Intérêt des codes FEC pour le stockage distribué
Le projet ANR FEC4Cloud et la solution RozoFS

RESSI-2015

Du 19 au 22 Mai, Université Technologique de Troyes

Benoît Parrein (Polytech Nantes, IRCCyN)
Jérôme Lacan (ISAE-SupAéro)
Nicolas Normand (Polytech Nantes, IRCCyN)
Dimitri Pertin (Polytech Nantes, IRCCyN et RozoSystems)
Jonathan Detchard (ISAE-SupAéro)
Alexandre van Kempen (Polytech Nantes, IRCCyN)

FEC4Cloud Project

- ANR 2012 (appel Emergence)
- Partenaires: IRCCyN (resp.), ISAE-SupAéro, SATT-Ouest Valorisation, Rozo Systems (prestataire)
- Budget: 256 K€
- Durée: 24 mois (orienté **produit**)
- Objectif: promouvoir les codes FEC au sein des infrastructures de stockage Cloud

Stockage distribué tolérant aux pannes: l'exemple de Facebook

- 260 Milliards de photos, 20 PB, 60 TB/semaine (2010) [1]
- Facebook ***f4 cells*** (pour données chaudes): 14 racks de 15 serveurs, 6.3K de disques (de 4TB chacun) soit **25 PB** par *cell* (1000PB par datacentre)
- Réduction du facteur de réplication de 3,6 à 2,1 par l'usage des codes FEC (soit une **économie de stockage de 87 PB**) [2]

[1] Beaver, D., Kumar, S., Li, H. C., Sobel, J., & Vajgel, P. (2010, October). Finding a Needle in Haystack: Facebook's Photo Storage. In OSDI (Vol. 10, pp. 1-8).

[2] Muralidhar, S., Lloyd, W., Roy, S., Hill, C., Lin, E., Liu, W., ... & Kumar, S. (2014, October). f4: Facebook's Warm BLOB Storage System. In Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation (OSDI) (pp. 383-398). USENIX Association.

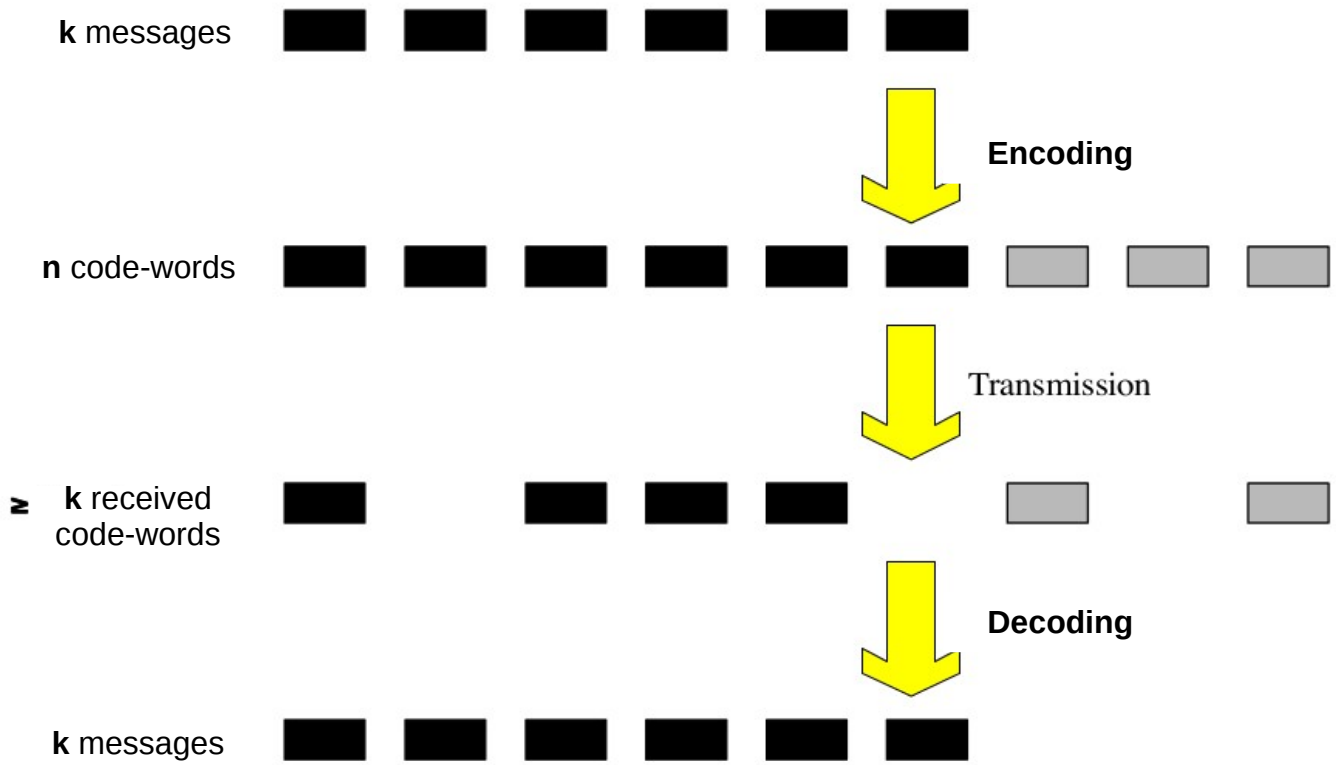
Objectifs

- Construire des codes efficaces (en matière de débit) pour tous types de données et d'accès
- Intégration de ces codes dans des solutions logicielles de stockage (*Software Defined Storage*)
- Faciliter la confidentialité des données (*privacy*)

