Enable and Optimize Erasure Code for Big data on Hadoop

High Performance Computing, Intel
Jun Jin (jun.i.jin@intel.com)

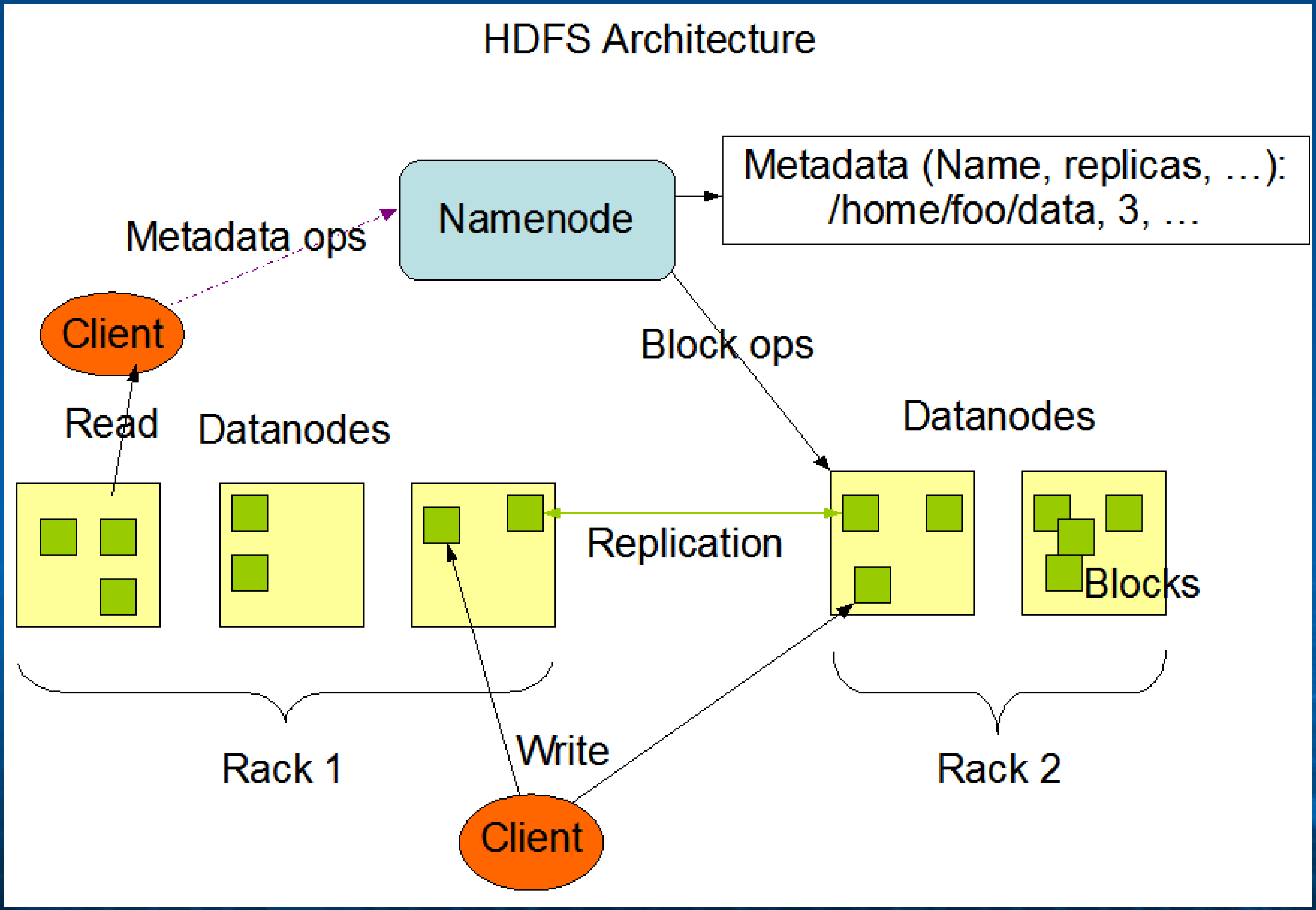
Agenda

- Background and overview
- Codec performance
- Block placement Policy
- Performance evaluation
- Local Repairable Codes
- Summary

