

The Logic of Physical Garbage Collection in Deduplicating Storage

Fred Douglis

Abhinav Duggal

Philip Shilane

Tony Wong

Dell EMC

Shiqin Yan

University of Chicago

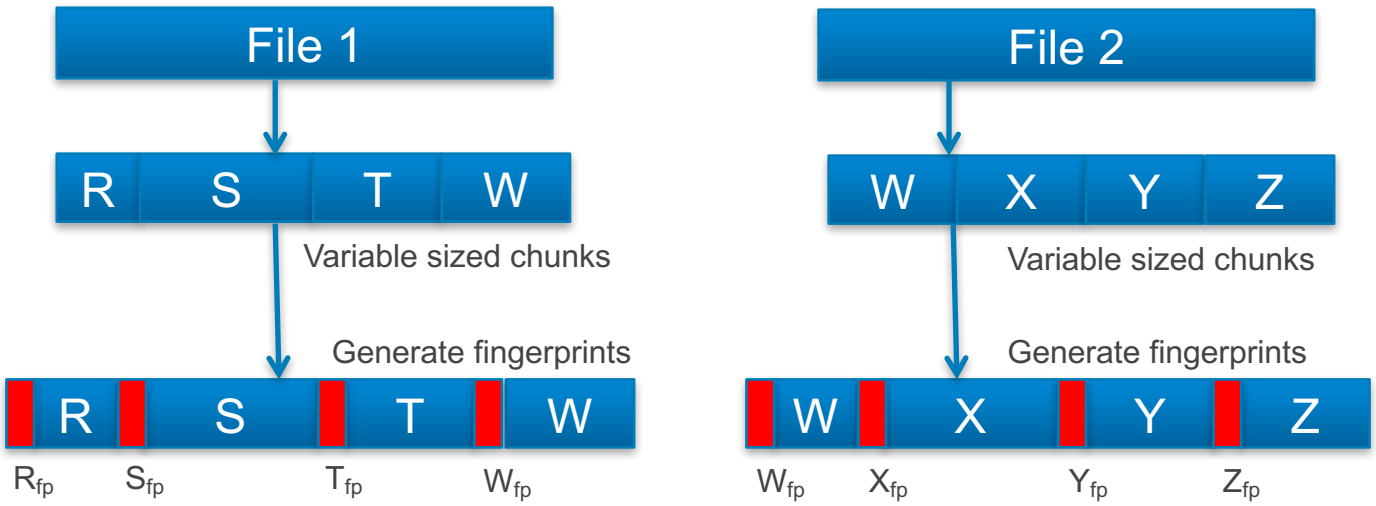
Fabiano Botelho

Rubrik

DELL EMC

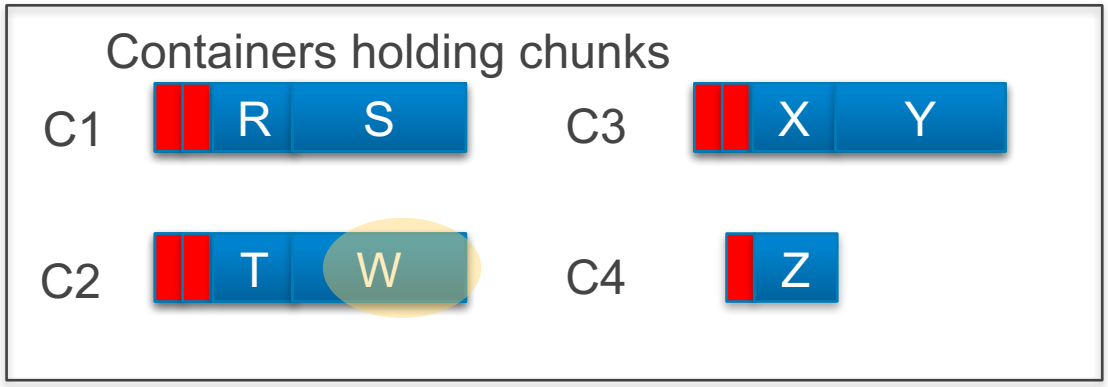
© Copyright 2017 Dell Inc.

