Sandy Bridge E5-2680, 2.7 GHz, 8 cores) with

128 GB of memory per node, i.e. a total of 2,880 cores (Bull),

QDR Infiny Band (36 GB/s)

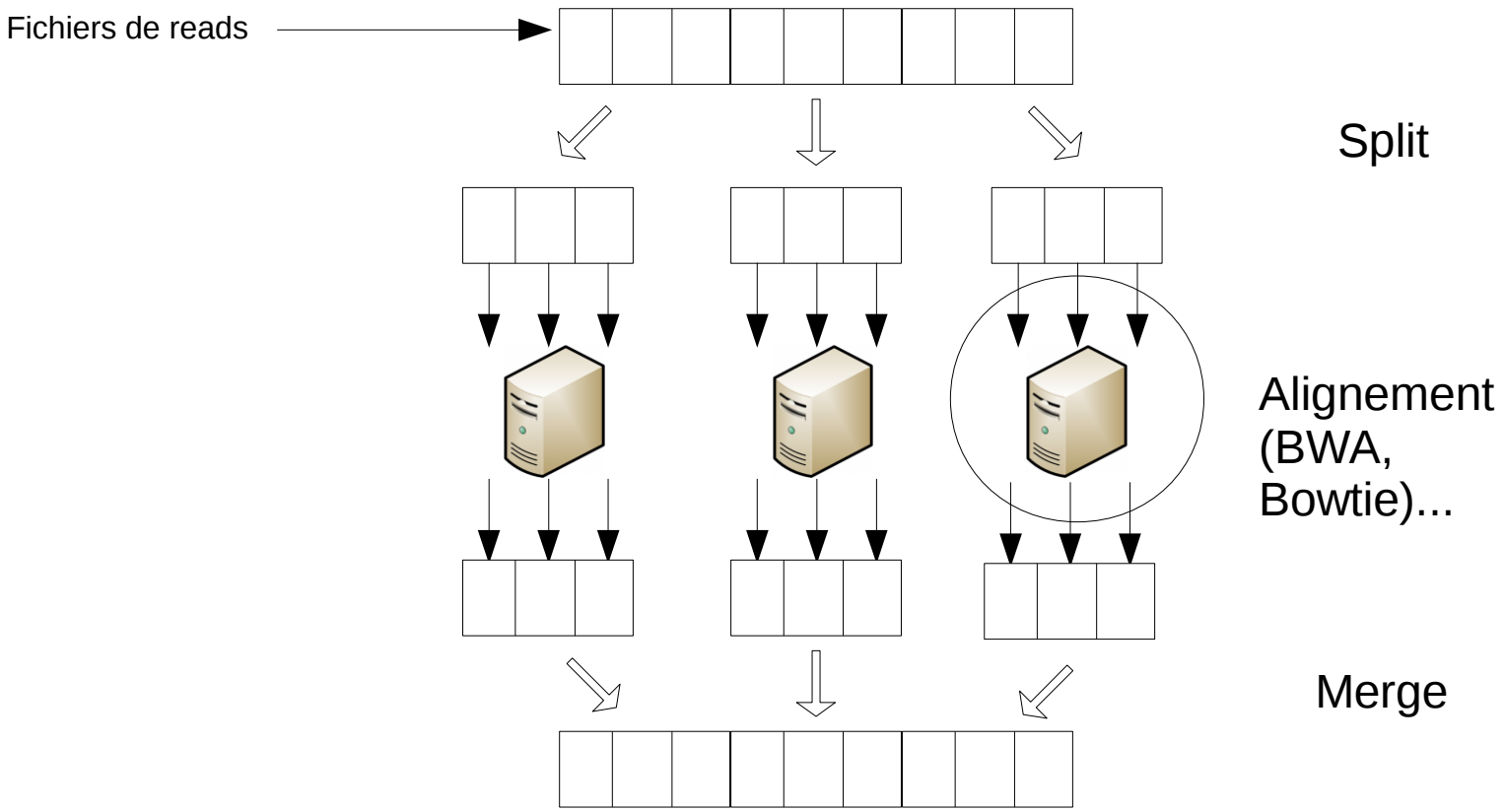
Lustre file System

L'alignement principe

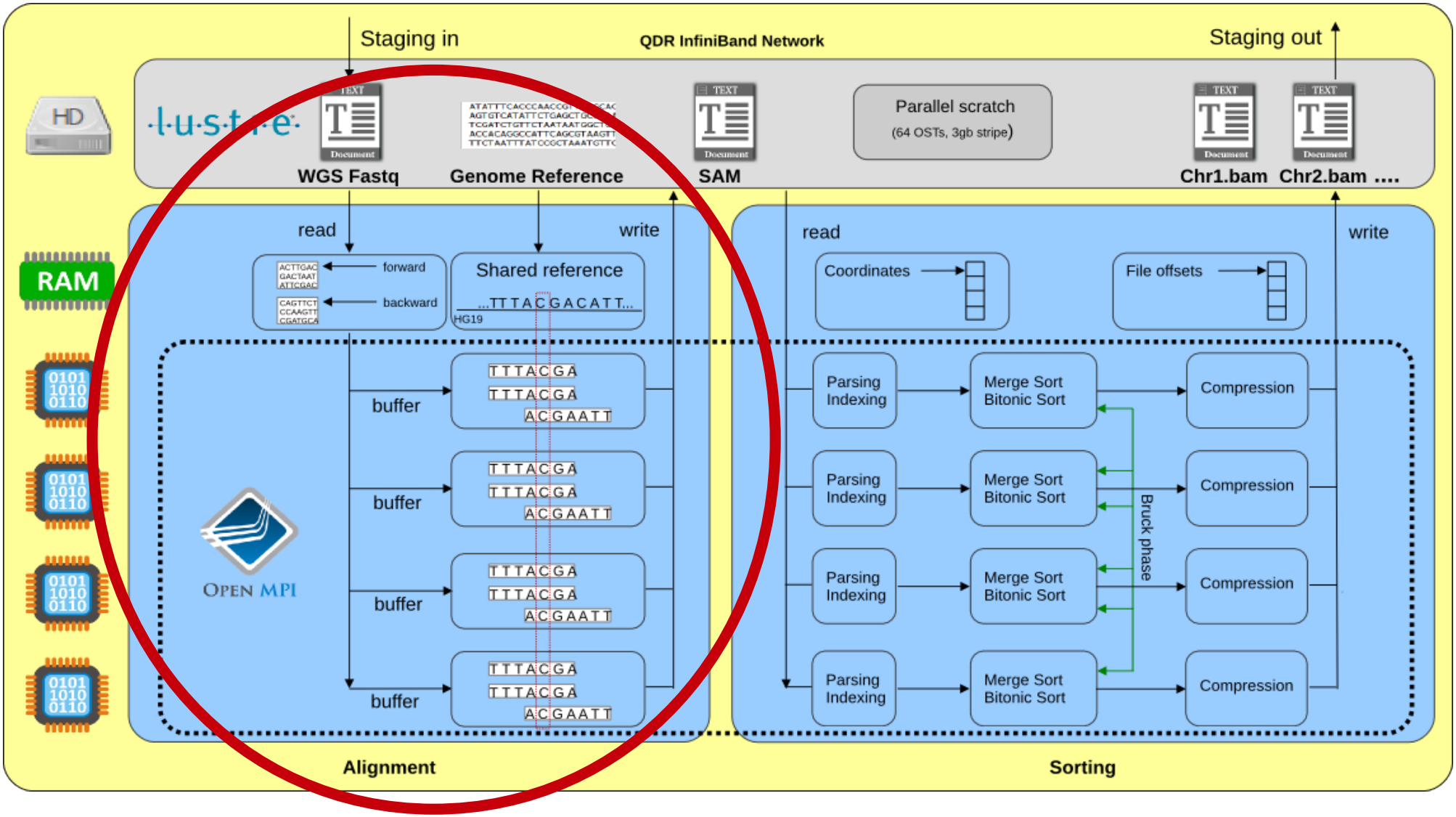