Overview

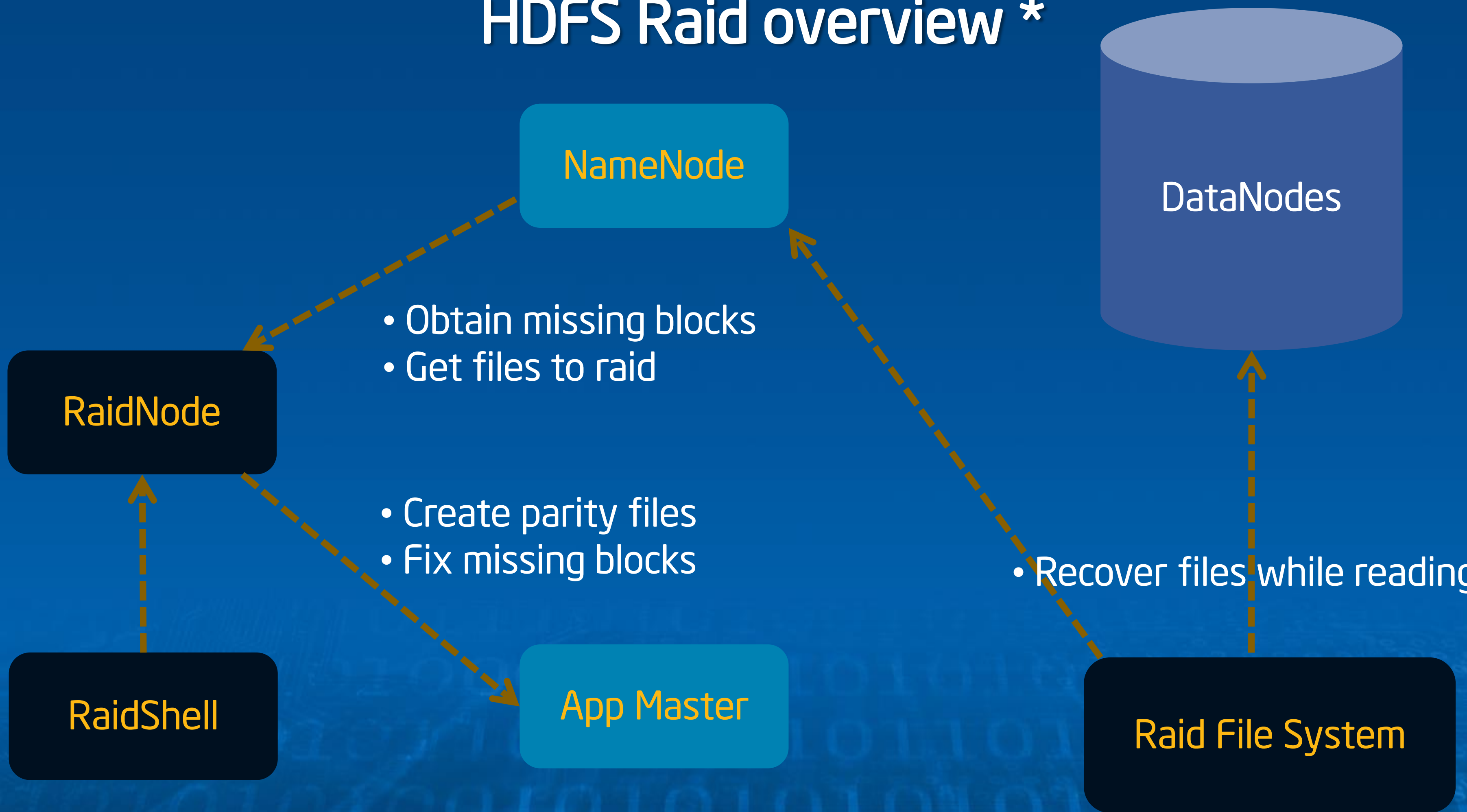
- Hadoop and HDFS is popular now, the driver of growth in IPDC market
- Data is growing faster than infrastructure
- 3x replication of everything is too expensive, especially on Petabyte scale
- Use erasure codes for cold data
  - LRU to track data block, code cold files
  - Classical codes are unsuitable for distributed environment
  - Google GFS2(Colossus), Microsoft Azure and facebook(hdfs-20)

Revisit HDFS *



* From Apache website http://hadoop.apache.org/docs/stable/hdfs_design.html

HDFS Raid overview *



DRFS: provides access to files and transparently recovers any corrupt or missing blocks encountered when reading a file
RaidNode: a daemon that creates and maintains parity files for all data files stored in the DRFS
BlockFixer, which periodically recomputes blocks that have been lost or corrupted
RaidShell, allows administrator to manually trigger the recomputation of missing or corrupt blocks and check for files
ErasureCode, which provides the encode and decode of the bytes in blocks

* From Apache website http://wiki.apache.org/hadoop/HDFS-RAID

Replication vs Erasure codes

- Erasure coding, facebook's approach for code data

640MB data => 10 blocks

40% overhead

1 2 3 4 5 6 7 8 9 10

Source file

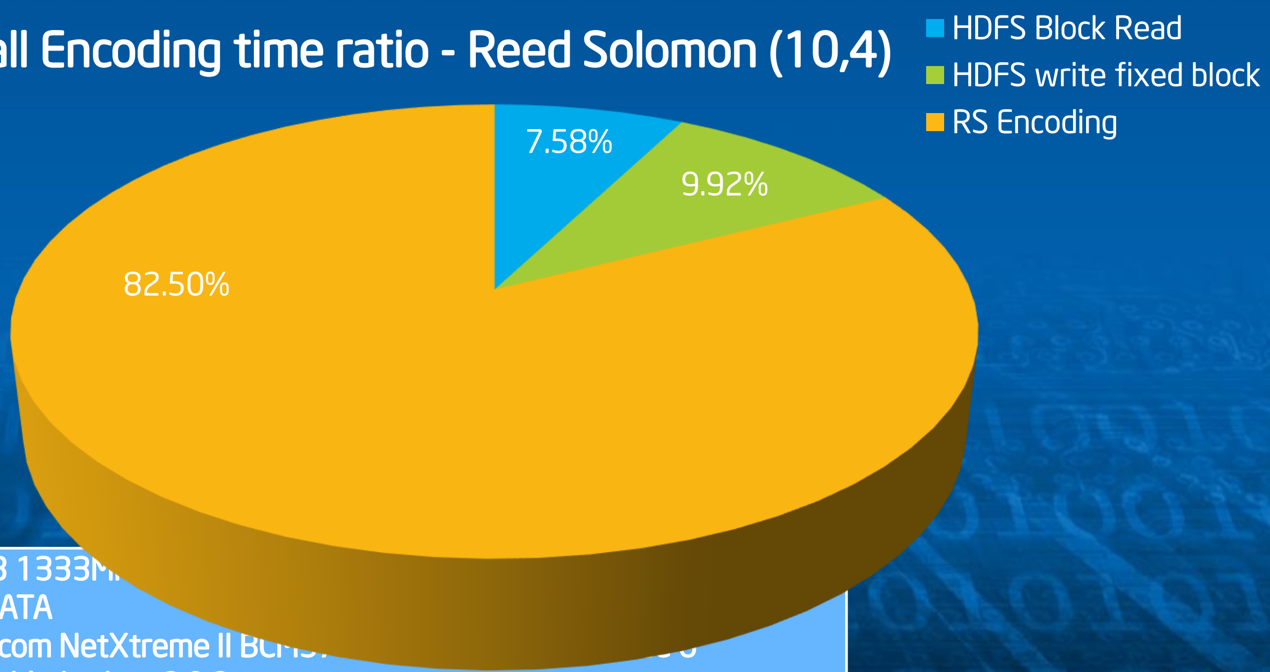
P1 P2 P3 P4

Parity file

In facebook, older files are switched from 3 replication to (10, 4) Reed Solomon

Overview - Performance characterization

- Perf. characterization for baseline HDFS Raid (v2.0.2)
  - Reed Solomon(RS) encoding is CPU intensive, takes 80+% of time
  - Network and disk IO takes ~20%
  - High bin CPU could greatly improve the codec performance