Sommaire

- Le projet FEC4Cloud (en chiffres)
- Les codes FEC (MDS)
- Le code Mojette
- Performances (*micro-benchmark*)
- La solution RozoFS
- Expérimentations *in vivo* Grid5K
- Conclusions

Code FEC (MDS)


k messages
Encoding
n code-words
Transmission
≥ k received code-words
Decoding
k messages

Le code Reed-Solomon (RS)

- Usage d'une matrice génératrice *G* (code linéaire)
  - matrice de Cauchy ou Vandermonde

$$c = G.b$$

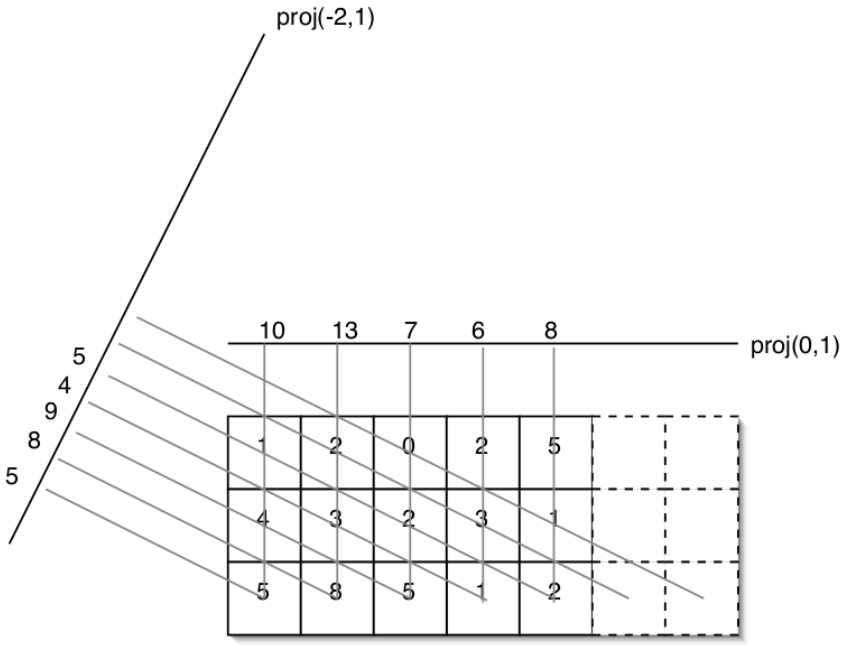
(si $G = \{I_k | P_{k,n-k}\}$, le code est **systématique** par définition)

Mises en œuvre des codes RS

- Reed-Solomon (matrice de Cauchy [Byers, 1995])
- Reed-Solomon (matrice de Vandermonde [Rizzo, 1998], RFC5510)
- ...
- Cauchy “Good” [Planck, 2008] in Jerasure 1.2
- Intel ISA-L (inclus instructions SSE)

Le code Mojette

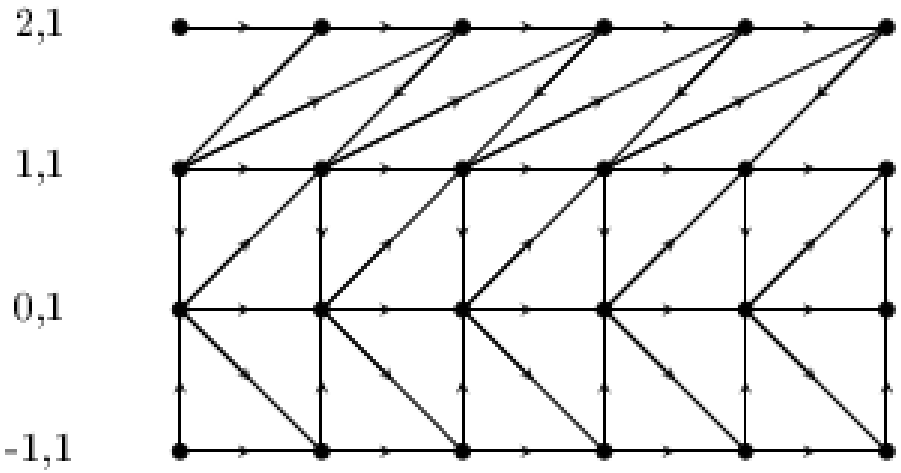
- based on Radon transform [Guédon, 1995]
- compute 1D projections from a 2D geometrical buffer



Décodage (1/2)

- Chemin de reconstruction déterministe [Normand, 2006]