- Trouver les coordonnées des reads sur le genome de référence

- Référence (humaine) = 3,4 milliards de nucléotides
- Des milliards de reads suivant la profondeur du séquençage
- L'alignement est un processus très complexe

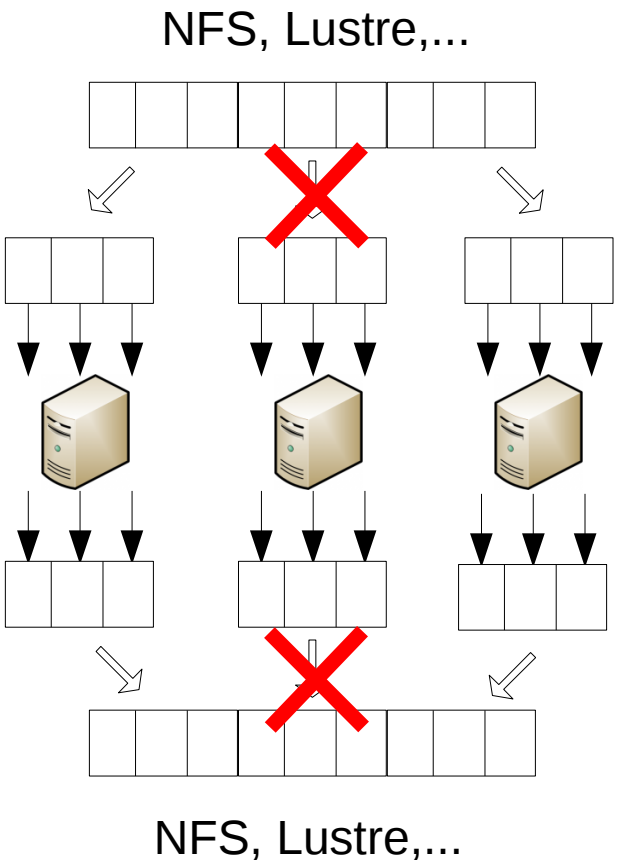
- # L'alignement “Embarrassingly parallel”

