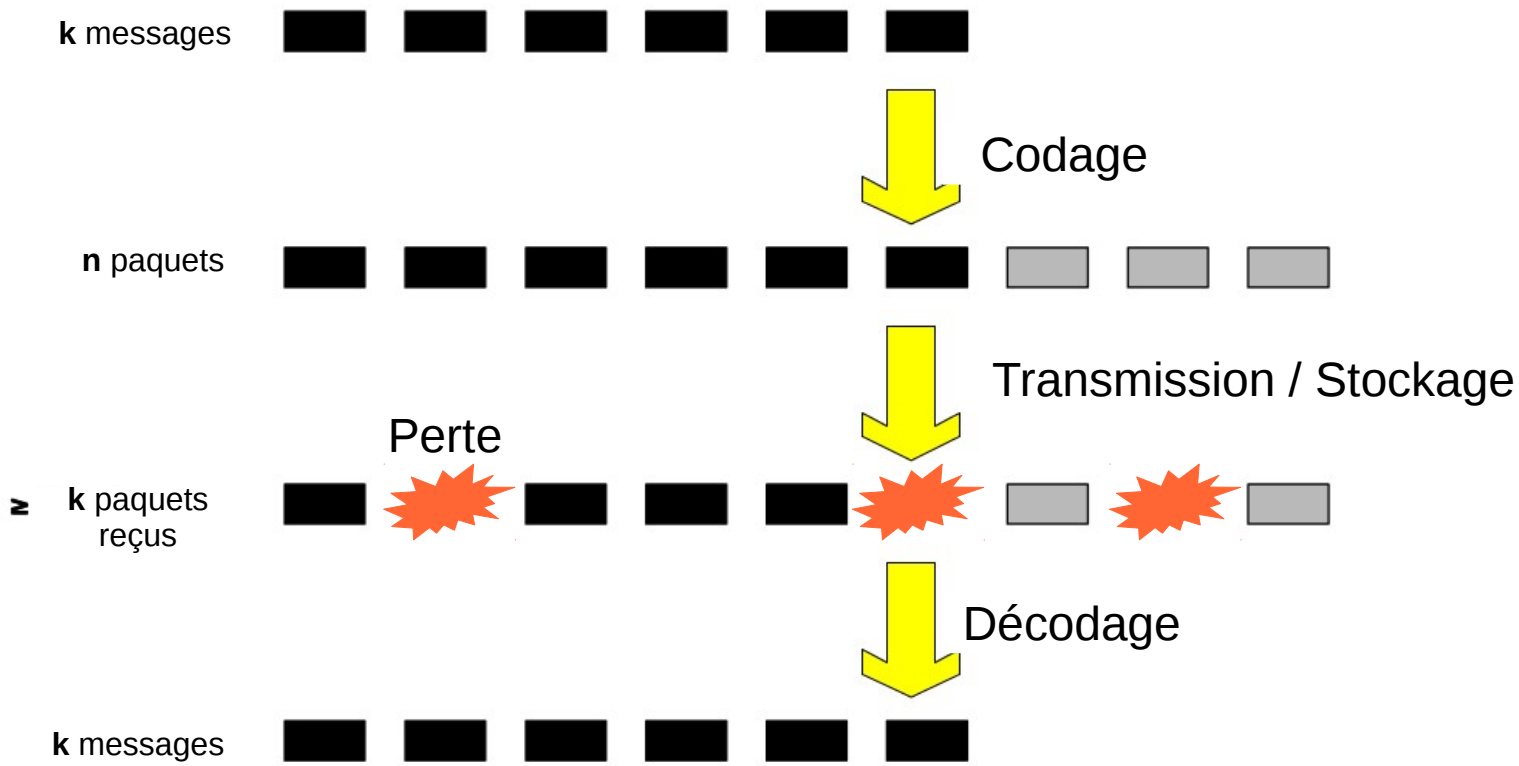
FEC4Cloud: pour la promotion des codes à effacement dans des architectures de stockage distribuées

Journée Virtualisation et Cloud, GdR RSD
14/09/2015, Jussieu, LiP6

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

Les codes à effacement



(propriété MDS)

Les codes à effacement dans f4 (Facebook)

- **87 PB** de moins à stocker [1]
- **730 serveurs** en moins
- **42%** de réduction de la taille de l'infrastructure
- **20 MW** de consommation en moins pour un datacentre [2]

[1] Muralidhar, S., Lloyd, W., Roy, S., Hill, C., Lin, E., Liu, W., ... & Kumar, S. 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, October, 2014.

[2] Orgerie, Anne-Cecile, Assuncao, Marcos Dias de, & Lefevre, Laurent . A survey on techniques for improving the energy efficiency of large-scale distributed systems. ACM Computing Surveys (CSUR), vol. 46, no 4, p. 47, 2014.

Limitation

- Application uniquement sur des données “froides”

  i.e **80 lectures par seconde** max

Projet FEC4Cloud

- ANR 2012 (appel Emergence)
- Partenaires: IRCCyN (resp.), ISAE-SupAéro, SATT-Ouest Valorisation, Rozo Systems (prestataire)
- Budget: 256 K€
- Durée: 24+6 mois (orienté **produit**)

Objectifs du projet FEC4Cloud

- Proposer des algorithmes de codages performants permettant de satisfaire un grand nombre d'entrées/sorties par seconde (pour **des données chaudes**)

- Intégration dans RozoFS, solution logicielle de stockage distribué (*Software Defined Storage*)

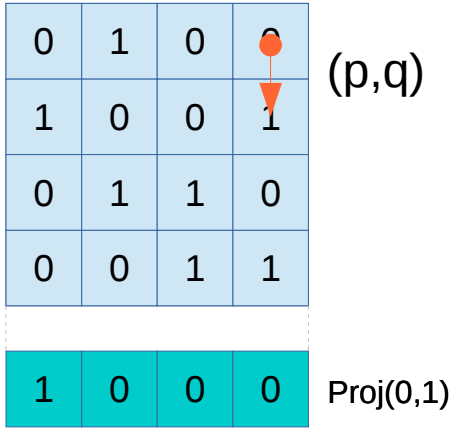
Sommaire

- Le code Mojette
- Performances (*micro-benchmark*)
- RozoFS
- Expérimentations *in vivo* Grid5000
- Conclusions