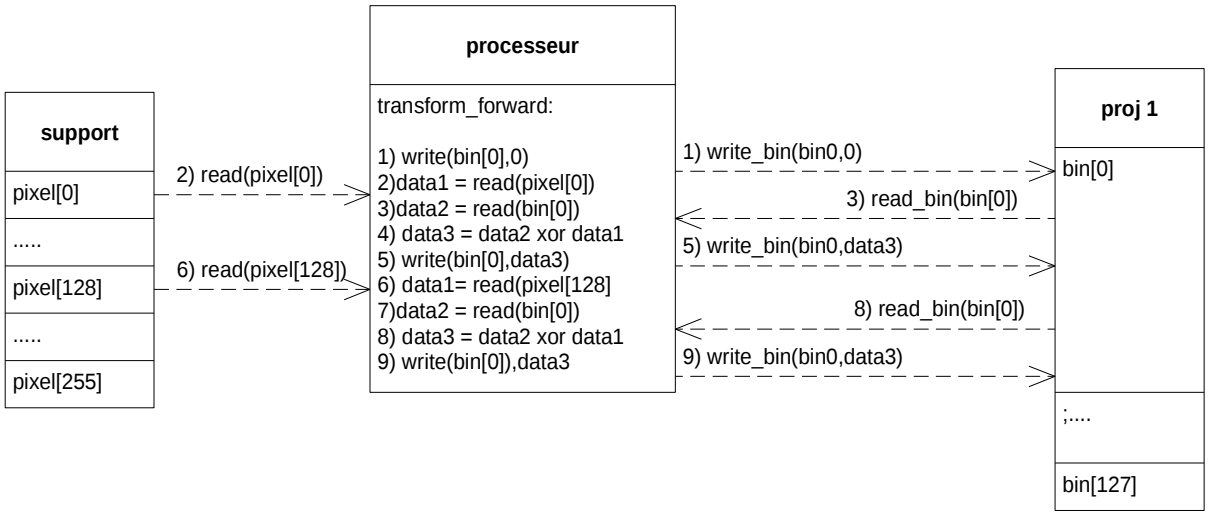
  (une projection reconstruit une ligne, de la droite vers la gauche)



Example on a 4 lines geometrical buffer
with 4 projections

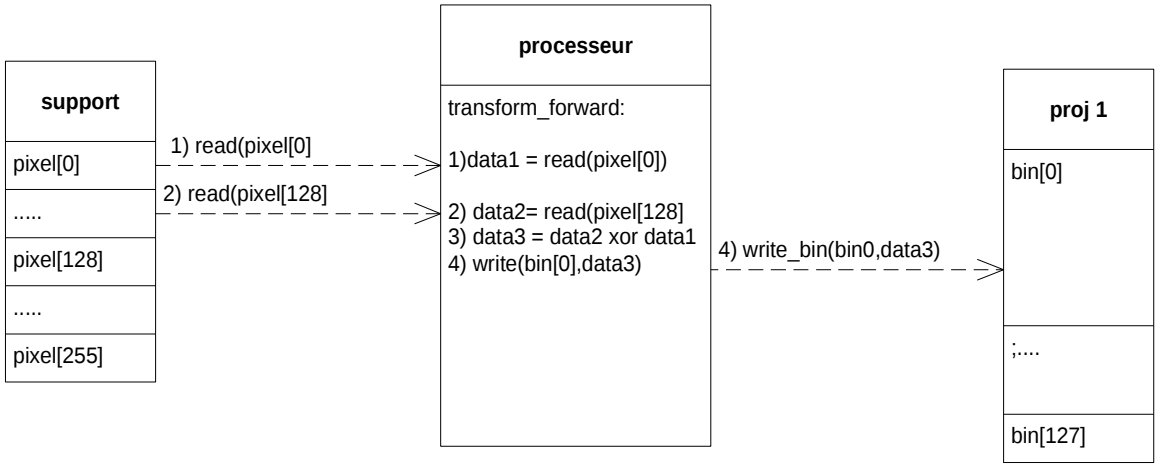
- +réduction du nombre d'écritures [Féron, 2014]

Optimisations (2/2)



Transformation Mojette Directe classique

Optimisations (2/2)