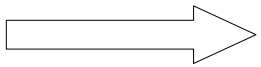


- **Le pipeline QUASART**

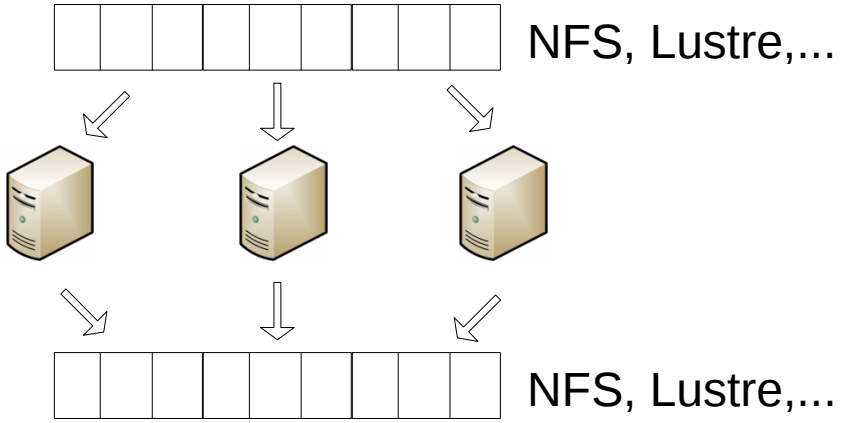


- **Le pipeline QUASART**



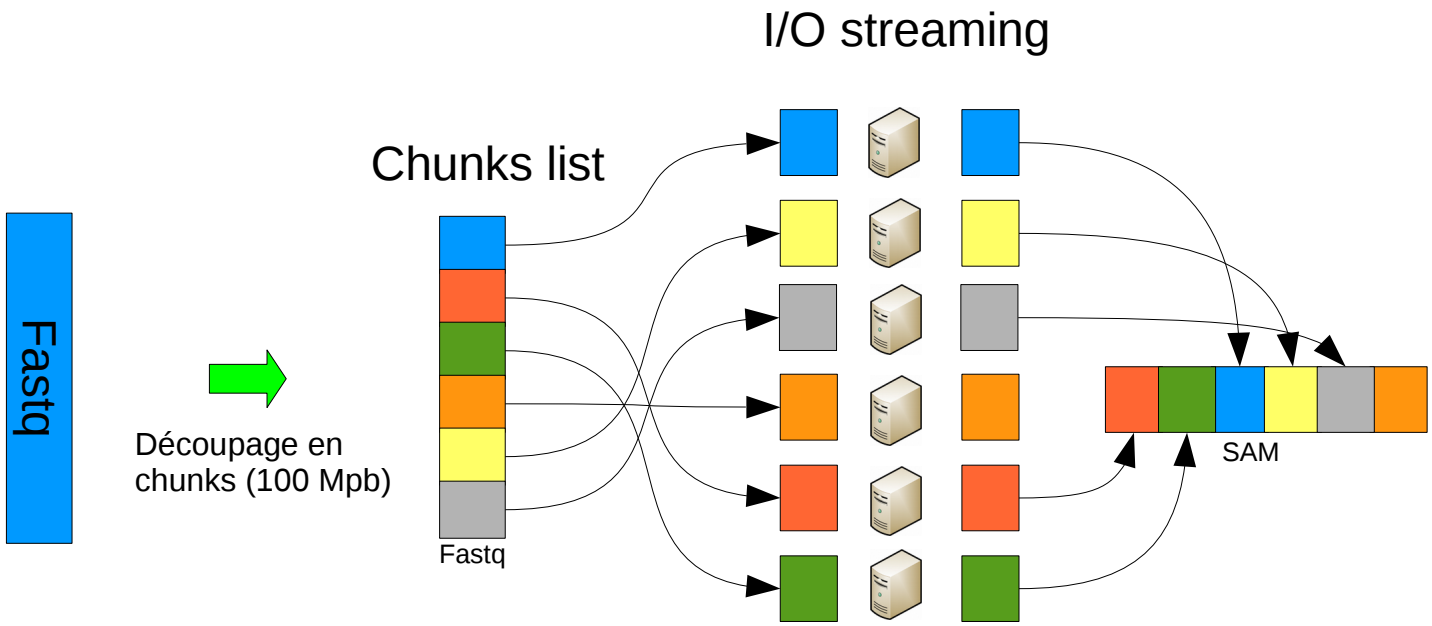


Shared File pointer operations (MPI)



Operation non collective
=> très peu de communications

Le pipeline QUASART

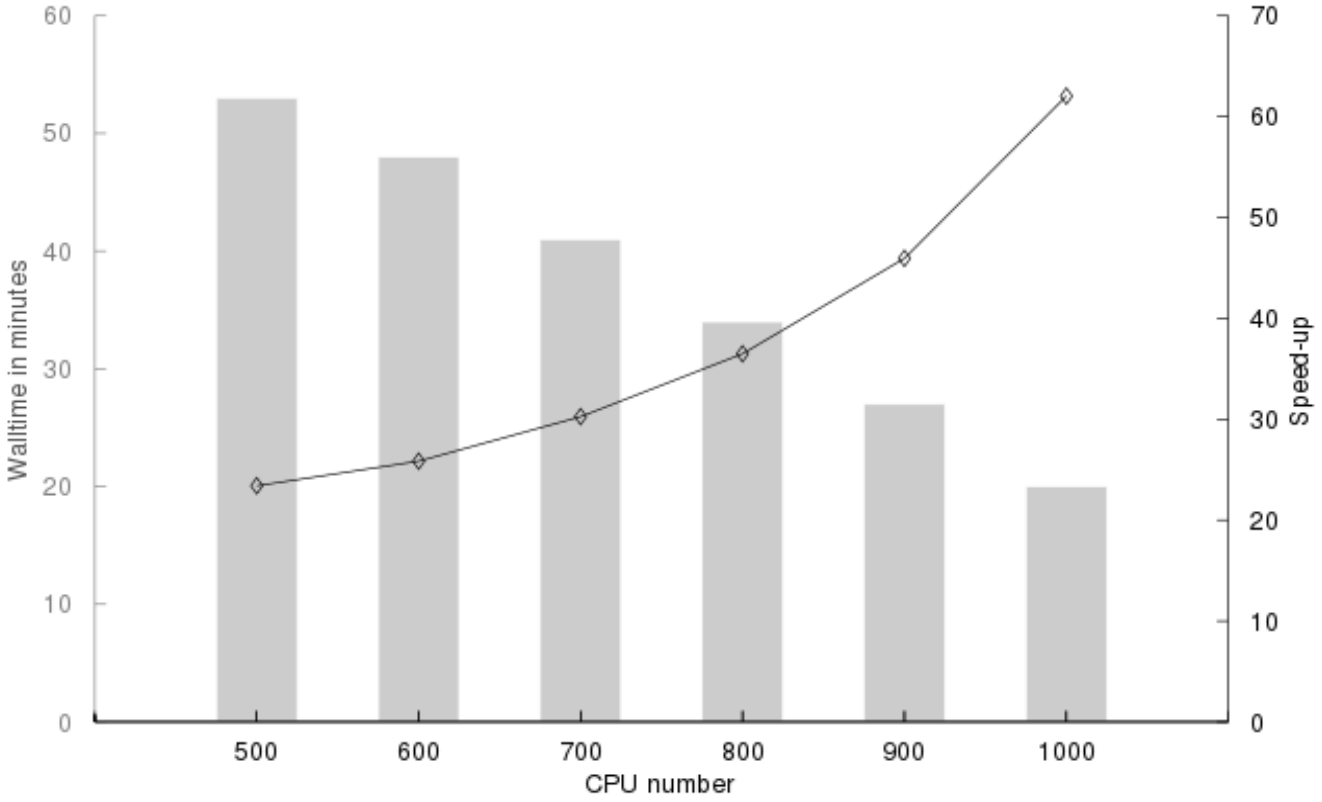


MPI Shared File pointer (Operations non bloquantes)

Supporté par NFS, Lustre,...