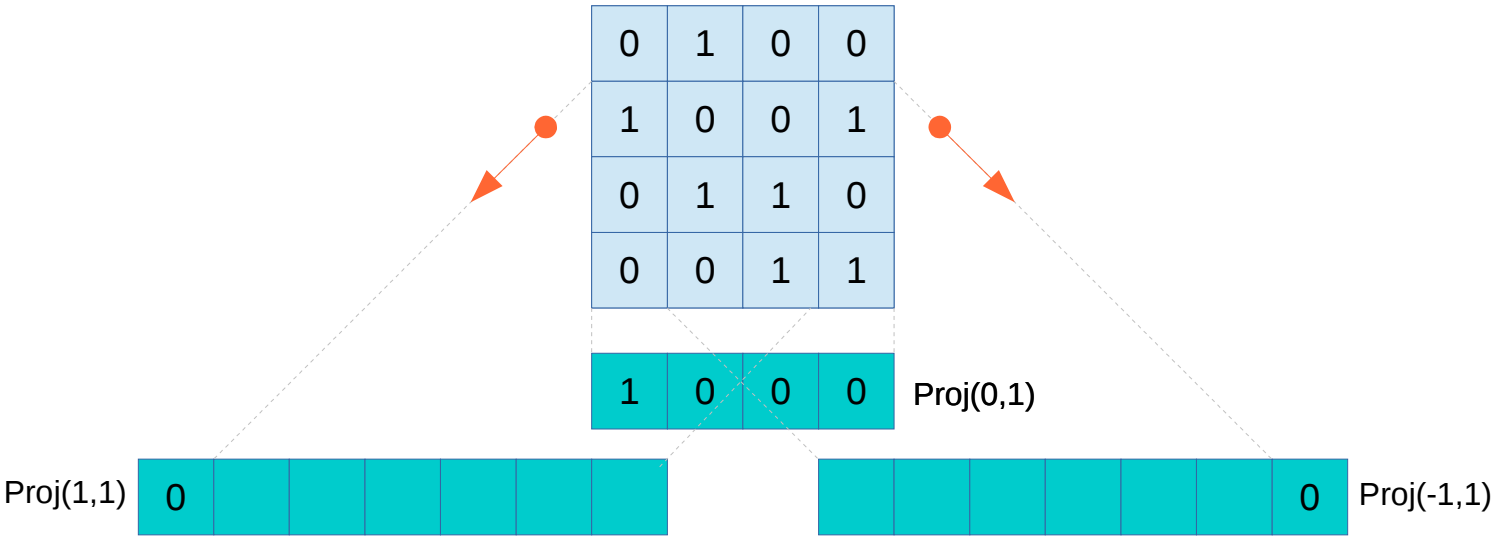
Le code à effacement Mojette

0	1	0	0
1	0	0	1
0	1	1	0
0	0	1	1

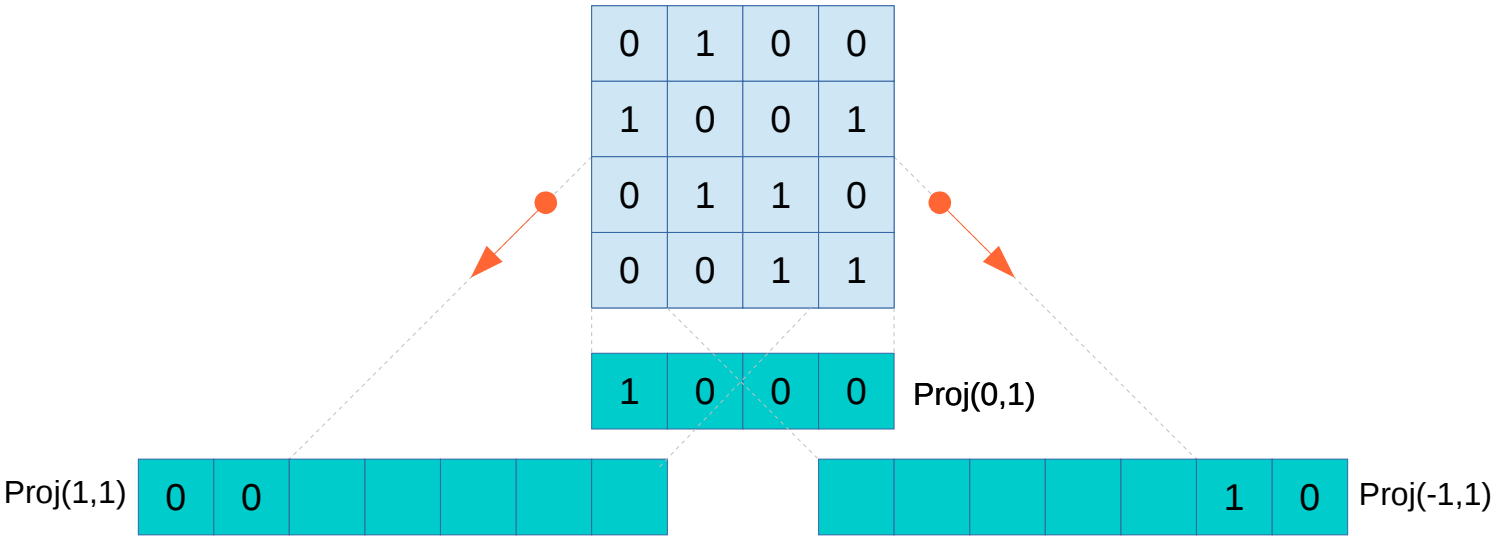
Le code à effacement Mojette



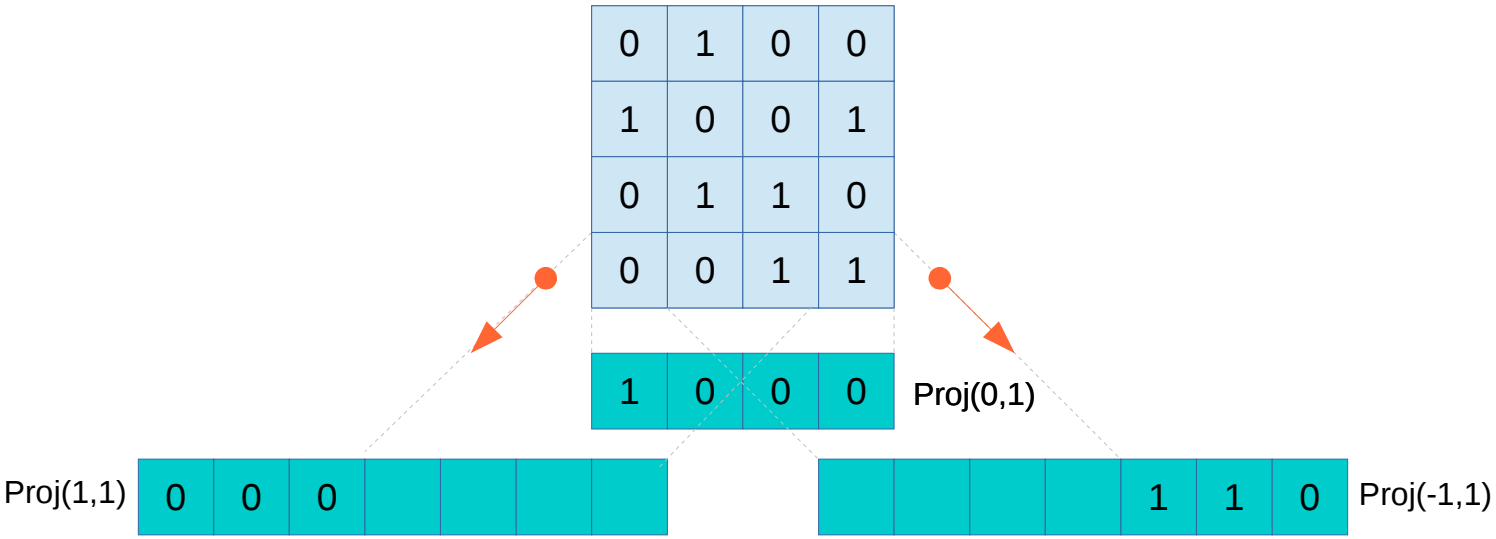
Le code à effacement Mojette



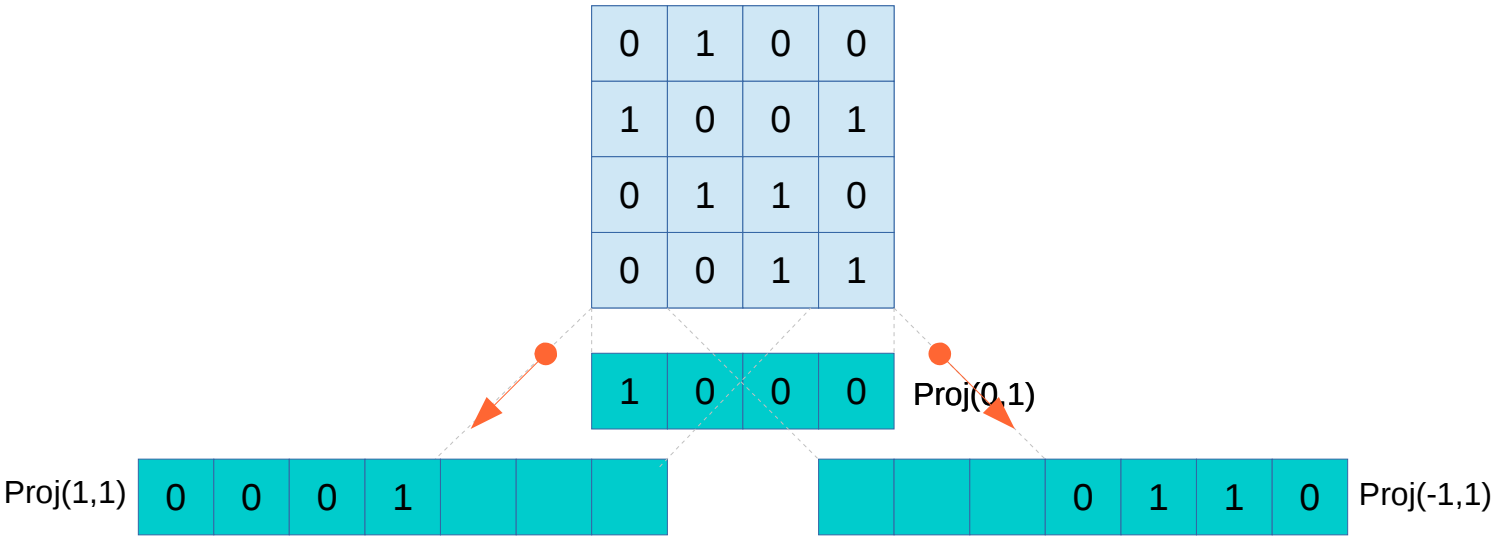