NN: 1 EP server (X5680)	• 64G DDR3 1333M [illegible] • 2 x 1TB SATA • 2 x Broadcom NetXtreme II BCM [illegible] • RH 6.2 64 bit, hadoop2.0.2
DN: 12 x EP server(E5645):	• 36G DDR3 1333Mhz • 5 x 500GB SATA • 2 x Broadcom NetXtreme II BCM5709 Gigabit, bind by mode 6 • RH 6.2 64 bit, hadoop2.0.2

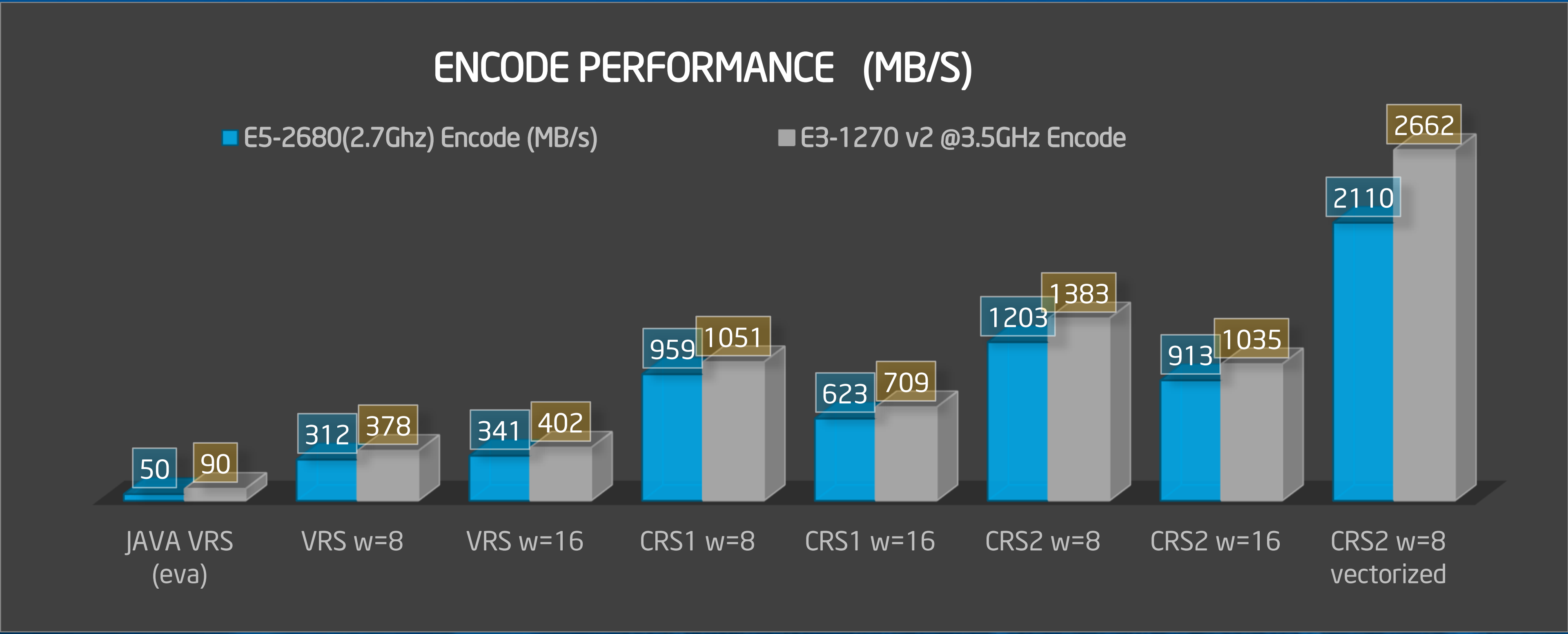
Agenda

- Background and overview
- Codec performance
- Block placement Policy
- Performance evaluation
- Local Repairable Codes
- Summary

Performance of Encoder

Summary:

- Pure encode/decode performance, no IO included
- Best performance based on successfully vectorize XOR loop for CRS2