- **Le pipeline QUASART**

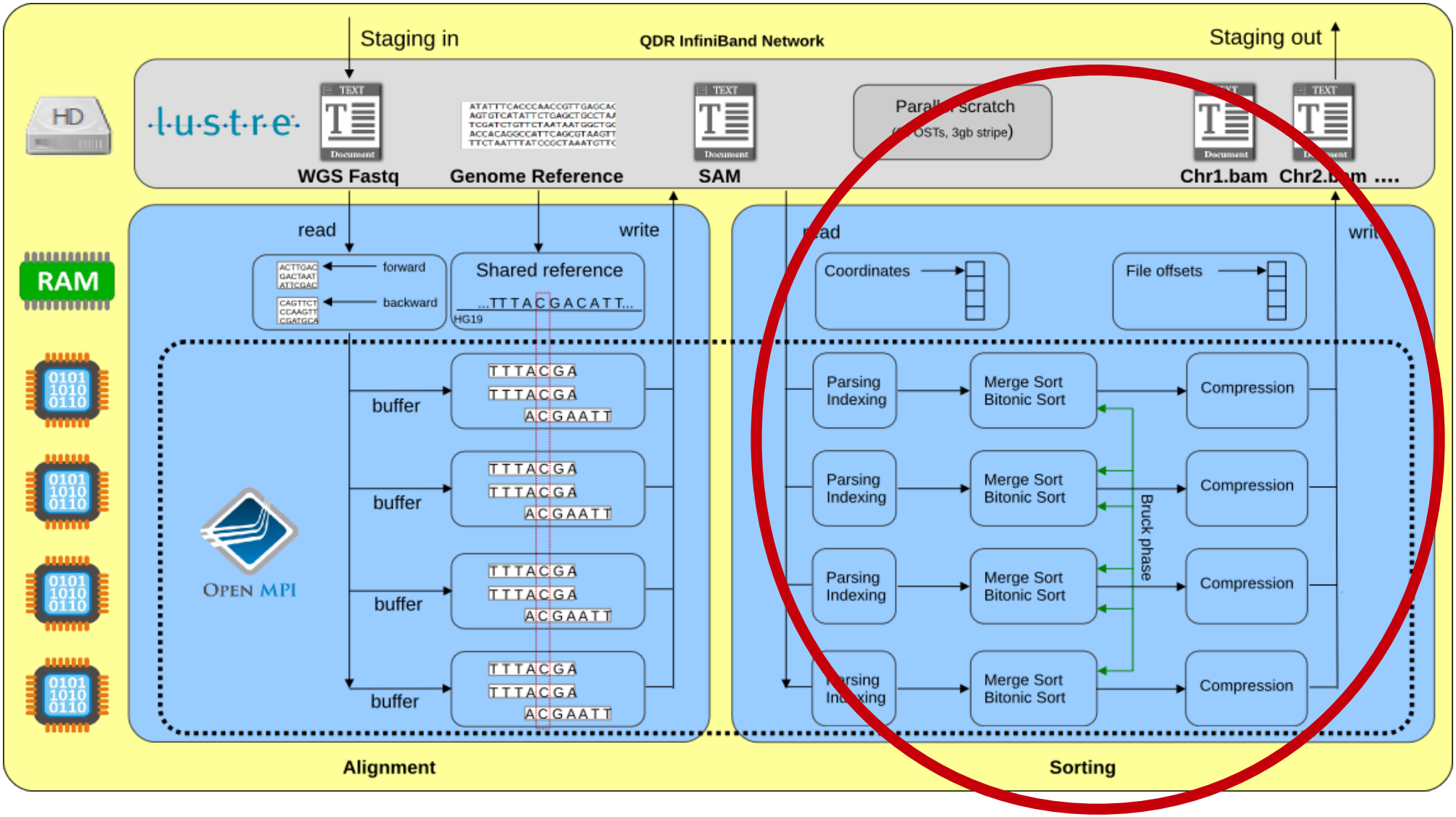
Speed alignment compare with BWA on 16 threads