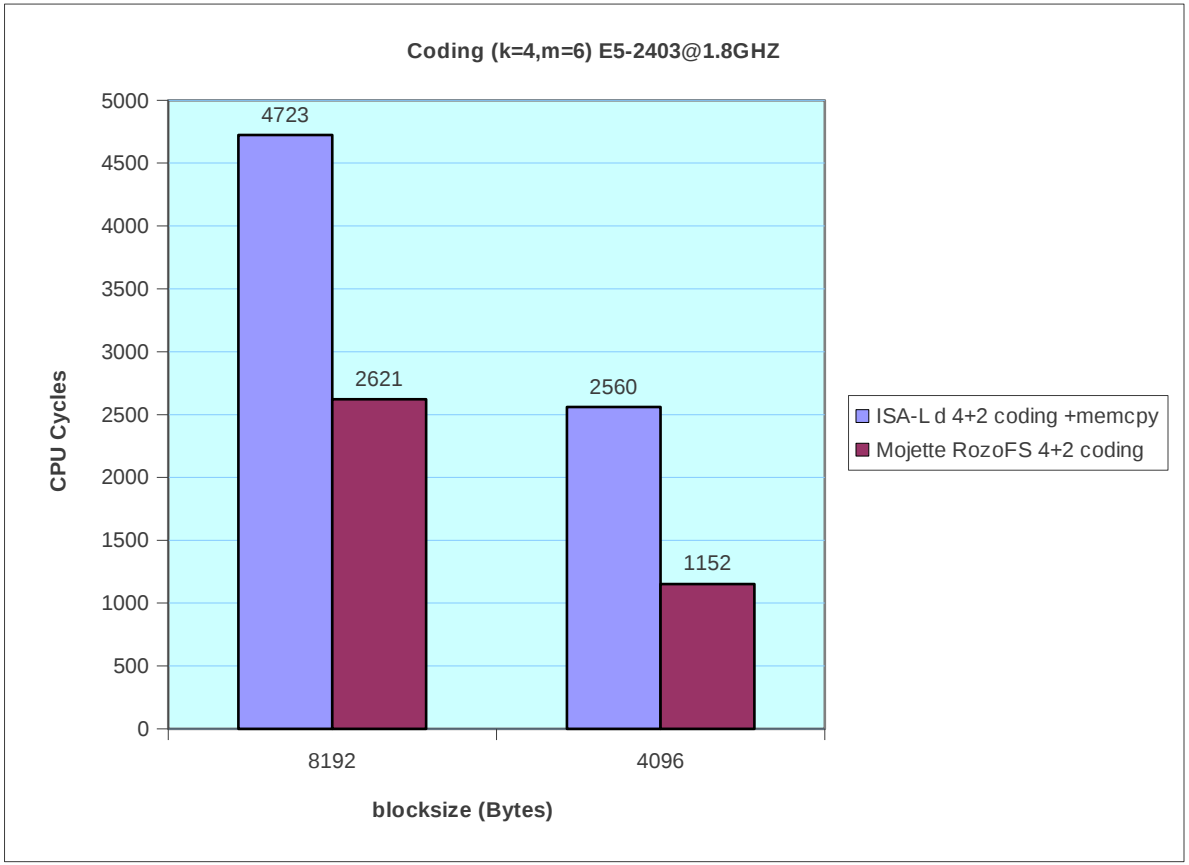


Transformation Mojette Directe optimisée [Féron, 2014]

Propriété du code Mojette

- Code à effacement de type (1+ϵ) MDS
- Forme non systématique (par défaut) et systématique
- Reconstruction asynchrone
- Pas de contrainte algébrique (cf corps de Galois)
- Pas de contrainte de primalité (cf FRT)
- Complexité linéaire [O(IN)] (naturelle)
- Codage et décodage logicielle

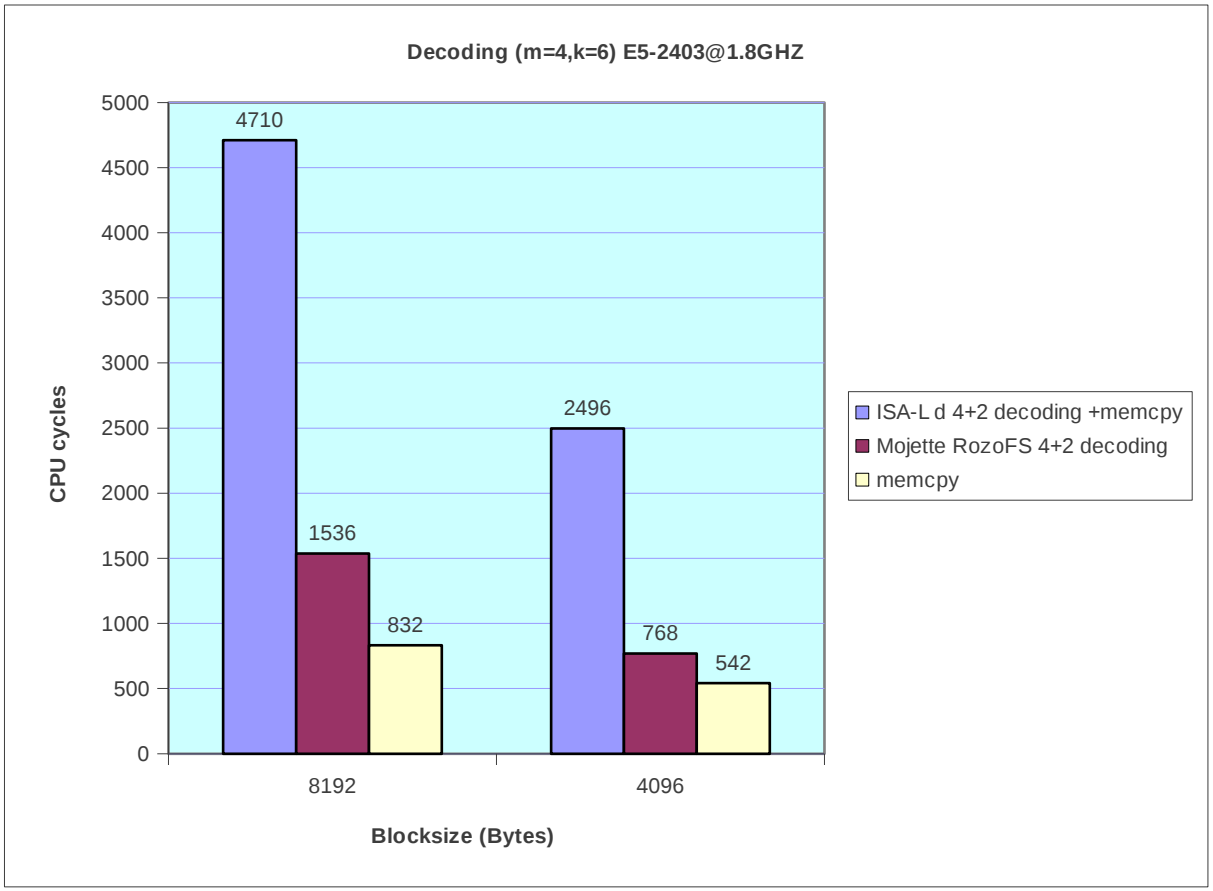
Performances (codage) en 8K et 4K



correspond à 5.625 GB/s (resp. 6.40 GB/s en 4KB) pour le code Mojette
et 3.122 GB/s (resp. 2.88 GB/s en 4KB) pour le code RS (ISA-L)

i.e x1.8 (resp. x2.22) plus rapide avec x3 plus de mot-codes (cf non systematique)