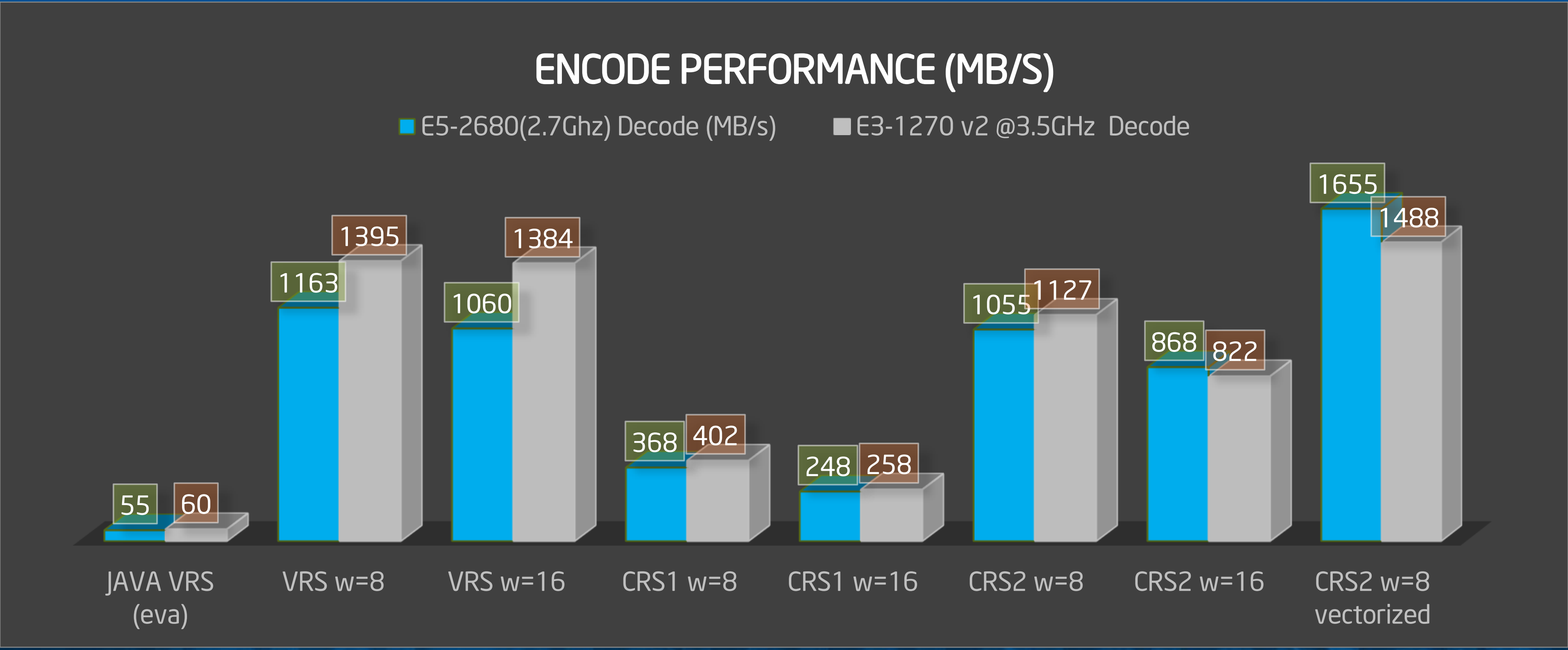
VRS = Vandermonde Reed Solomon
CRS1 = Cauchy Reed Solomon Original
CRS2 = Cauchy Reed Solomon Good

Note: Test based on Jerasure library: http://web.eecs.utk.edu/~plank/plank/papers/CS-08-627.html

Performance of Decoder

Summary:

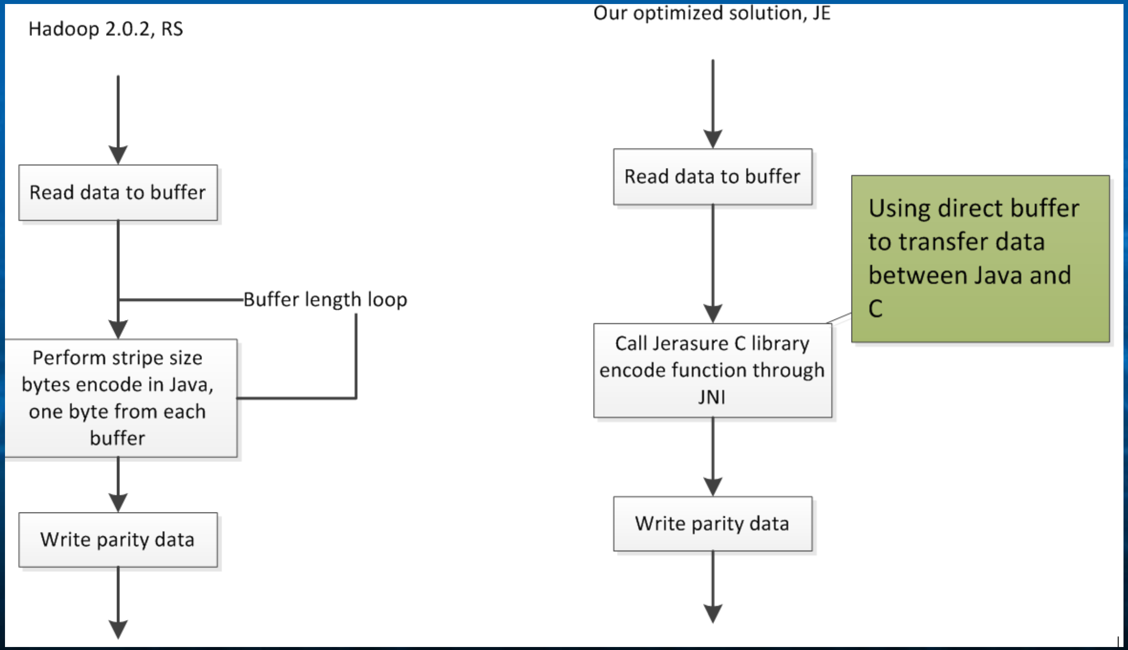
- Pure encode/decode performance, no IO included
- Best performance based on successfully vectorize XOR loop for CRS2



Note: Test based on Jerasure library: http://web.eecs.utk.edu/~plank/plank/papers/CS-08-627.html

New codec integration

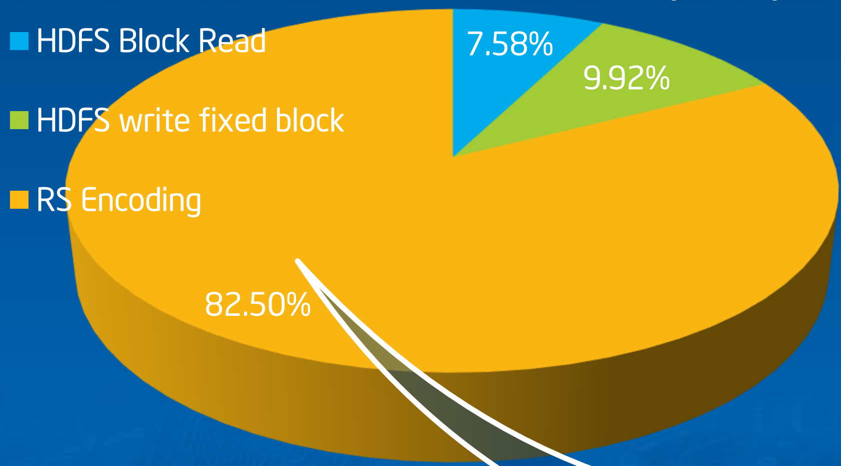
- The original codec in HDFS implemented Classical Reed-Solomon (RS), which proved very slow
- Cauchy RS is much better than Classical RS
- Native C code for Cauchy RS, from Jerasure 1.2
- Used direct buffer for Java and C communication to eliminate unnecessary memory copy