HCC1187C WG 90X breast ductal carcinoma cell line

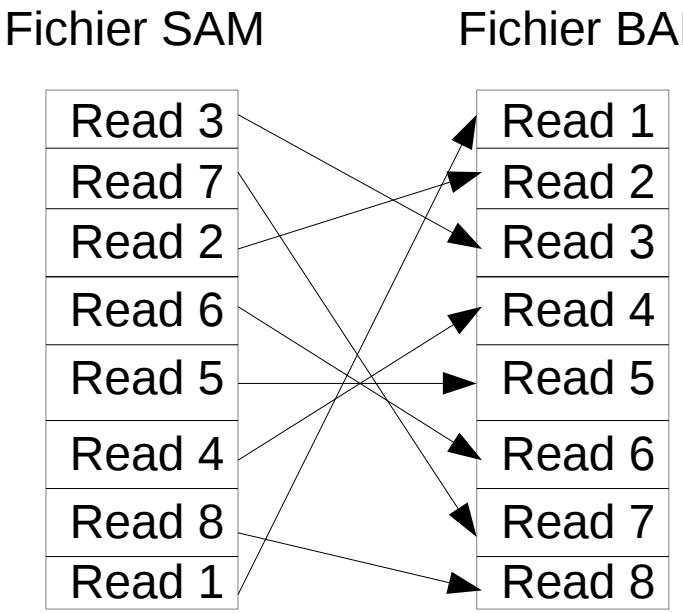
Input = 700 Go
Output = 900 Go

- **Le pipeline QUASART**



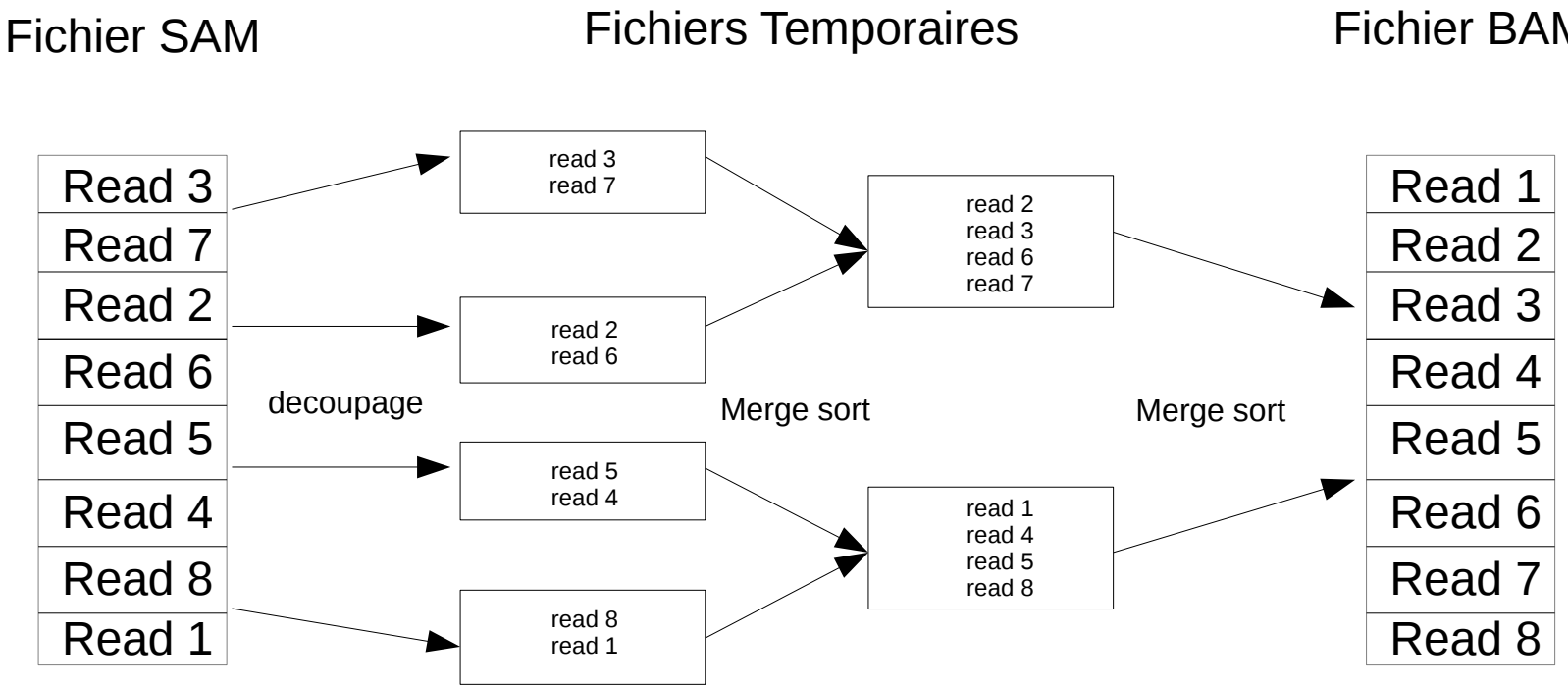
Le principe du tri

Le tri est réalisé sur les coordonées génomiques des reads



Conclusion : Trier c'est déplacer 1To (3 milliars de reads) (1 read = 300 carac. ASCII)

- **Le tri par merge sort**

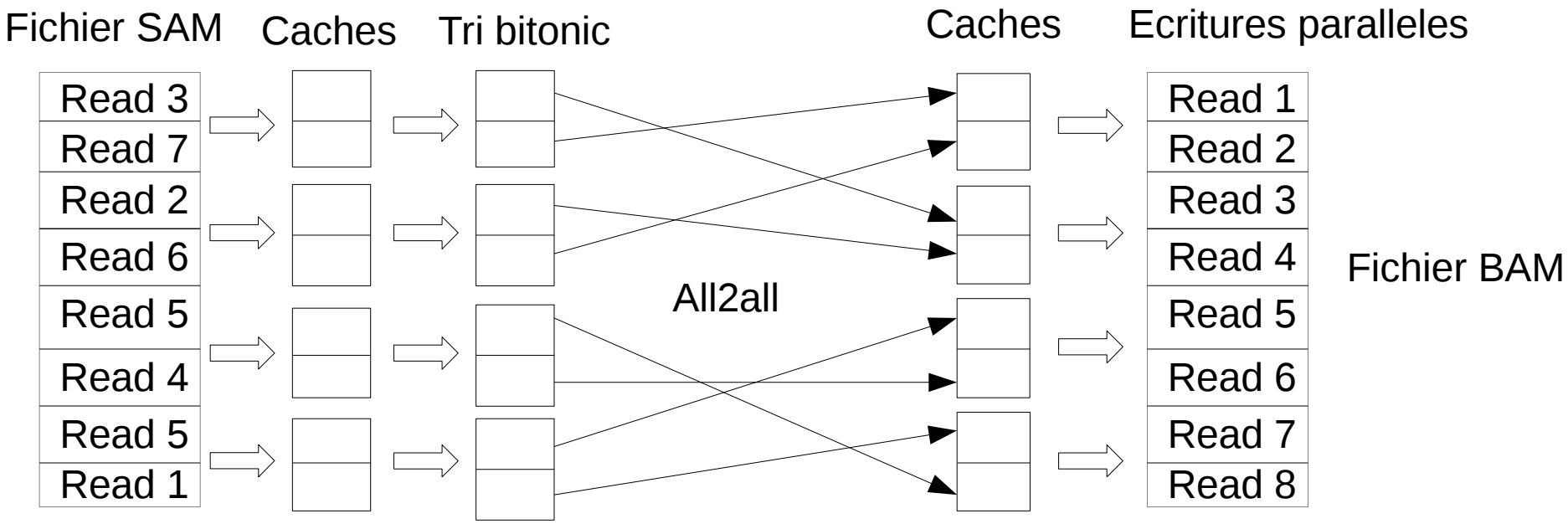


Solutions actuelles utilisent des fichiers temporaires => problème de scalabilité