Deduplication in *Data Domain Filesystem (DDFS)*



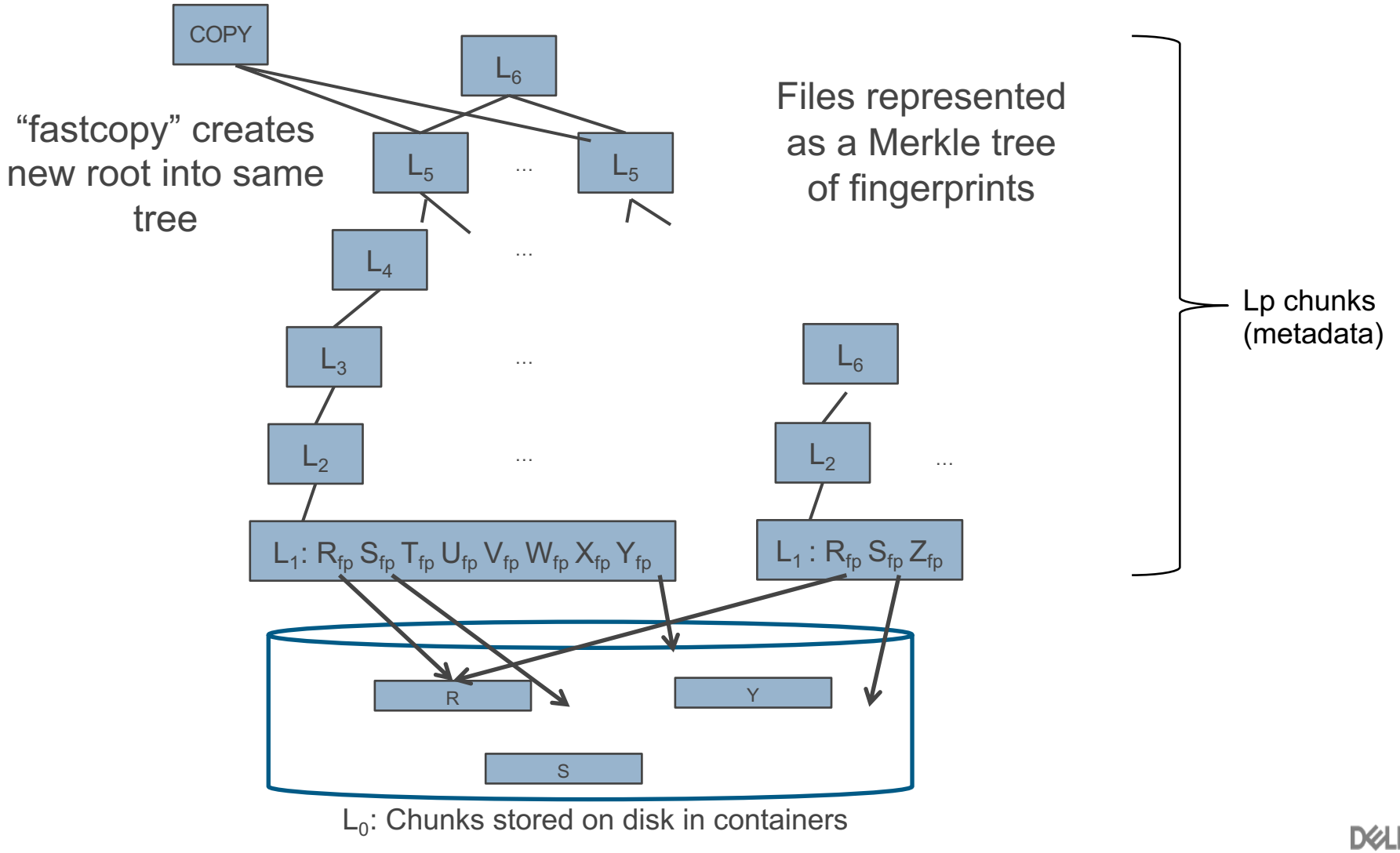
Fingerprint Index

fp	CID
R	C1
S	C1
T	C2
W	C2
X	C3
Y	C3
Z	C4



© Copyright 2017 Dell Inc.

File Representation in DDFS



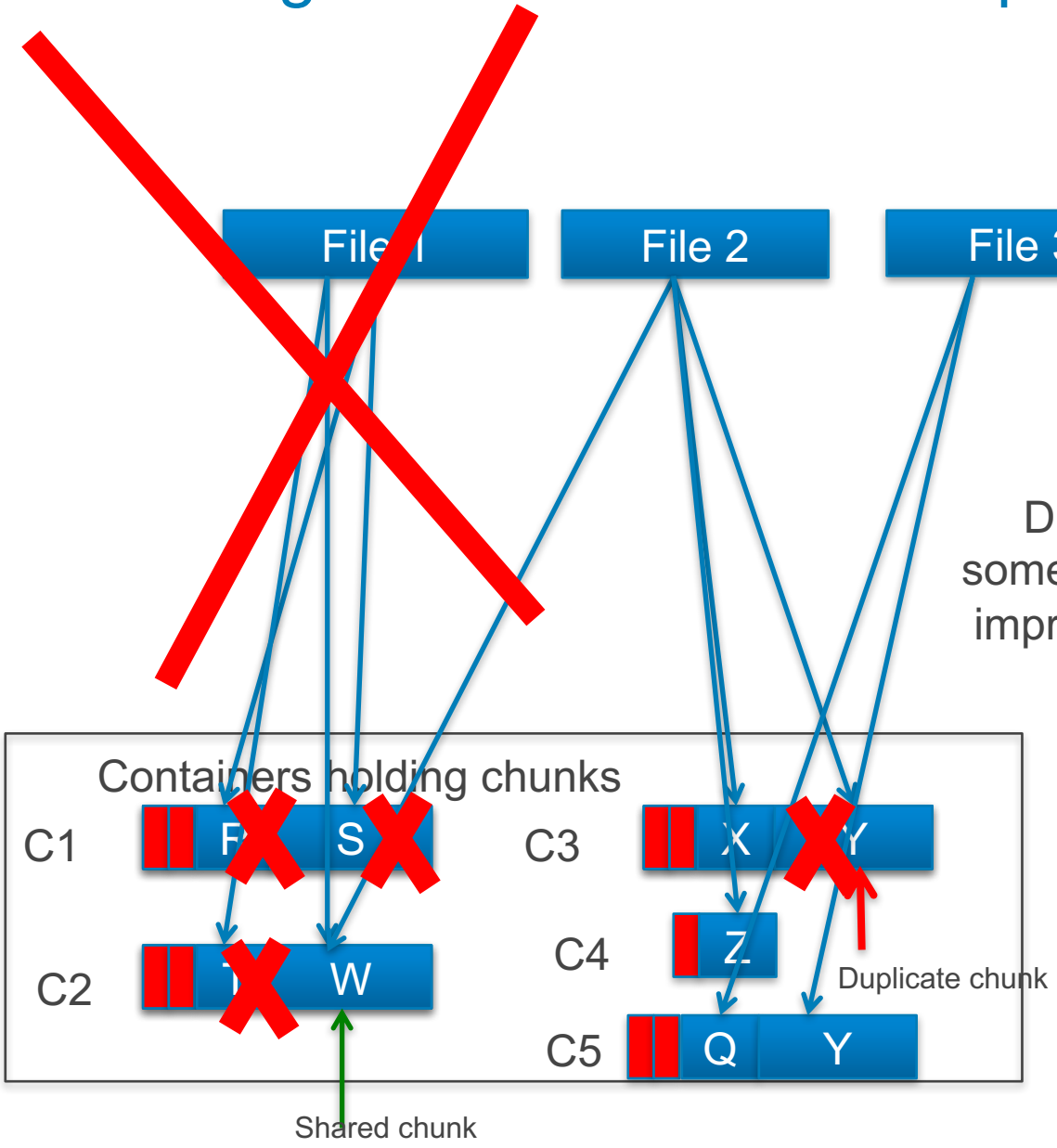
© Copyright 2017 Dell Inc.