Le code à effacement Mojette



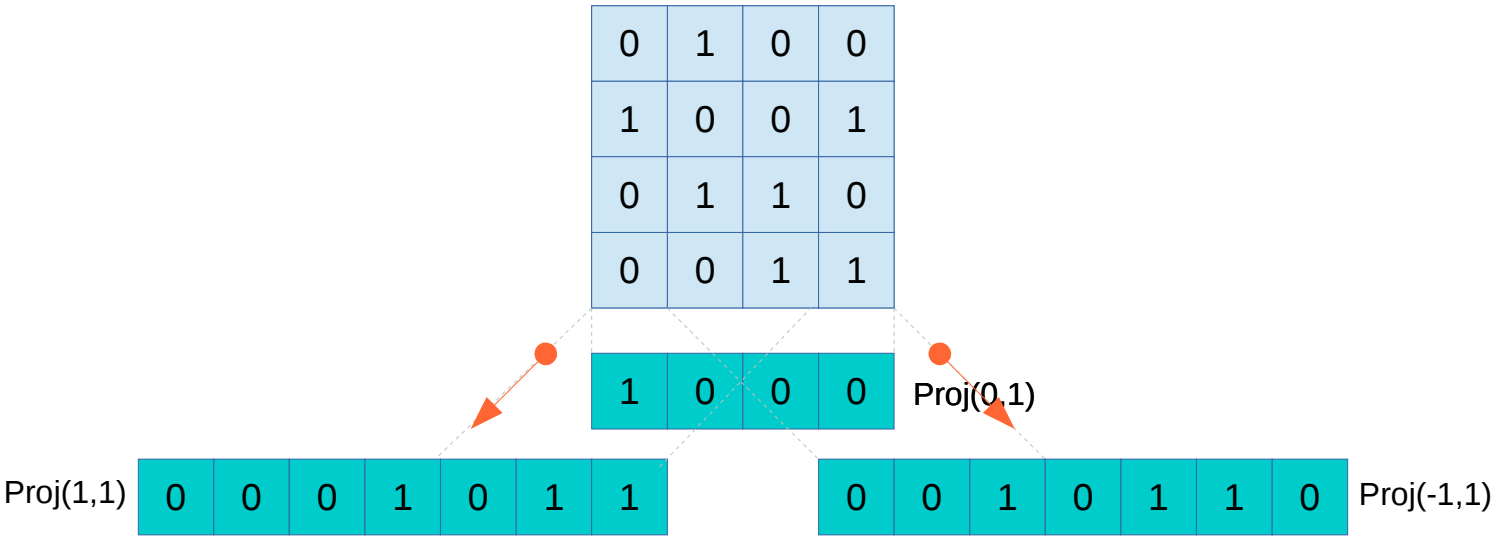
Le code à effacement Mojette



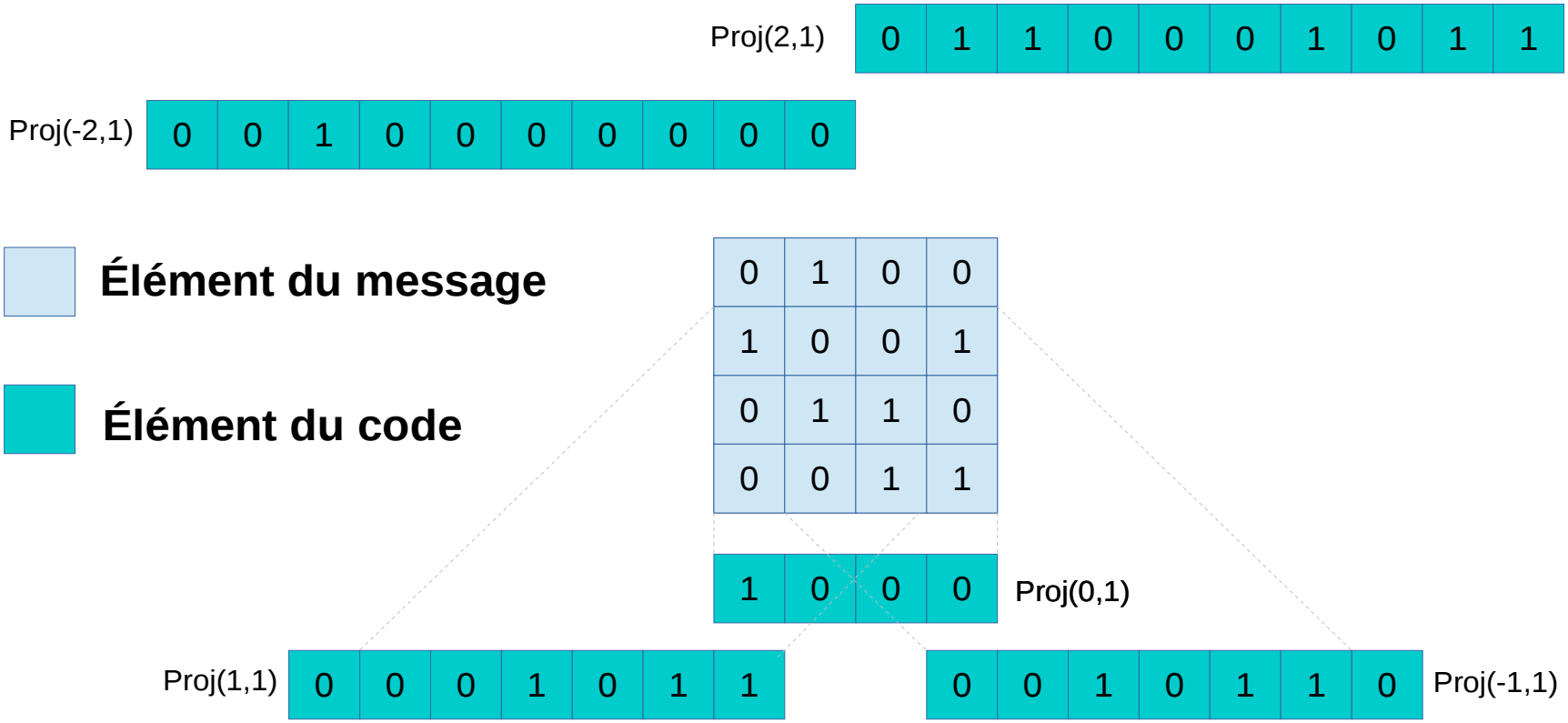
Le code à effacement Mojette



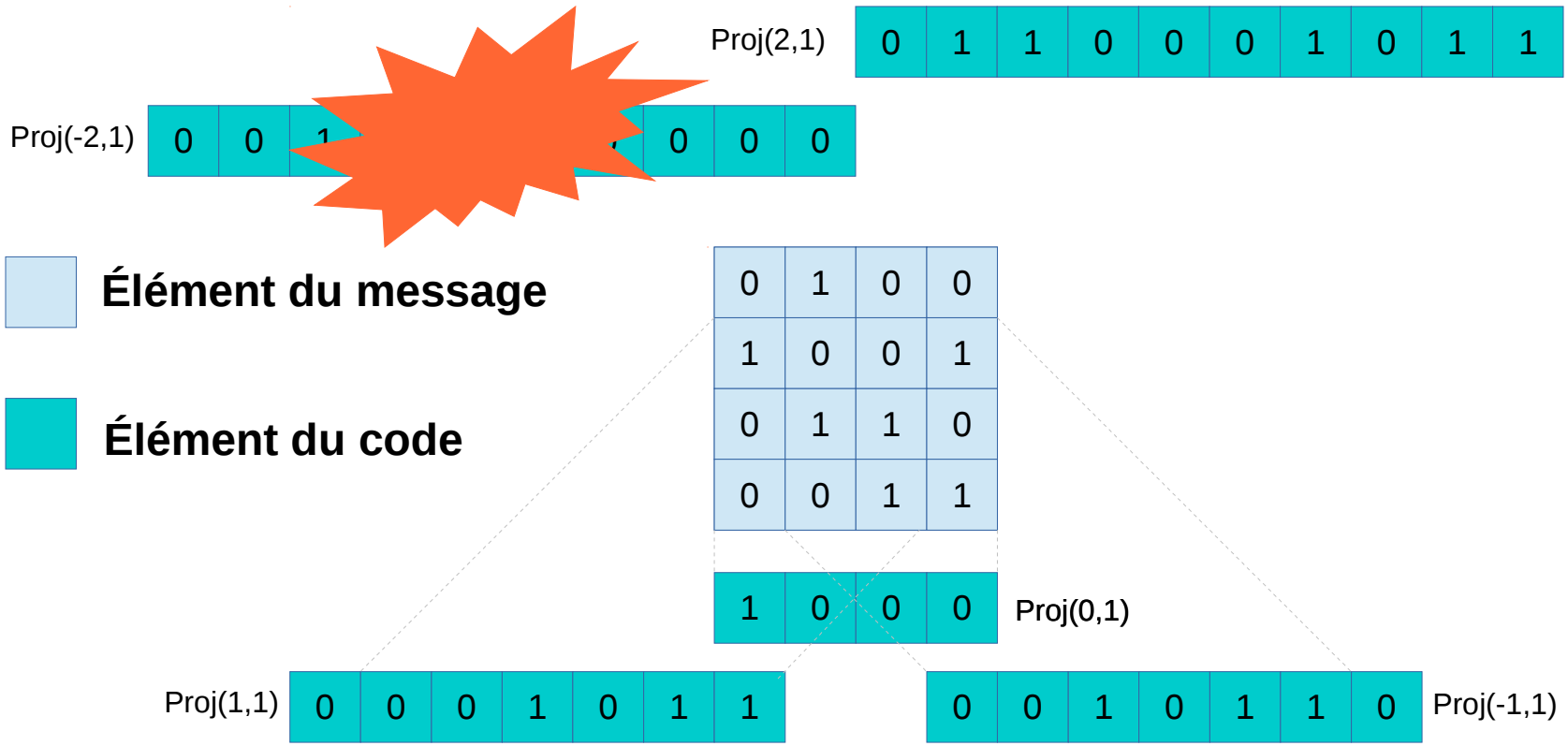
Le code à effacement Mojette



Le code à effacement Mojette