Performances (décodage)



correspond à 9.6 GB/s (resp. 9.60 GB/s avec blocs de 4KB) pour le décodage Mojette (en violet) et 3.130 GB/s (resp. 2.953 GB/s avec 4KB) pour le décodage RS (en bleue) en cas de pannes

x3 (resp. x3.25) plus rapide (pour x2 plus de mot-codes)

Intérêt des codes FEC pour le stockage distribué
Le projet ANR FEC4Cloud et **la solution RozoFS**

RESSI-2015

Du 19 au 22 Mai, Université Technologique de Troyes

Benoît Parrein (Polytech Nantes, IRCCyN)
Jérôme Lacan (ISAE-SupAéro)
Nicolas Normand (Polytech Nantes, IRCCyN)
Dimitri Pertin (Polytech Nantes, IRCCyN et RozoSystems)
Jonathan Detchard (ISAE-SupAéro)
Alexandre van Kempen (Polytech Nantes, IRCCyN)

Haute disponibilité signifie...

- 99,999999....% accessible
- de la réplication le plus souvent...
- une infrastructure capable de supporter la charge
- une haute consommation énergétique
- ...et des problèmes de *Privacy*
  - les codes à effacements réduisent fortement la taille de l'infrastructure pour le même taux de disponibilité

Distributed File Systems (DFS)

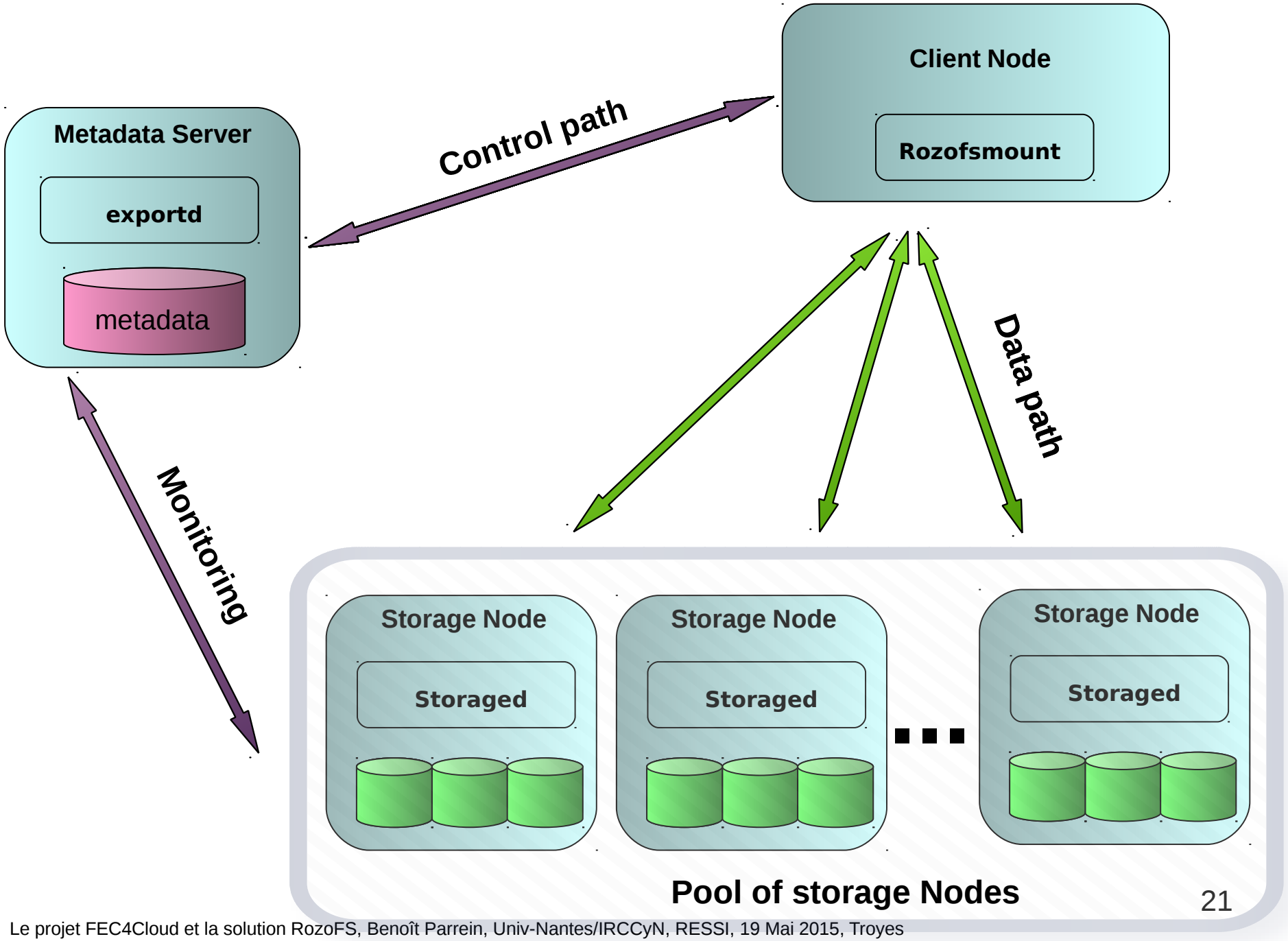
- HDFS (Hadoop)
- Facebook file system (f4)
- Scality (basé sur Chord)
- CephFS, GlusterFS, ...
- ...