Deduplication Workloads on Data Domain

- Traditional backups
  - Weekly full and daily incremental backups
    - Full backups tend to be very large – 100GBs to TBs
    - Much content in full backups repeats previous full
  - Typically, 10-20x total compression (TC)
    - 20x TC = 10x dedup and 2x compression
- New workloads
  - “Synthetic” full backups
    - Send changes and a recipe to create a single full backup from some previous backup
    - Daily fulls
    - High TC (100x-400x or higher)
  - High file count
    - 100M to 1 billion small files

© Copyright 2017 Dell Inc.

Garbage Collection in a Deduplication Filesystem



Duplicates are sometimes written to improve throughput

Fingerprint Index

fp	CID
R	C1
S	C1
T	C2
W	C2
X	C3
Y	C3
Z	C4
Q	C5
Y	C5

© Copyright 2017 Dell Inc.

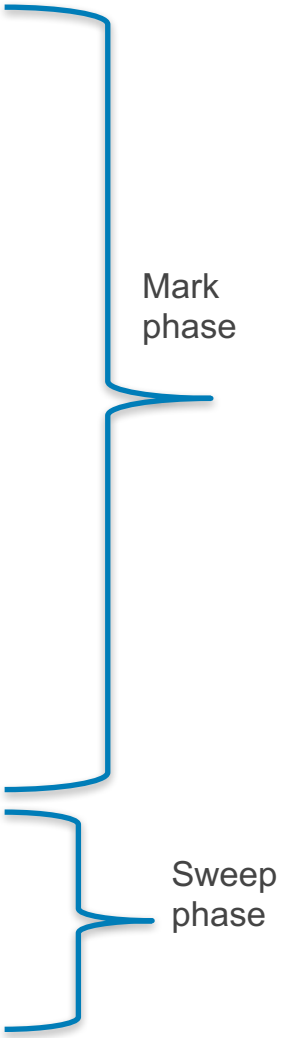
Evolution of GC in DDFS

- Logical GC (LGC)
  - Depth-first traversal of per-file Merkle tree on disk to mark live chunks in memory
  - In-memory data structures may not allow system to track all chunks, so an extra mark phase ("pre-phases") is used when necessary
- Physical GC (PGC)
  - Breadth-first traversal of the physical layout of Merkle trees to mark live chunks in memory
  - Similar to LGC, pre-phases may be needed
- Phase-optimized Physical GC (PGC+)
  - Improvement over PGC by removing pre-phases, plus other optimizations

© Copyright 2017 Dell Inc.