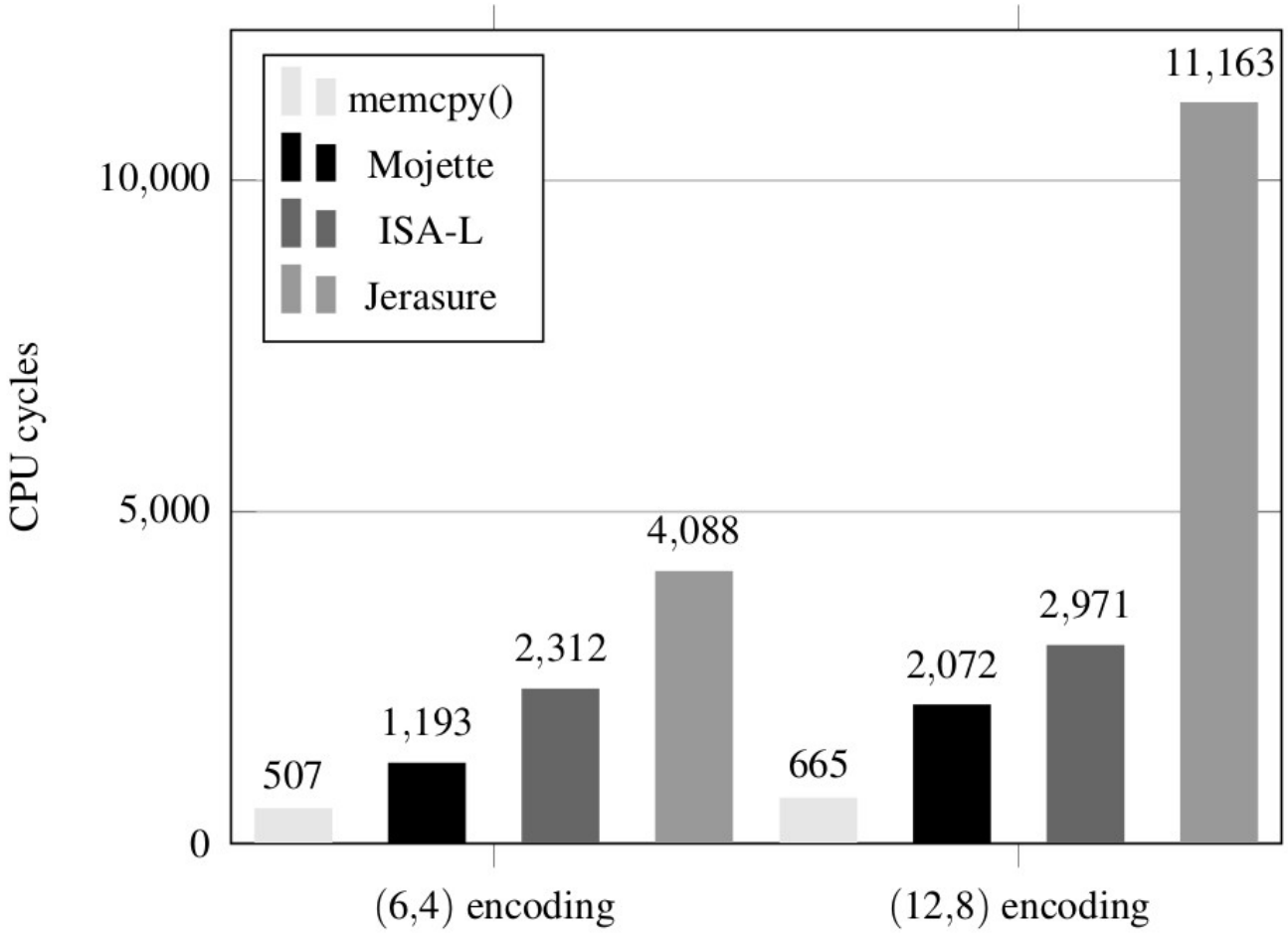
Le code à effacement Mojette



Propriété du code Mojette

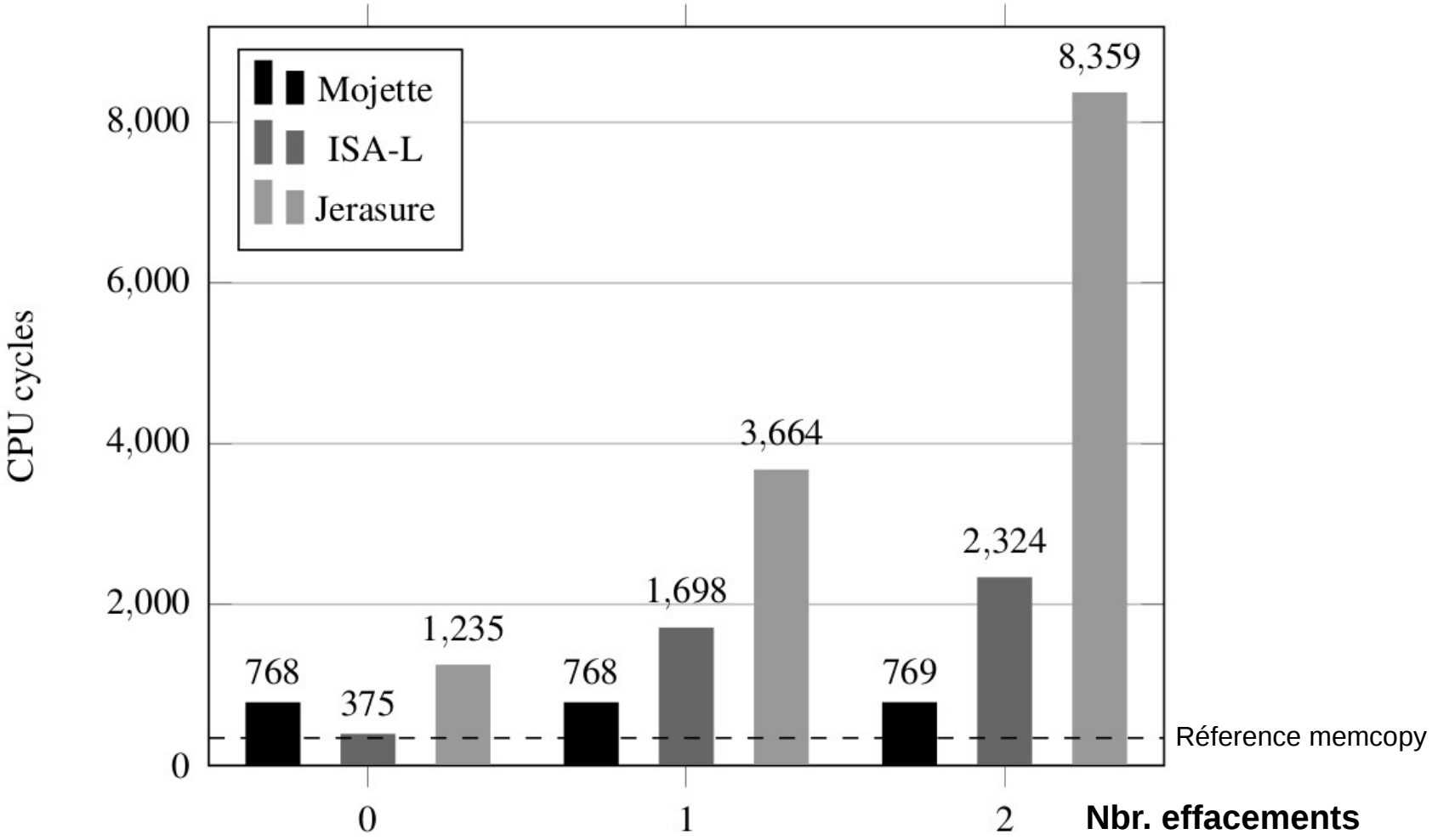
- Code à effacement de type (1+ϵ) MDS
- Forme non systématique (par défaut) et systématique
- Reconstruction asynchrone
- Complexité linéaire [O(IN)]
- Codage et décodage logicielle

Performances (codage) sur bloc 4K



Source : Pertin, D., Féron, D., Van Kempen, A., & Parrein, B. (2015). Performance evaluation of the Mojette erasure code for fault-tolerant distributed hot data storage. arXiv preprint arXiv:1504.07038 [cs.IT].

Performances (décodage) sur bloc 4K



Source : Pertin, D., Féron, D., Van Kempen, A., & Parrein, B. (2015). Performance evaluation of the Mojette erasure code for fault-tolerant distributed hot data storage. arXiv preprint arXiv:1504.07038 [cs.IT].