New codec integration in HDFS

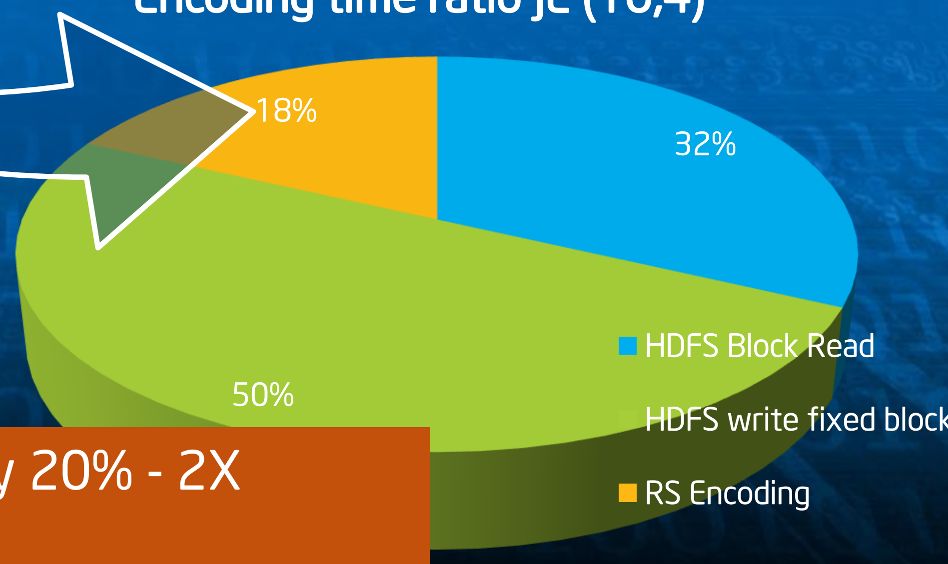
- Performance compare

Encoding time ratio RS (10,4)



NN: 1 EP server (X5680)	• 64G DDR3 1333Mhz • 2 x 1TB SATA • 2 x Broadcom NetXtreme II BCM5709 Gigabit, bind by mode 6 • RH 6.2 64 bit, hadoop2.0.2-rc1
DN: 12 x EP server(E5645)	• 36G DDR3 1333Mhz • 5 x 500GB SATA • 2 x Broadcom NetXtreme II BCM5709 Gigabit, bind by mode 6 • RH 6.2 64 bit, hadoop2.0.2-rc1

Encoding time ratio JE (10,4)



Overall performance greatly improved by 20% - 2X
Bottleneck move to network now.

Intel® ISA-L Library

- Intel® Intelligent Storage Acceleration Library
- Algorithmic Library to address key Storage market segment needs
  - Optimized for Intel Architecture
  - Enhances efficiency, data integrity, security/encryption, erasure codes, compression, CRC, AES, etc.
- Benefits of using Intel® ISA-L
  - Highly optimized based on Intel New Instructions
- Working on ISA-L library for HDFS erasure coding

Agenda

- Background and overview
- Codec performance
- Block placement Policy
- Performance evaluation
- Local Repairable Codes
- Summary

New Block Placement Policy

- Problem: Encoding brings heavy network and disk IO
- Blocks are randomly placed in HDFS by default, data locality can't be guaranteed.
- Need a new block placement policy.

Agenda

- Background and overview
- Codec performance
- Block placement Policy
- Performance evaluation
- Local Repairable Codes
- Summary

Performance evaluation

- Workload 1: 1TB data, each file 5GB, block size 256MB
- Workload 2: 145 GB data, each file 2.5GB, block size 256MB
- Used 60 containers (5 on each server), 59 for map tasks and 1 for Application Master
- Compared 3 different code
  1. Baseline code with VRS codes
  2. Integrated JE code with default block placement
  3. Integrated JE code with new block placement

NN: 1 EP server (X5680)	• 64G DDR3 1333Mhz • 2 x 1TB SATA • 2 x Broadcom NetXtreme II BCM5709 Gigabit, bind by mode 6 • RH 6.2 64 bit, hadoop2.0.2-rc1
DN: 12 x EP server(E5645)	• 36G DDR3 1333Mhz • 5 x 500GB SATA • 2 x Broadcom NetXtreme II BCM5709 Gigabit, bind by mode 6 • RH 6.2 64 bit, hadoop2.0.2-rc1