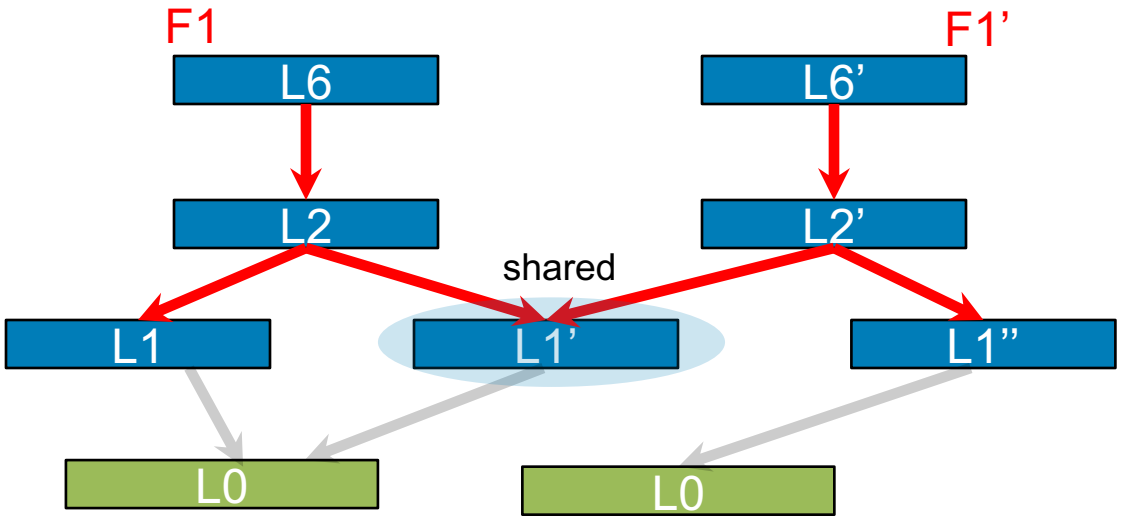
Logical GC Phases

- Merge
  - Merge in-memory Index on disk
- Enumeration
  - Depth-first walk and mark live chunks in an in-memory Bloom filter called **live vector**
- Filter
  - Create **live instance vector** **(also a Bloom filter)** from live vector to remove the duplicates
- Select
  - Select best containers to compact
- Copy
  - Copy live chunks from selected containers into new containers and delete old containers



© Copyright 2017 Dell Inc.

Enumeration Phase (Logical GC)



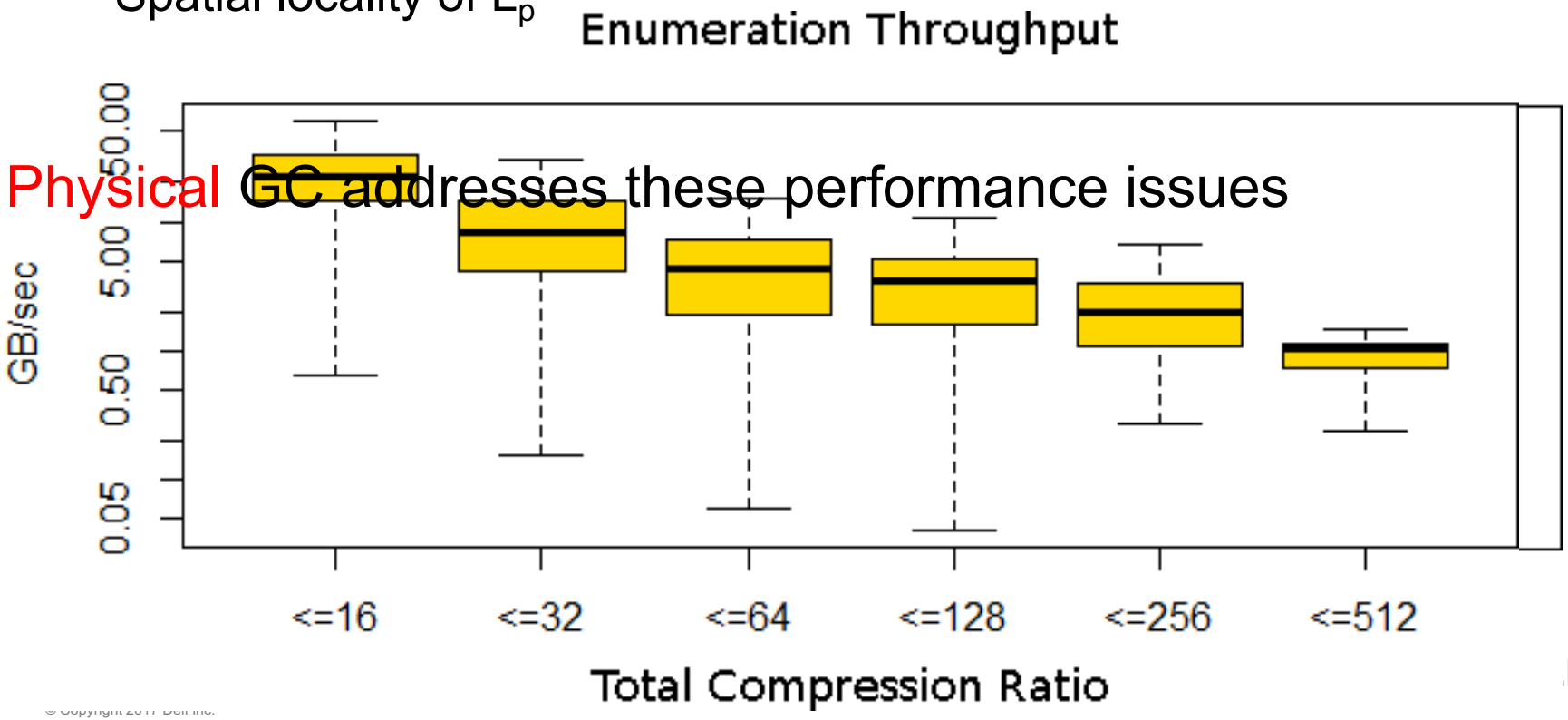
Only L_p chunks are traversed

© Copyright 2017 Dell Inc.