Application au stockage distribué

Stockage avec haute disponibilité

- 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

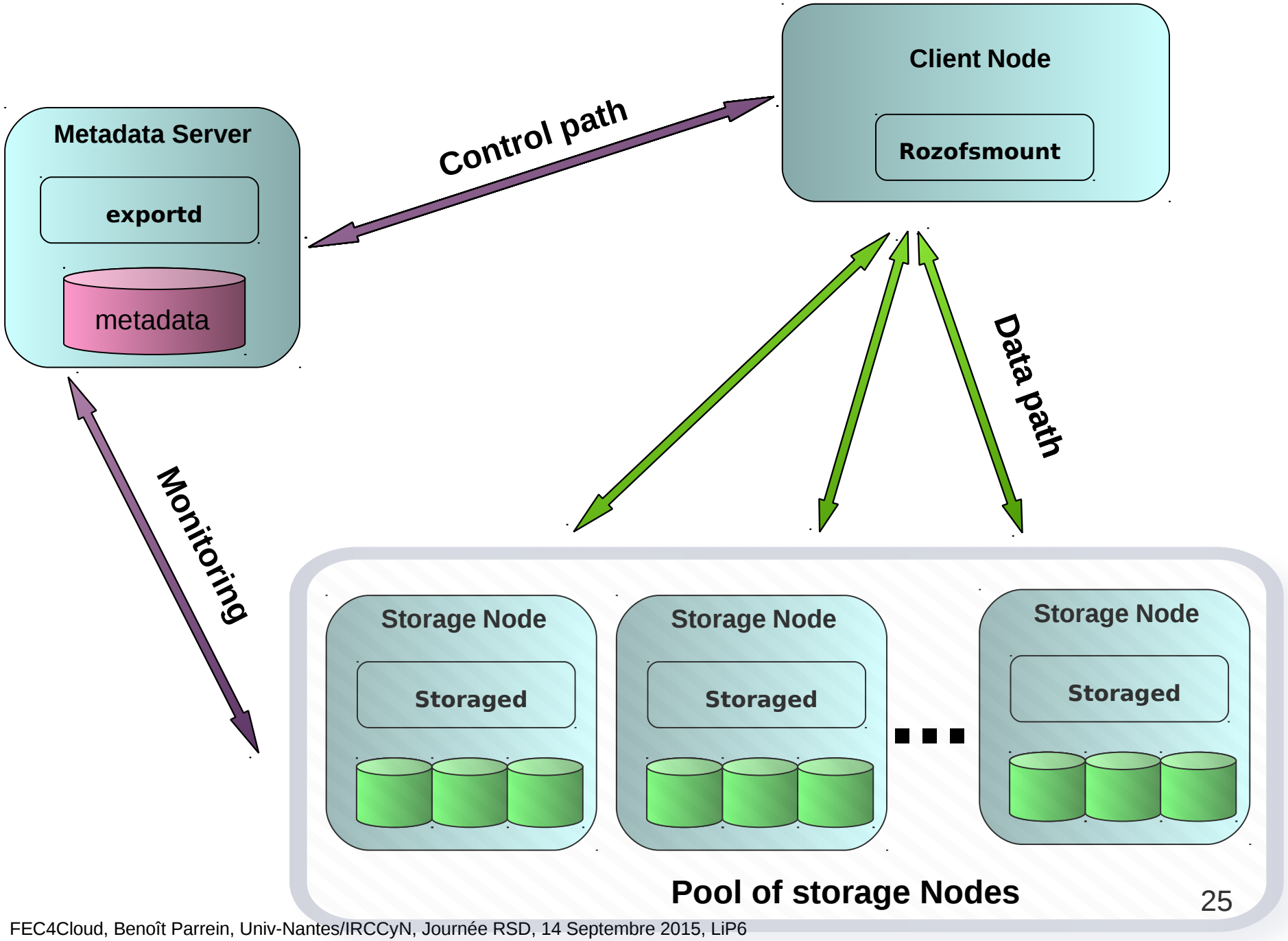
Distributed File Systems (DFS)