  Mélange de réplicats (données chaudes) et de codes à effacement (données froides)
  - **aucun DFS n'utilise aujourd'hui des codes pour des données soumises à des I/O intensives**


ROZOBOX v1
We've got the technology!

RozoFS

- I/O Centric Distributed File System
  - Software Defined Storage (SDS)
  - Scale-Out NAS
  - POSIX (based on FUSE)
  - Commodity hardware
  - Fault tolerance (up to 4 failures)
  - Based on erasure coding (Mojette coding)
  - Dedicated to cold, warm and hot data
- Open source project

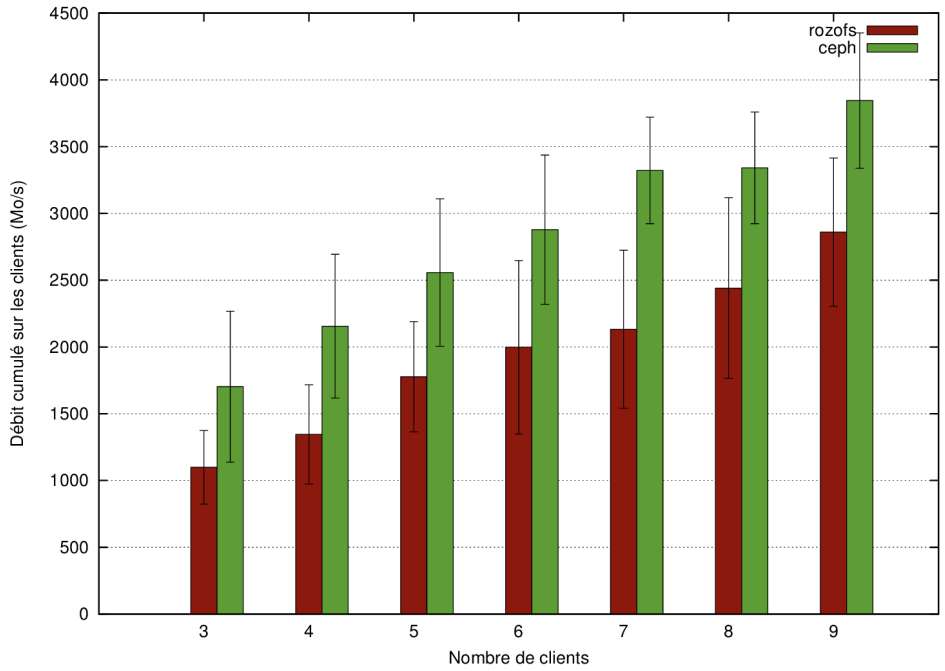

Client Node
Rozofsmount
Metadata Server
exportd
metadata
Control path
Data path
Monitoring
Storage Node
Storaged
Storage Node
Storaged
Storage Node
Storaged
Pool of storage Nodes

Expérimentations *in vivo* sur GRID5K

- sur cluster de Nantes@EMN
  - 18 nœuds, Intel Xeon 2.2Ghz, 64 GB RAM, 10GbE
- Layout 1 i.e Mojette(6,4) vs triplication pour Ceph
- 10 Go de données, découpage fichier en blocs de 8Ko
- Accès séquentiels et aléatoires en lecture et écriture (via *IOZone*)