New Block Placement

- Performance overview on two data set

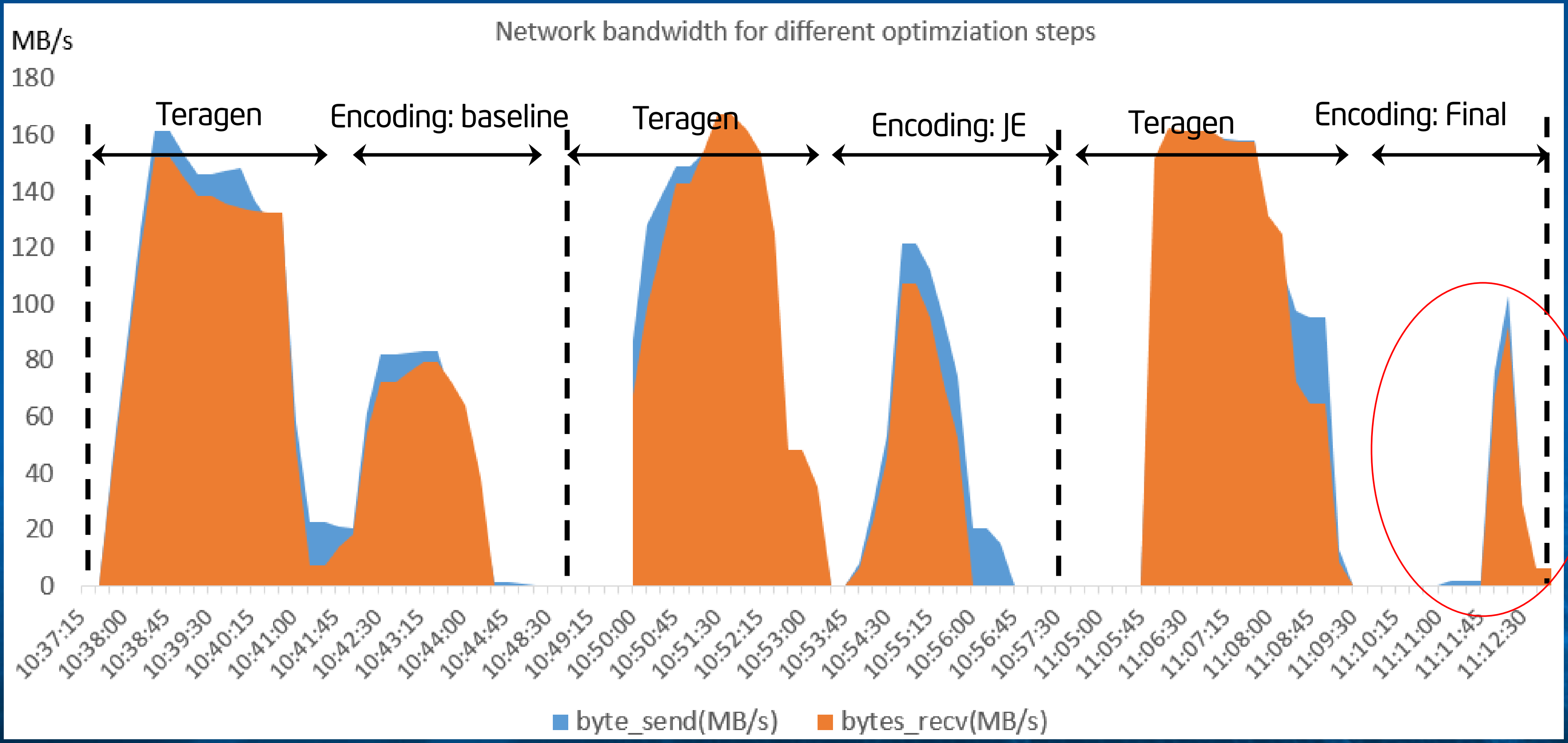
Workload 1: 145GB data	Time	Speedup
Baseline: Default + VRS	165s	1
Integrated JE codec	135s	1.22X
Integrated JE + new block placement	69s	2.4X

Workload 2: 1 TB data	Time	Speedup
Baseline: Default + VRS	1086s	1
Integrated JE codec	738s	1.47X
Integrated JE + new block placement	427s	2.53X

Performance improved up to 2.53X over the baseline

New Block Placement

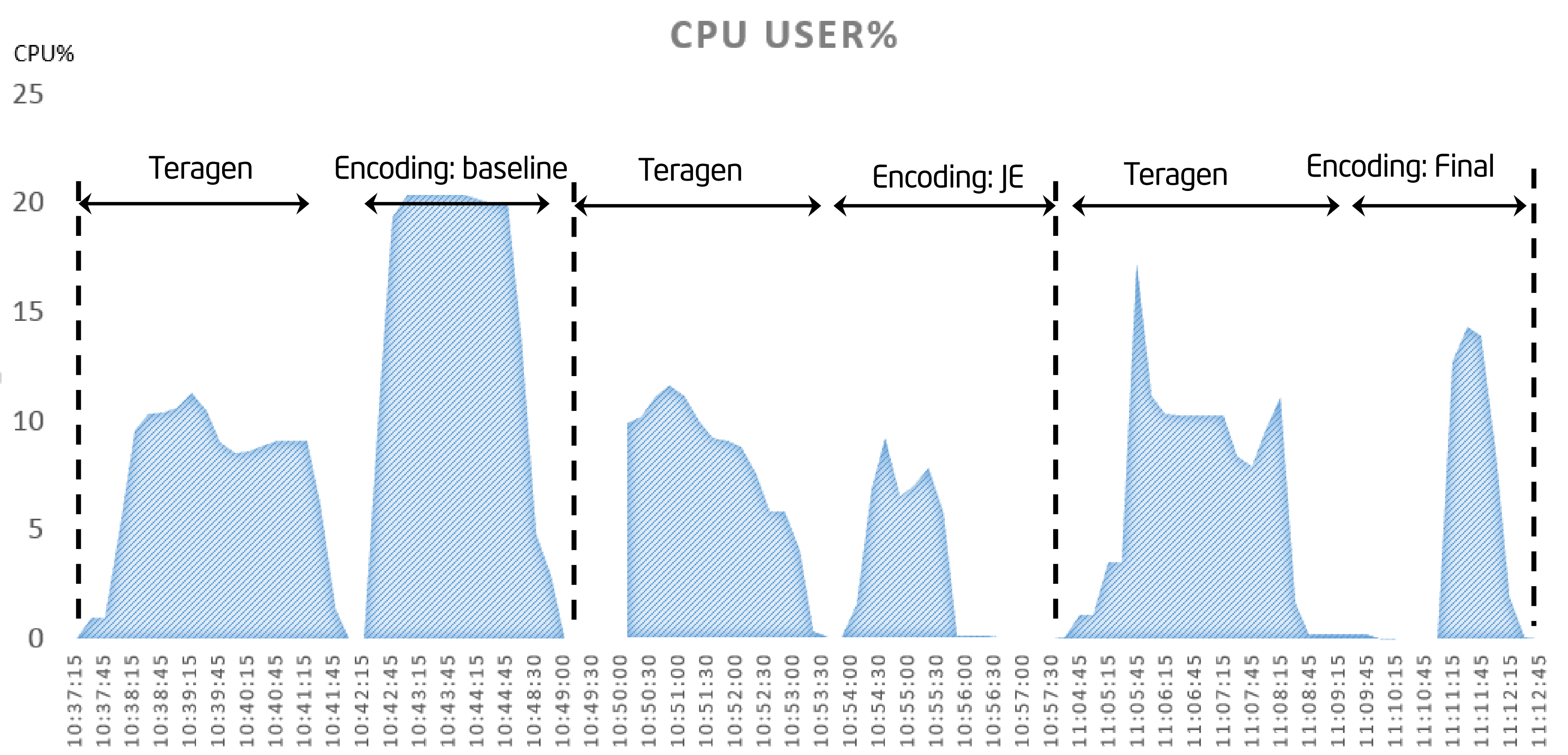
- Network BW greatly reduced based on new block placement