Samtools, sambamba

Le tri par QUASART

Pour éviter les accés constants au file system on utilise des caches

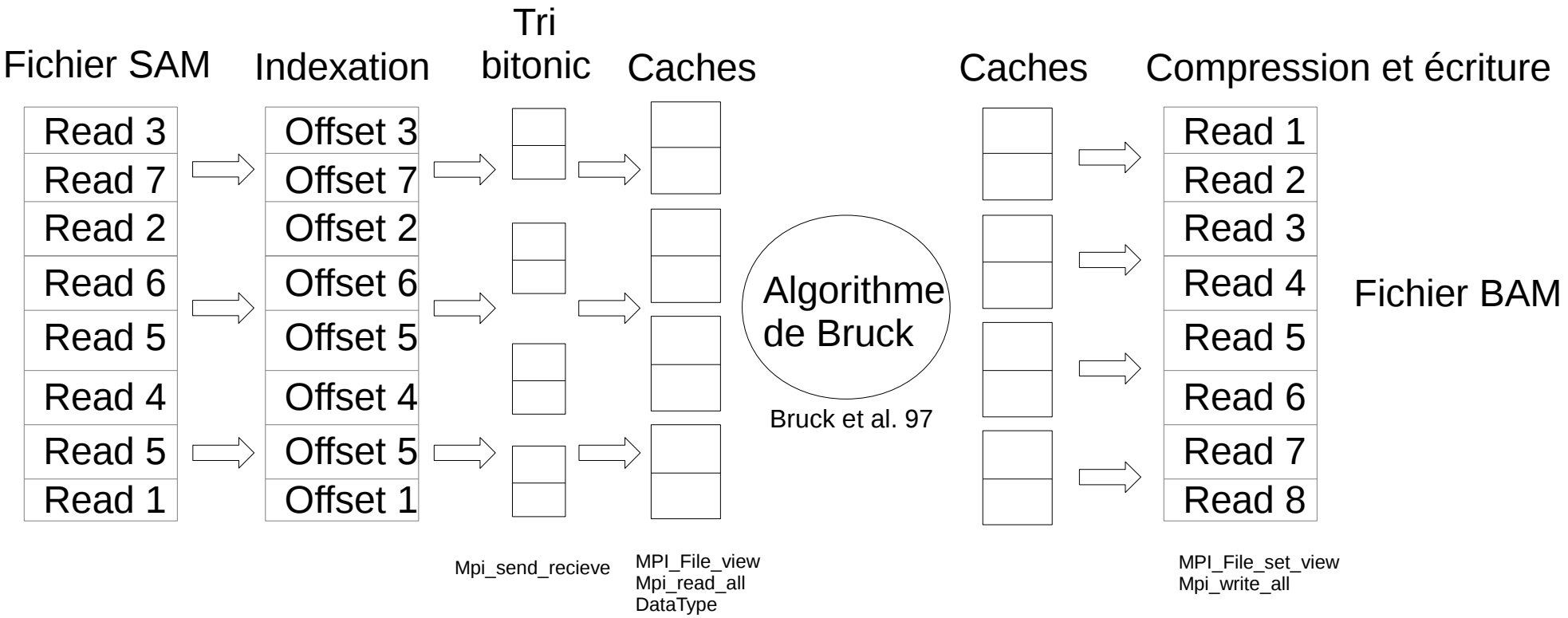


=> nouveau bottle neck = les communications, All2all de complexité n^2

=> 5h pour trier 1 To avec 500 cpu, 3To de RAM

- # Le tri par QUASART

Pour l'echange entre les caches on utilise l'agorithme de Bruck



=> 5h reduit à 30mn pour trier 1 To avec 320 cpu, Total RAM <1To

- **Le tri par QUASART**

Coverage	Sambamba (s)	mpiSort (s)	CPUs	speed-up	efficiency (%)
10X	1875	484	36	3.87	70.96
20X	4058	602	74	6.74	65.69
30X	6030	687	128	8.77	54.42
40X	8166	854	145	9.56	52.61
50X	10222	1000	182	10.22	44.36
60X	12590	1134	218	11.1	38.63
70X	14452	1246	256	11.59	30.69
80X	16625	1341	292	12.39	22.44
90X	24860	1574	380	15.79	14.94
120X	fail	1836	512	x	x

32 threads

⇨ Speed up linéaire, efficacité, scalabilité

Le HPC appliqué aux données de séquençage

- **Conclusion**
  - Code disponible sour licence GPLv3 : https://github.com/InstitutCurie/QUASART
  - En HPC les accélérations matérielles sont essentielles : Réseau faible latence, File system distribué, Caches,...
  - HPC = 50% accélérateurs matériels + 50% algorithmes
  - MPI propose un paradigme interessant qui permet de faire un traitement global de la données
  - Les marqueurs du HPC : le speed up, l'efficacité, la scalabilité.