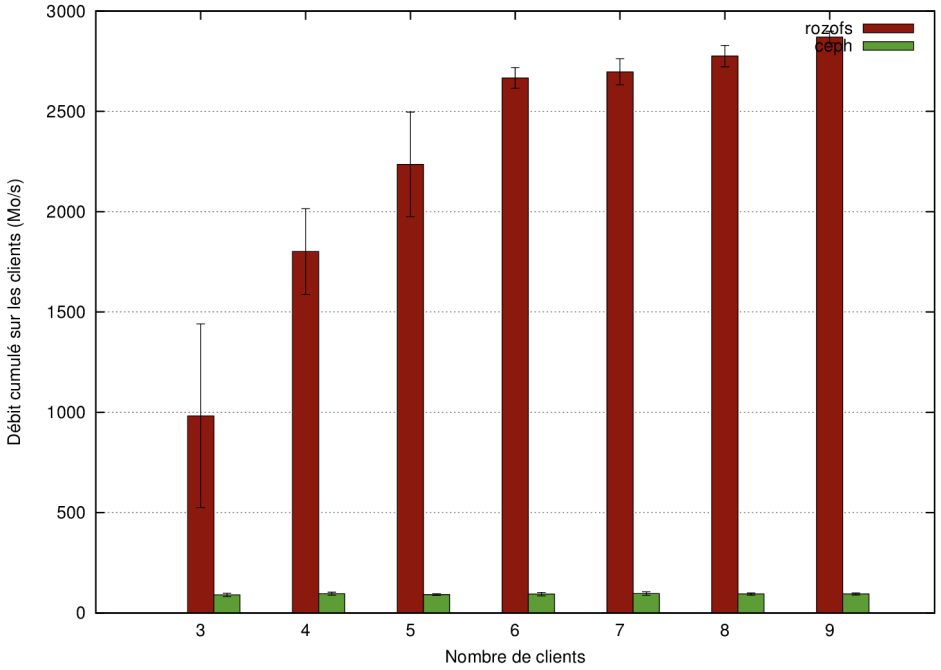
Expérimentations *in vivo* sur GRID5K

- Lecture séquentielle (en Mo/s)



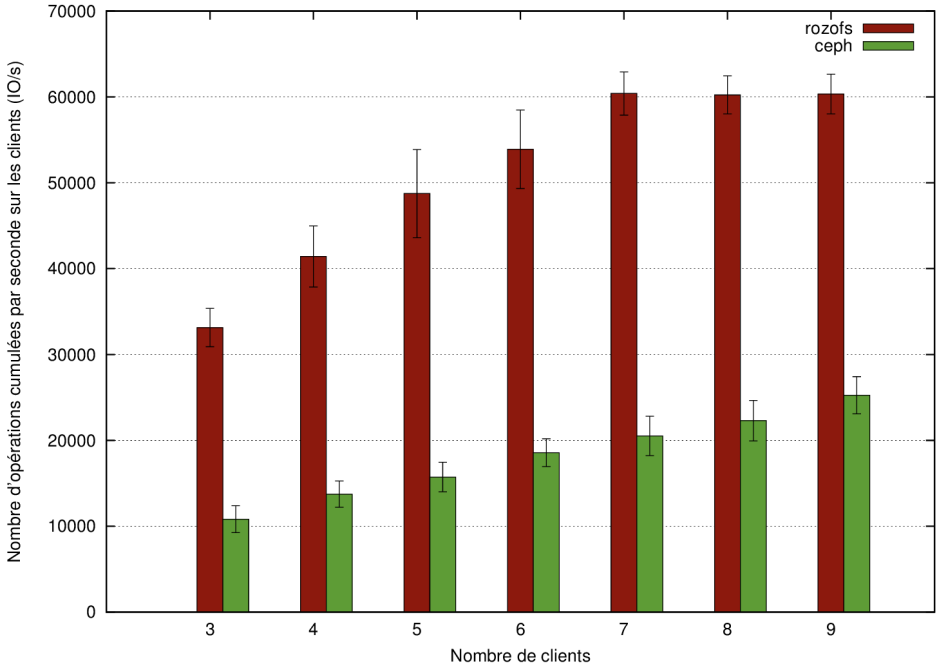
Expérimentations *in vivo* sur GRID5K

- Écriture séquentielle (en Mo/s)



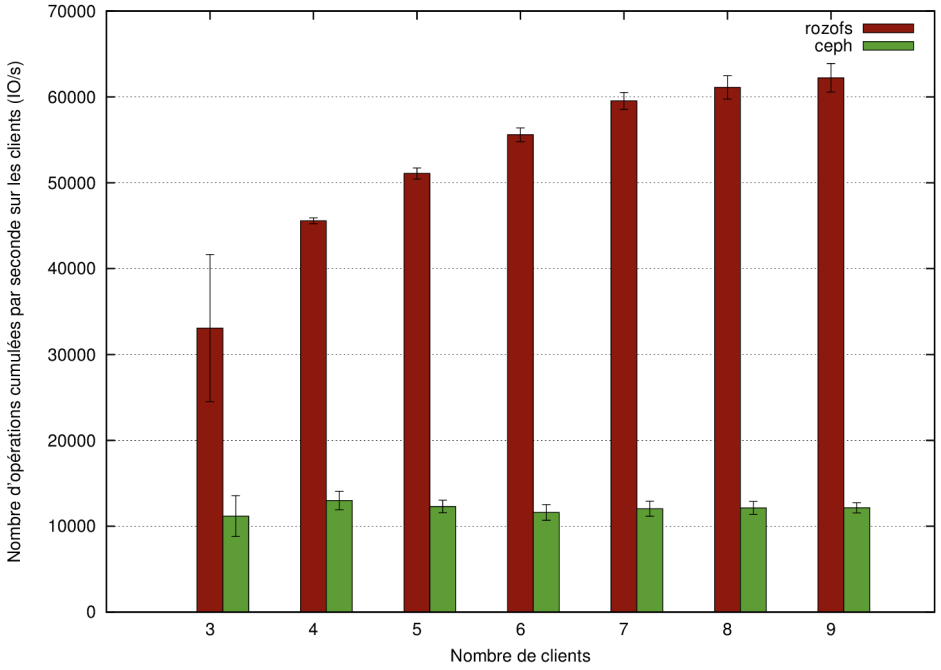
Expérimentations *in vivo* sur GRID5K

- Lecture aléatoire (en IOPS)