Notes: Use Teragen to generate source data then do encode

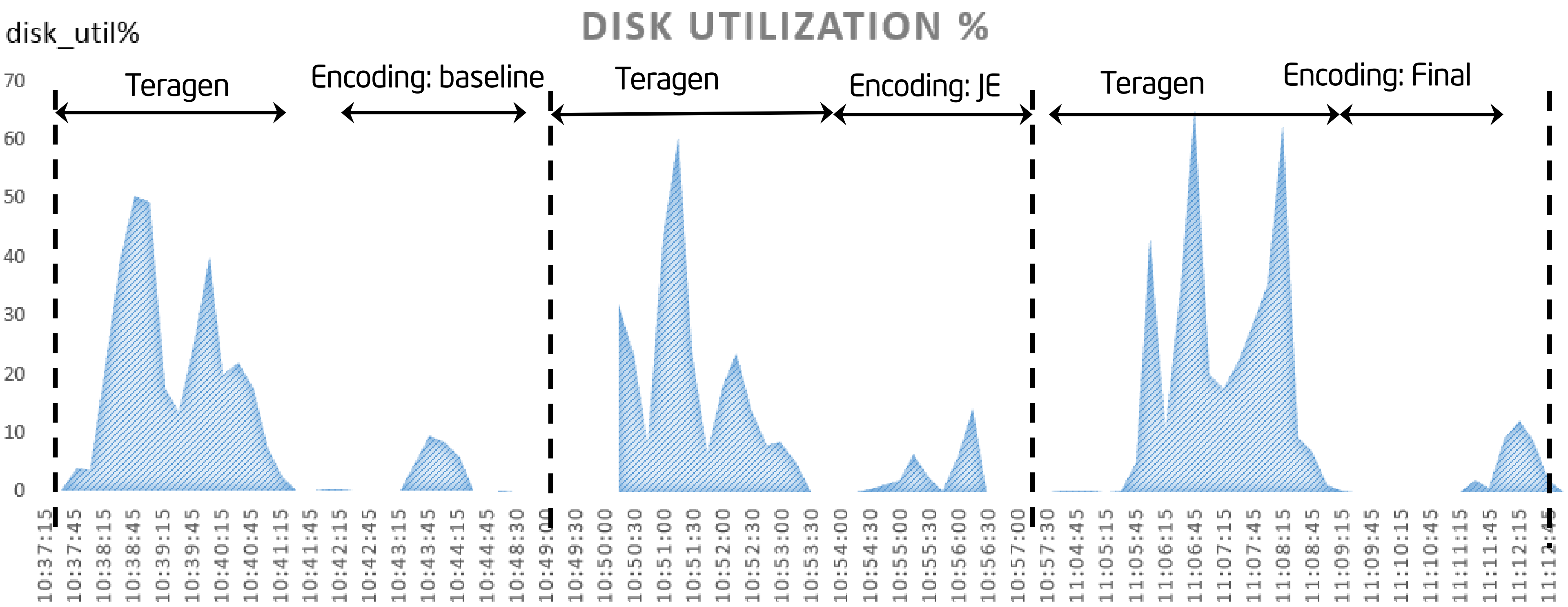
New Block Placement

- CPU utilization for different optimization steps



New Block Placement

- Disk utilization for different optimization steps

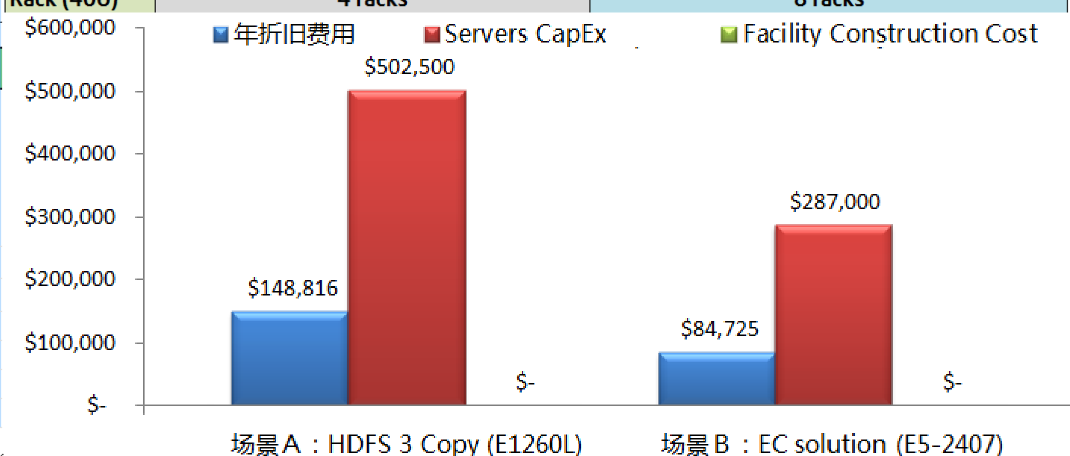


		场景A：HDFS 3 Copy (E1260L)	场景B：EC solution (E5-2407)
30			
31	服务器功耗总和	29	22
32	存储器功耗总和	-	-
33	网络设备功耗和	1	1
34	**关键负载**	30	23
35	设施功耗（非关键负载）	18	14
36	**TOTAL Energy**	49	37
37	**年运营费用**		
38	服务器能耗	$ 94,805	$ 72,603
39	存储器能耗	$ -	$ -
40	网络设备能耗	$ 3,889	$ 1,945
41	基础设施能耗	$ 59,217	$ 44,729
42	**总能耗费用(运营费用)**	$ 157,911	$ 119,276
43	动力&冷却设备折旧	$ -	$ -
44	其他设施折旧	$ -	$ -
45	服务器折旧	$ 145,017	$ 82,826
46	存储器折旧	$ -	$ -
47	网络折旧	$ 3,798	$ 1,899
48	**年折旧费用**	$ 148,816	$ 84,725
49			
50	**Annual OpEx and Dep Expense**	$ 306,727	$ 204,001
51			
52	Storage CapEx	$ -	$ -
53	Network CapEx	$ 16,000	$ 8,000
54	Servers CapEx	$ 502,500	$ 287,000
55	Power & Cooling CapEx	$ -	$ -
56	Other Facility CapEx	$ -	$ -
57	Facility Construction Cost	$ -	$ -
58			
59	**资产性费用总和**	$ 518,500	$ 295,000
60			
61	支持人员费用 (估)	$ 428,571	$ 200,000
62	基础设施费用（估）		
63			
64	架高地板面积	120	60
65	平均机柜功率密度	3.81	5.75
66	总性能值	71	70
67			
68	**Upfront Capex**	$ 518,500	$ 295,000
69	**Annual Operating Expense**	$ 157,911	$ 119,276
70	**Annual Depreciation Expense**	$ 148,816	$ 84,725
71	**TCO**	**$ 940,599**	**$ 613,827**

TCO analysis (E3 vs E5)

- Assume to store 1PB data, EC with (10,4) codec needs 70 servers, while 3 copy solution needs about 150 servers.
- TCO for 3 copy 940K vs EC 614K
- $/GB： 0.9 vs 0.59

	EC soluton 70 servers			original HDFS 3 copy: 150 servers		
	type	Price ($)	power (W)	type	Price ($)	power (W)