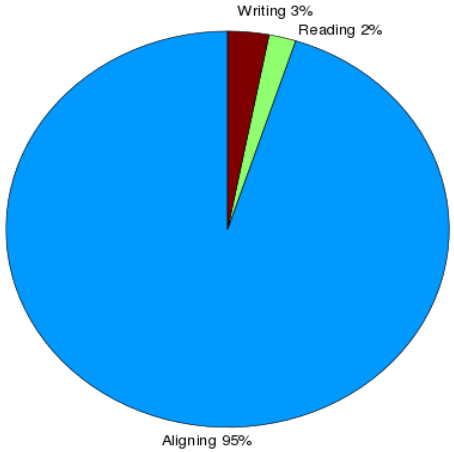
Le HPC appliqué aux données de séquençage

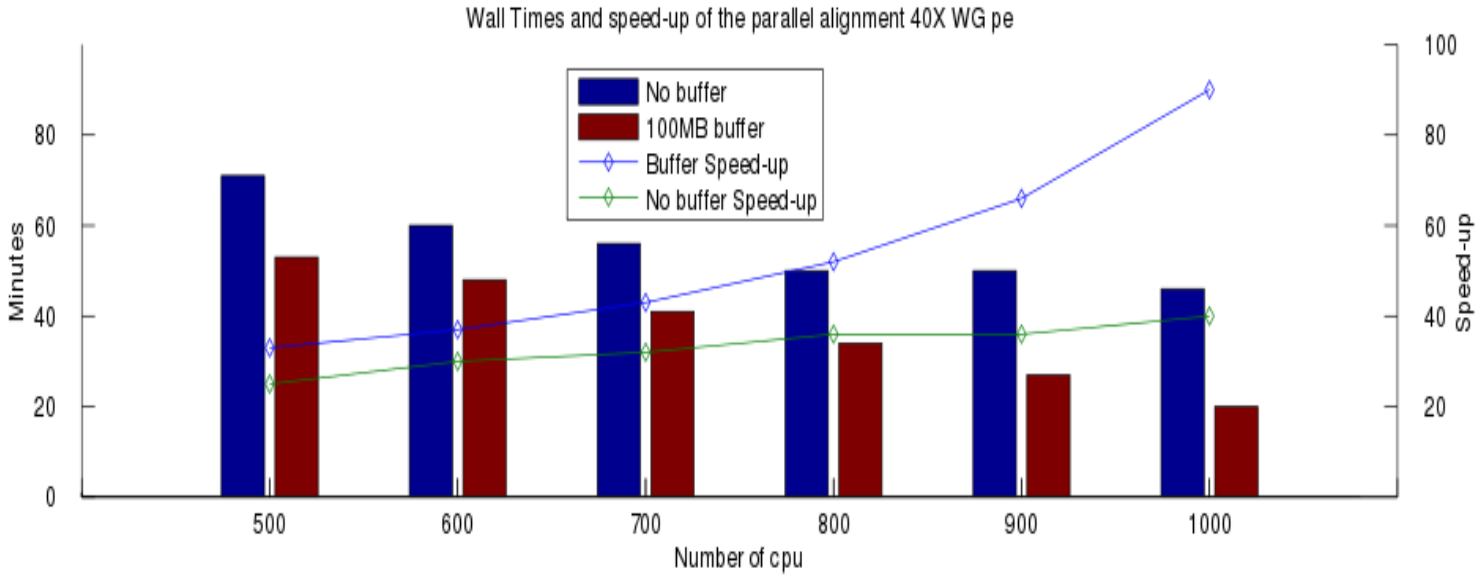
- **Remerciements**
  - De l'Institut Pasteur :
    - Nicolas Joly
  - De l'Institut Curie :
    - Philippe Hupe, François Prud'Homme, Maxime Chevailliot
  - Les étudiants de Paris Descartes :
    - Nicolas Fedy, Leonor Sirotti, Thomas Magalhaes, Paul Paganiban

Le HPC appliqué aux données de séquençage

- **Annexe : Gestion de la donnée avec MPI**
  - Utiliser les structures de données (DataType) pour éviter la recopie en mémoire
  - Masquer les communications par le calcul (operation non bloquantes)
  - Utilisation des vues (view) pour la lecture en paralléle
  - Plusieurs mode lecture parallele:
    - Data sieving
    - 2 phases collective read