Logical GC →Physical GC

Logical enumeration performance is sensitive to the following parameters

- Total compression factor
- Number of small files
- Spatial locality of L_p

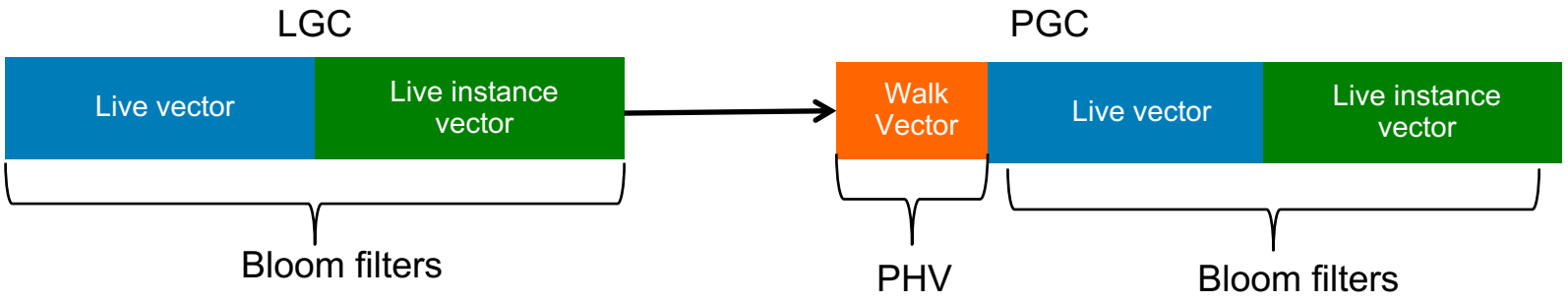


Physical GC addresses these performance issues

© Copyright 2017 Dell Inc.

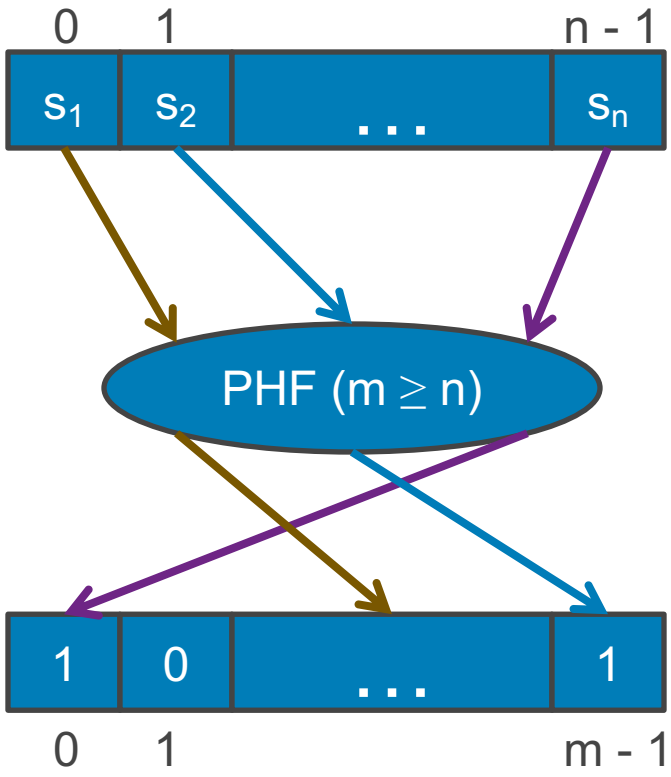
Physical GC (PGC)

- Uses breadth-first walk instead of per-file depth-first walk during enumeration
- Uses Perfect Hash Vector(PHV) to store L_Ps for assisting the breadth-first walk
  - Uses less memory
  - Needed for doing checksums to prevent corruption
- New analysis phase to build Perfect Hash Functions for L_Ps
- Remaining phases are same as logical GC



© Copyright 2017 Dell Inc.

Collision Free - Perfect Hashing Vector (PH_{vec})