CPU	E5-2407(2.2G) x 2	500	160	E3-1260L (2.4G)	300	45
memory	48GB	700	60	24GB	350	30
disk	2T x 10	2000	100	2T x 10	2000	100
others	network, boards etc	900	20	network, boards	700	20
Total HW	70 servers	287000	23800	150 servers	502500	29250
Rack (40U)	4 racks			8 racks		



Agenda

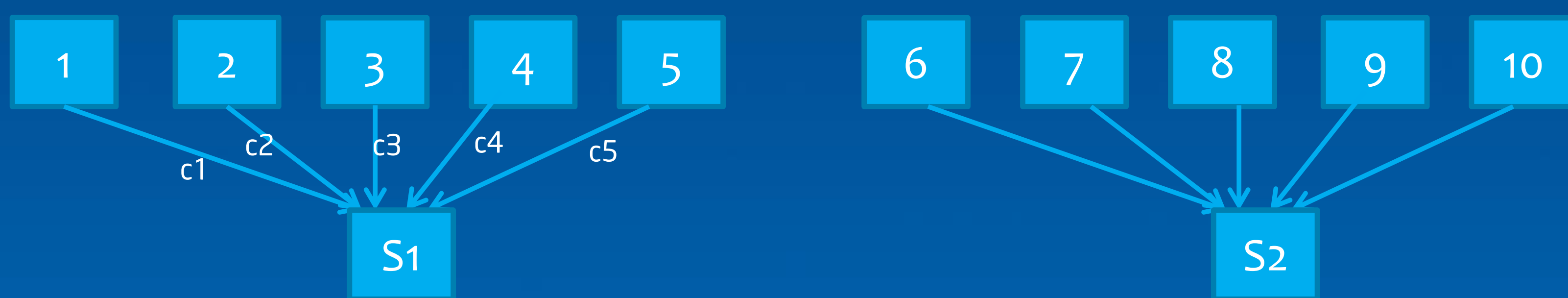
- Background and overview
- Codec performance
- Block placement Policy
- Performance evaluation
- Local Repairable Codes
- Summary

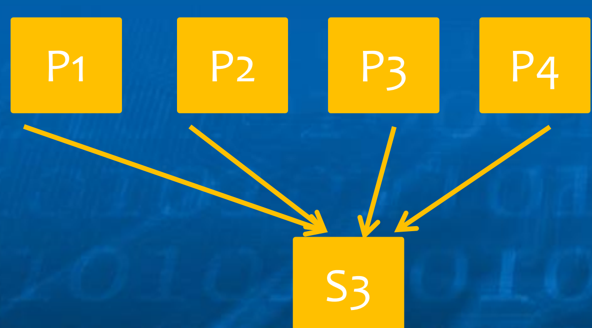
Local Repairable Codes*

- Local repairable codes to reduce network/disk IO during decoding

60% overhead

Source file





P1, P2, P3, P4: Reed Solomon encoding
Use the first 5 and second 5 blocks create 2 local parity blocks,S1, S2
One lost block requires only 5 other blocks to be read.
Choose coefficients to let s3 = s1 + s2

Single failures represent 99.75% of recoveries**

* Erasure Coding in Windows Azure Storage
** Source: Rethinking Erasure Codes for Cloud File Systems: Minimizing I/O for Recovery and Degraded Reads
Simple Regenerating Codes: Network Coding for Cloud Storage, Novel Codes for Cloud Storage:https://mhi.usc.edu/files/2012/04/Sathiamoorthy-Maheswaran.pdf

Other considerations

- HAR support
- Block balance
  How to balance the blocks when new node added or deleted
- Set replication
- Raid Shell tool
- Parallel reader
- And many others…

Summary

- Jerasure CRS codec proved bring good performance advantage over baseline Java implementation, we working on ISA-L now
- Developed a new block placement, which effectively reduces the network bandwidth
- Working on local repairable codes.
- Erasure Code is an effective way to reduce storage cost for cold data.