- 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**

RozoFS

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

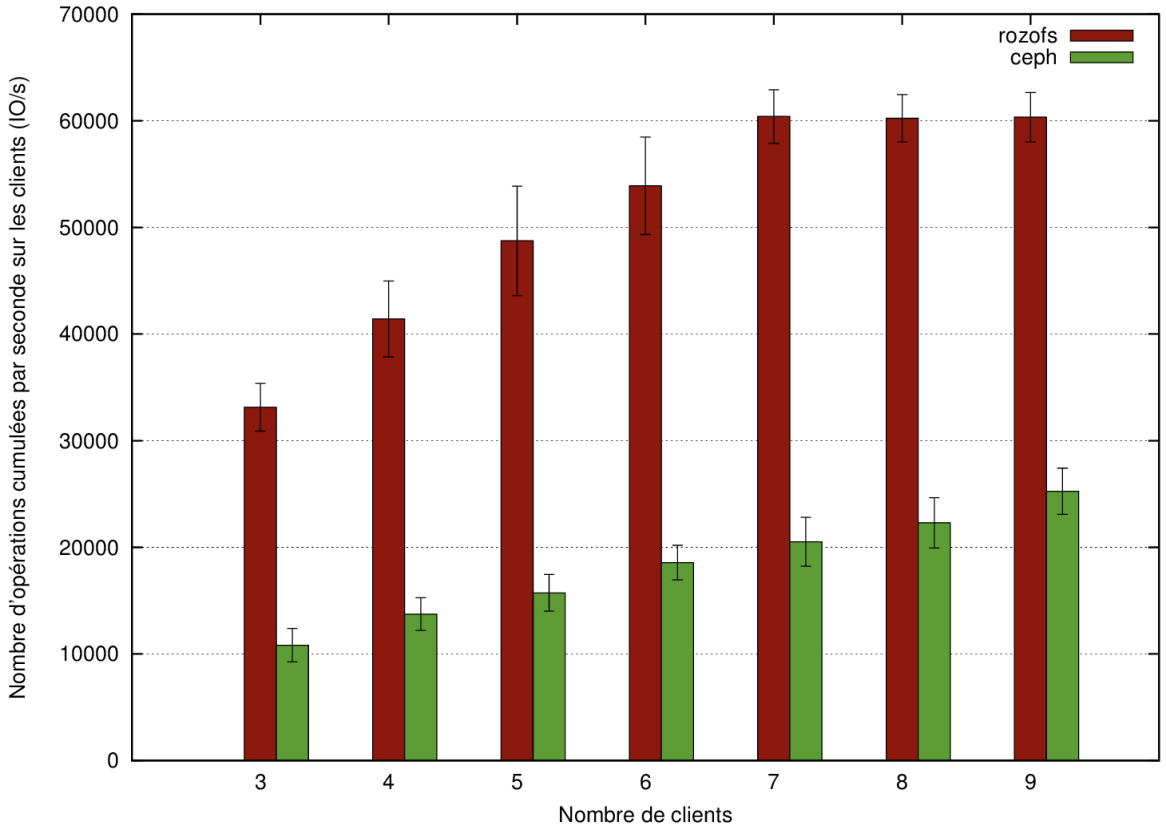

Client Node
Rozofsmount
Metadata Server
exportd
metadata
Control path
Data path
Monitoring
Storage Node
Storaged
Storage Node
Storaged
Storage Node
Storaged
Pool of storage Nodes

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 CephFS
- 10 Go de données, découpage fichier en blocs de 4Ko
- Accès séquentiels et aléatoires en lecture et écriture (via *IOZone*)
- 40 essais par mesure (moyenne, écart-type)

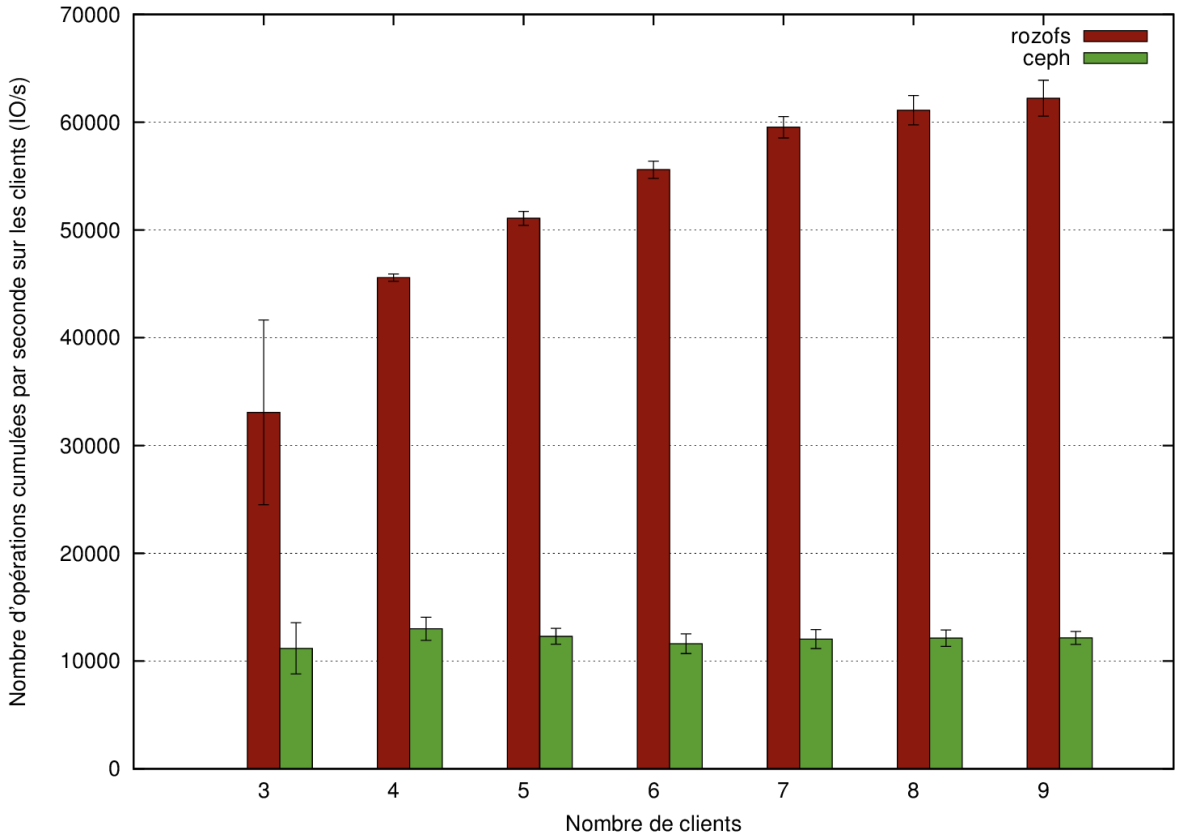
Expérimentations *in vivo* sur GRID5K

- Cluster de Nantes (18 nœuds): Intel Xeon E5@2.2 GHz, 10GbE
- Layout 1 – code Mojette (6,4) vs triplication CephFS
- **lecture aléatoire** (blocs 4Ko) par *IOZone*



Expérimentations *in vivo* sur GRID5K

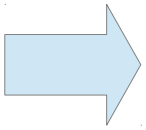
- Cluster de Nantes (18 nœuds): Intel Xeon E5@2.2 GHz, 10GbE
- Layout 1 – code Mojette (6,4) vs triplication CephFS
- **écriture aléatoire** (blocs 4Ko) par *IOZone*



Réduction de l'espace de stockage

- Capacité de stockage pour 10 Go utile

	Storage servers (GB)								Total (GB)
	1	2	3	4	5	6	7	8	
RozoFS	2.0	2.0	2.0	2.0	2.0	1.8	1.8	1.5	15.3
CephFS	6.0	5.3	5.0	5.0	4.7	4.3	2.2	2.0	34.5



Réduction d'un facteur 2 de l'espace de stockage
Par l'usage des codes à effacement

Cas d'usage de RozoFS

- Montage vidéo en ligne [Pertin et al., 2014]
- Exécution de machines virtuelles (QEMU/KVM)
- Exécution de bases de données distribuées PostgreSQL
- Streaming vidéos distribuées sur Rasbperry PI (*Fog Computing*)
- ...

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
- *Rozo Systems* emploie aujourd'hui 8 personnes (dont 5 en R&D) et s'ouvre à l'international (USA)

Merci de votre attention!