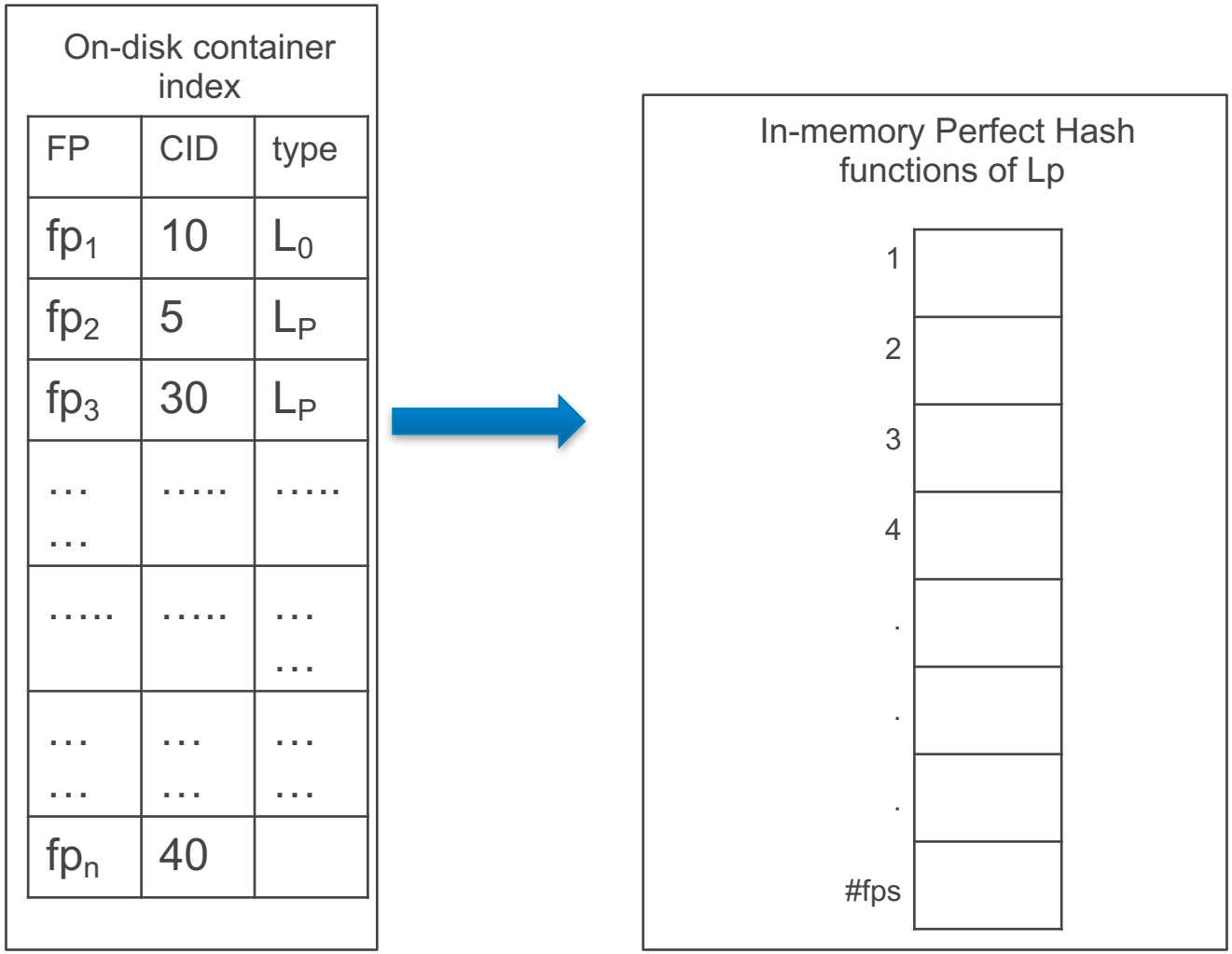
Fingerprint set S

Collision-free hash function which maps a fingerprint to a unique position in a bit vector

Bit vector

© Copyright 2017 Dell Inc.

Analysis Phase



© Copyright 2017 Dell Inc.

Benefits & Costs of Physical Enumeration

- Pro: Sequential scan of containers on disk
  - All L_6, then all L_5, down to L_1s
  - Relatively few containers store high-level metadata
  - No need to keep revisiting same L_p containers due to fastcopy (high deduplication)
- Con: extra analysis cost doesn't help "traditional" workloads
- … and due to pre-phases we may have to run analysis twice!

© Copyright 2017 Dell Inc.

LGC and PGC phases (including pre-phases)

Logical GC

1. Pre-merge
2. Pre-enumeration
3. Pre-filter
4. Pre-select
5. Candidate
6. Enumeration
7. Merge
8. Filter
9. Copy
10. Summary

Pre-phases/sampling phases

Physical GC

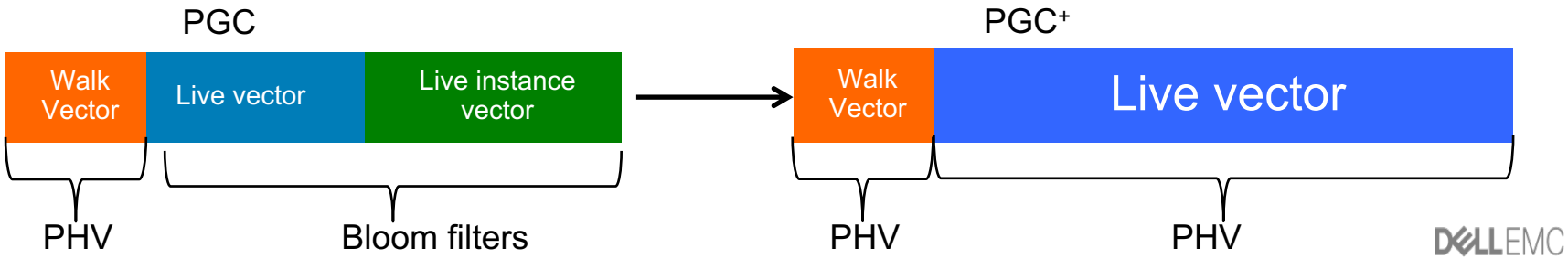
1. Pre-merge
2. Pre-analysis
3. Pre-enumeration
4. Pre-filter
5. Pre-select
6. Merge
7. Analysis
8. Candidate
9. Enumeration
10. Filter
11. Copy
12. Summary

Pre-phases / sampling phases

© Copyright 2017 Dell Inc.