Expérimentations *in vivo* sur GRID5K

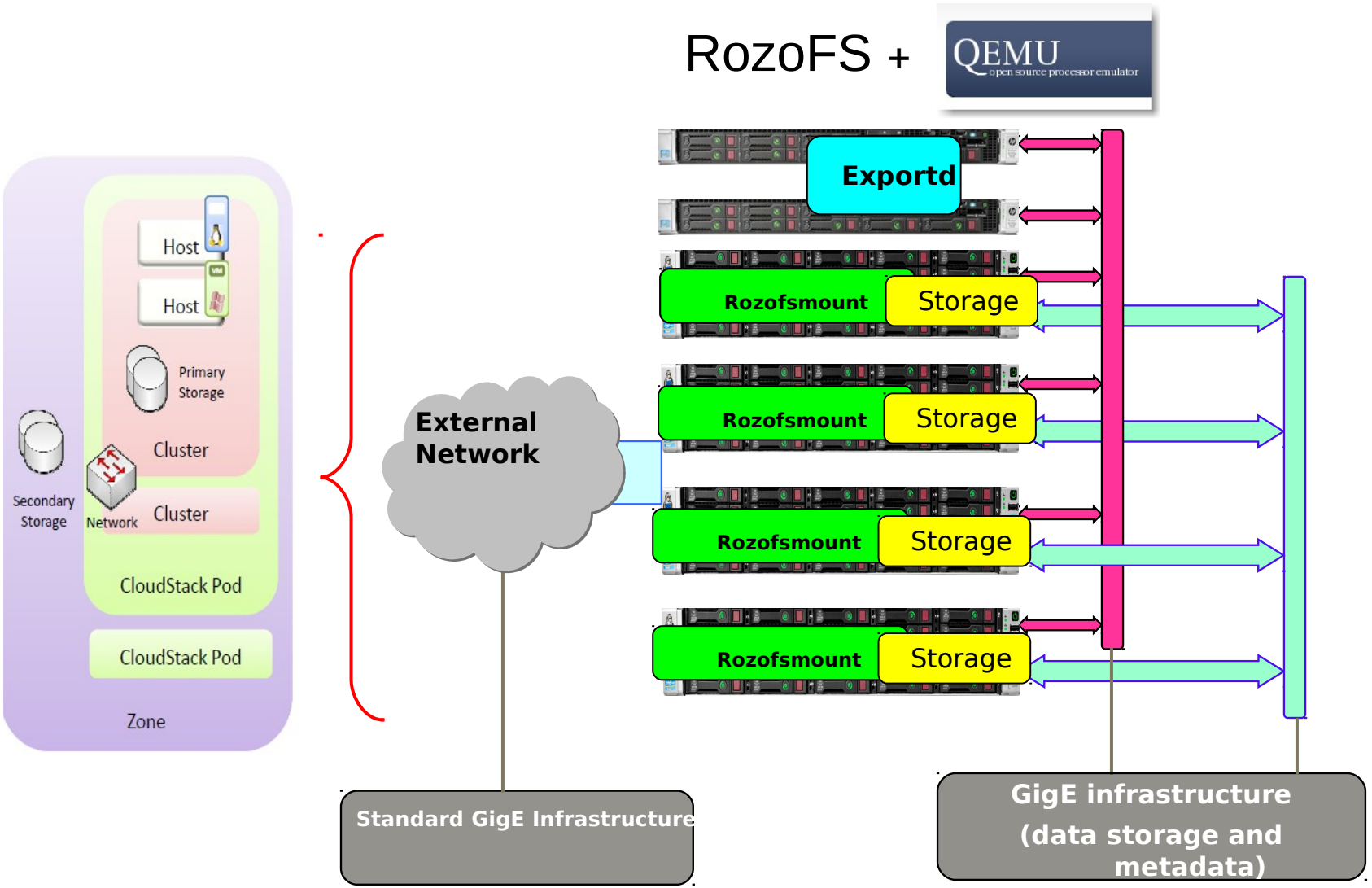
- Écriture aléatoire (en IOPS)



Expérimentations *in vivo* sur GRID5K

- Capacité de stockage pour 10 Go utile

	Serveurs de stockage (Go)								Total (Go)
	1	2	3	4	5	6	7	8	
rozoFS	2.4	2.4	2.1	2.1	2.1	2.1	2.1	2.1	17.4
rozoFS v2	2.0	2.0	2.0	2.0	2.0	1.8	1.8	1.5	15.3
ceph	6.0	5.3	5.0	5.0	4.7	4.3	2.2	2.0	34.5


RozoFS +
QEMU
open source processor emulator
Exportd
Rozofsmount
Storage
Rozofsmount
Storage
Rozofsmount
Storage
Rozofsmount
Storage
External Network
Host
Host
Primary Storage
Cluster
Secondary Storage
Network
Cluster
CloudStack Pod
CloudStack Pod
Zone
Standard GigE Infrastructure
GigE infrastructure (data storage and metadata)

Conclusions

- Le code Mojette proche de l'instruction `memcpy`
- 2 à 3 fois plus rapide que la librairie ISA-L
- RozoFS: 1er DFS (I/O *centric*) utilisant un code FEC
- Capacité de stockage réduite de moitié
- Applications: au montage vidéo en ligne, exécution de machines virtuelles et bases de données transactionnelles, traitements distribués type MapReduce,...
- Rozo Systems emploie aujourd'hui 8 personnes (dont 5 en R&D) et s'ouvre à l'international (USA)

Merci de votre attention!