Annexe : QUASART load balancing

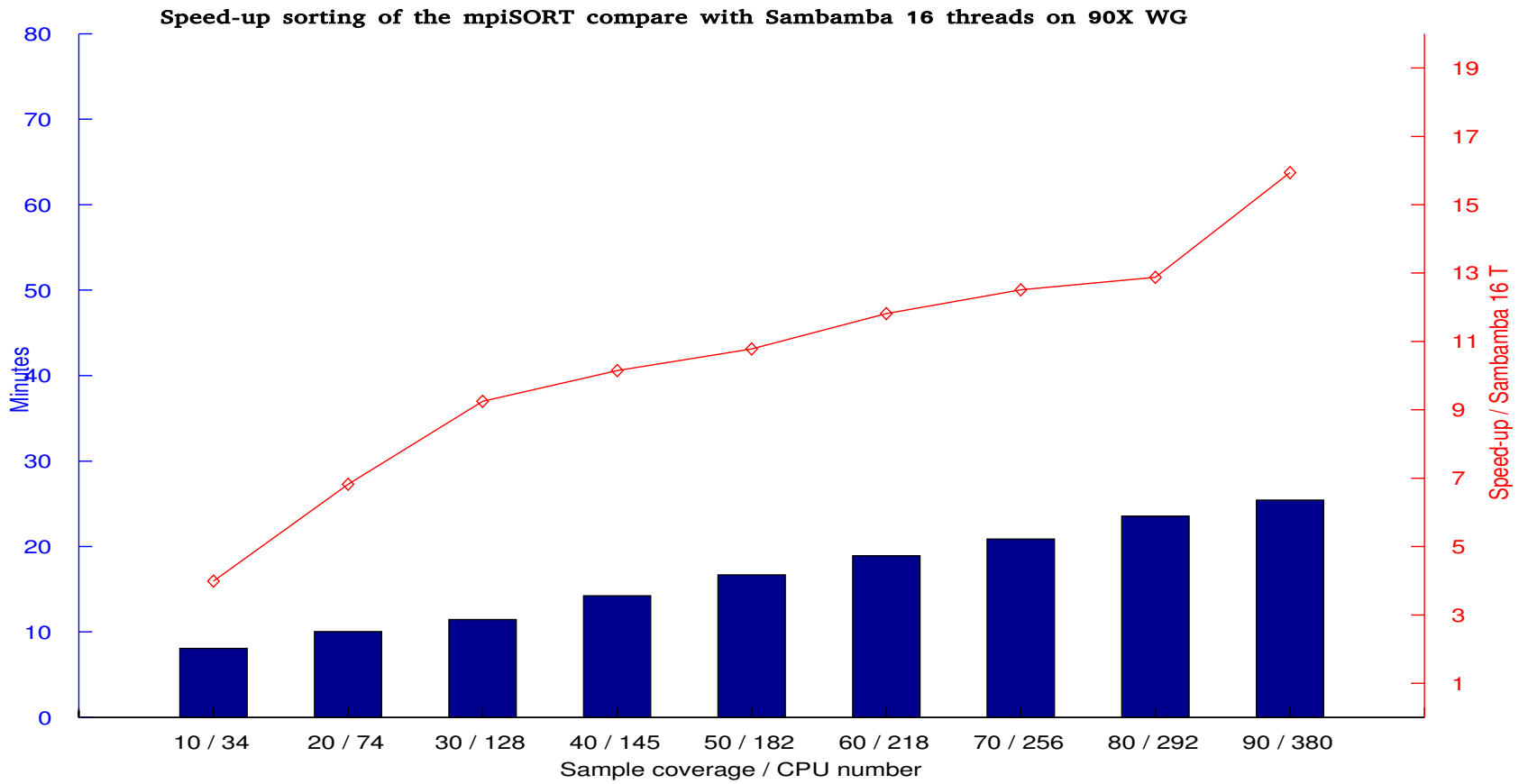




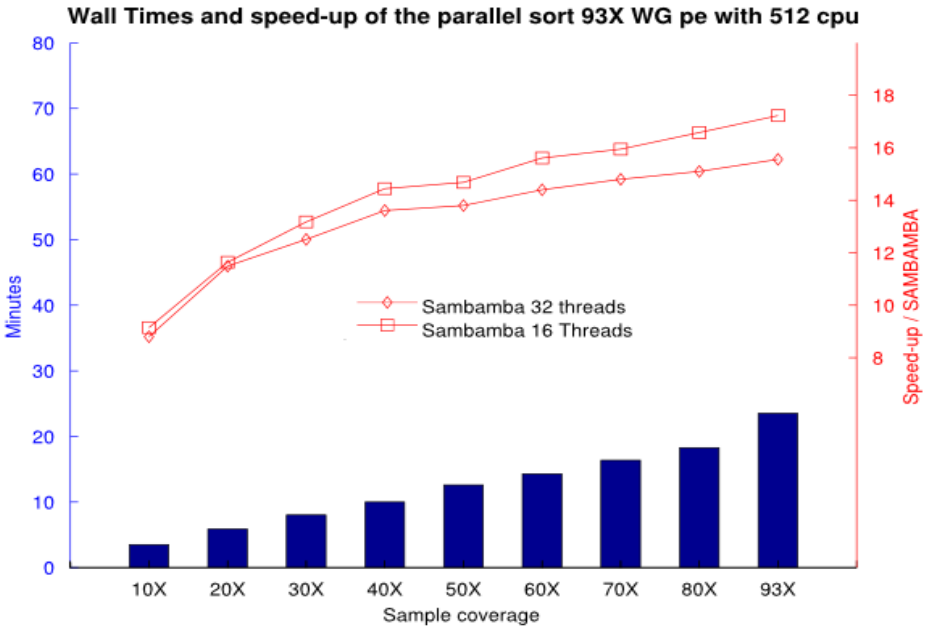
Algorithme efficace : 90% du temps est passé dans l'alignement

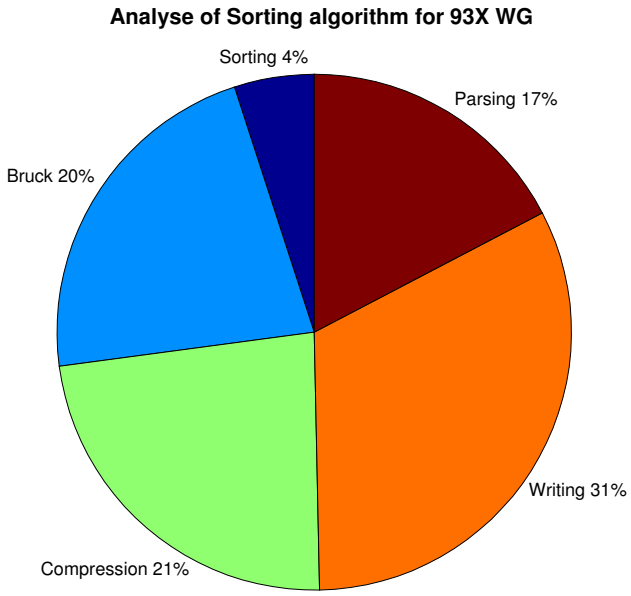
Taille des buffers pour load balancing

- # Annexe : le tri speed up



Annexe : Analyse du tri par QUASART





➡ Pas d'influence des threads

➡ Pipeline équilibré

Annexe : Discussions

- MPI (Message Passing Interface)
  - Avantages
    - Basé sur les messages entre jobs
    - Depuis MPI2 supporte les IO distribuées
    - Meilleure gestion des ressources (mémoire, cpu)
    - 2 implémentations OpenMPI, MVAPICH2
  - Inconvénients
    - Implémentation en langage bas niveau C, C++
    - Matériels adaptés

MVAPICH

THE OHIO STATE UNIVERSITY

Annexe : Discussion

- Méthodes basées sur le l'utilisation d'un cloud
  - Techniques “In memory” : SPARK
  - Hadoop
    - facile à implémenter (MapReduce)
    - scalabilité
    - Tout ne peut pas être “MapReducer”
    - File system spécifique HDFS
    - Protocole sécurisé

- HPC
  - XeonPhi, GPU
    - Bande passante limitée