Physical GC → Phase-optimized Physical GC

- Limitations of Physical GC
  - Adds 2 extra phases (pre-analysis and analysis)
  - Slightly degrades GC performance for customers with traditional backup workloads
- Motivation for Phase-optimized Physical GC (PGC+)
  - Avoid pre-phases by representing all chunks in memory
  - Can we use Perfect hash as a live vector?
    - Need only 2.7 bits per fingerprint instead of a 6 bits in Bloom filter
  - Can we maintain duplicate recipe without using a Bloom filter?
    - Get 50% memory back



© Copyright 2017 Dell Inc.

Phase-optimized Physical GC (PGC+) Phases

1. Merge
2. Analysis
3. Enumeration
4. Select
5. Copy
6. Summary

© Copyright 2017 Dell Inc.

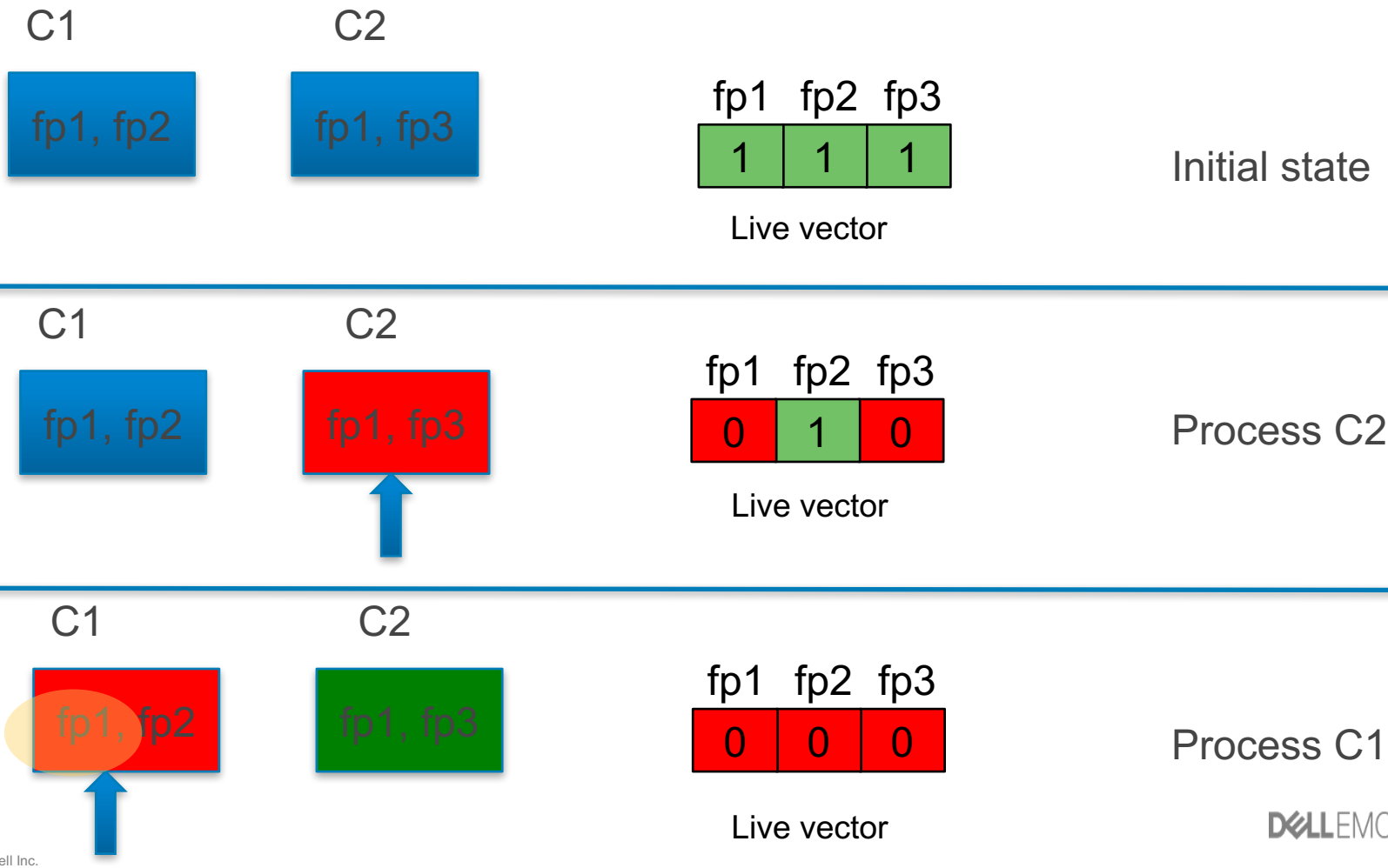
PGC+ Analysis and Enumeration

- Replace Bloom filter with Perfect Hash vector for tracking live and dead chunks
- In analysis phase build two Perfect hash vectors
  - Lp vector called the walk vector (similar to PGC)
  - All fingerprints(Lp + L0) based Perfect Hash vector called live vector
- Perfect hashing optimizations
  - NUMA-aware Perfect Hashing
  - Cache prefetching of Perfect hash functions and values in the Perfect Hash Vector

© Copyright 2017 Dell Inc.

PGC+ Copy phase

Dynamically remove duplicates during Copy phase



© Copyright 2017 Dell Inc.

Evaluation

- Deployed systems
  - Comparison of GC runs for systems upgraded from LGC to PGC
- Controlled experiments on 4 systems
  - Comparison of LGC vs PGC vs PGC+
    - One phase versus two phase GC
  - DD860 used as default for all experiments
  - Workload used was Synthetic dataset similar to some past deduplication work (e.g., Botelho, et al., FAST 2012)

Systems	DD2500	DD860	DD890	DD990
CPU(cores*GHz)	8*2.2 GHz	16*2.53 GHz	24*2.8 GHz	40*2.4 GHz
Mem(GB)	64 GB	70 GB	94 GB	256 GB
Physical Capacity (TB)	122 TB	126 TB	167 TB	319 TB

© Copyright 2017 Dell Inc.

Deployed System Results- LGC vs PGC

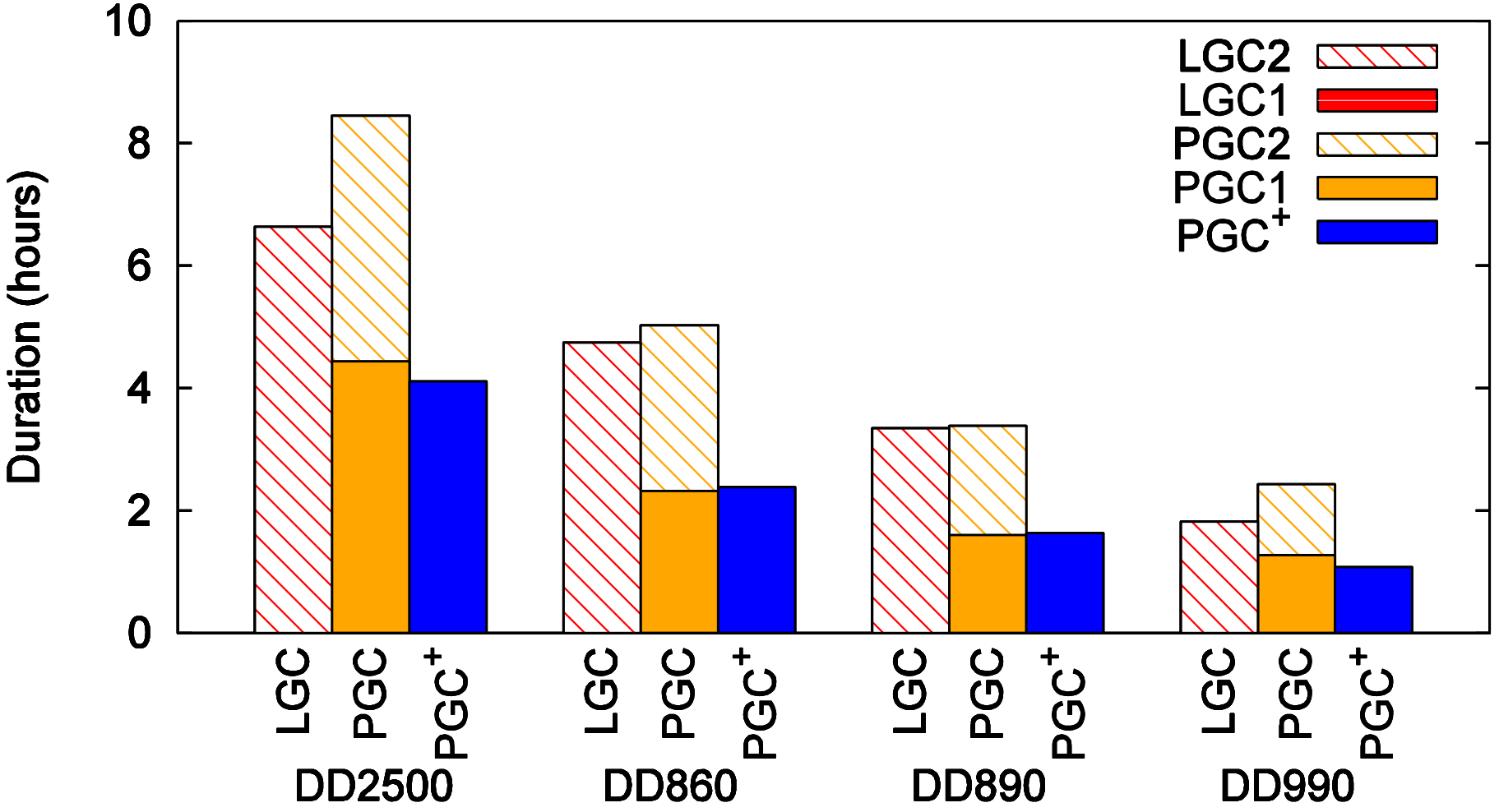
- For high TC workloads, PGC improved from LGC up to 20x
- For high file count workload, PGC improved over LGC by 7x
- 75% of systems upgraded from LGC to PGC suffered from some degradation but usually not much
  - Hard to compare LGC v/s PGC systems because of some other performance changes introduced with PGC

- Lab experiments to compare all GC variants with same performance parameters

© Copyright 2017 Dell Inc.

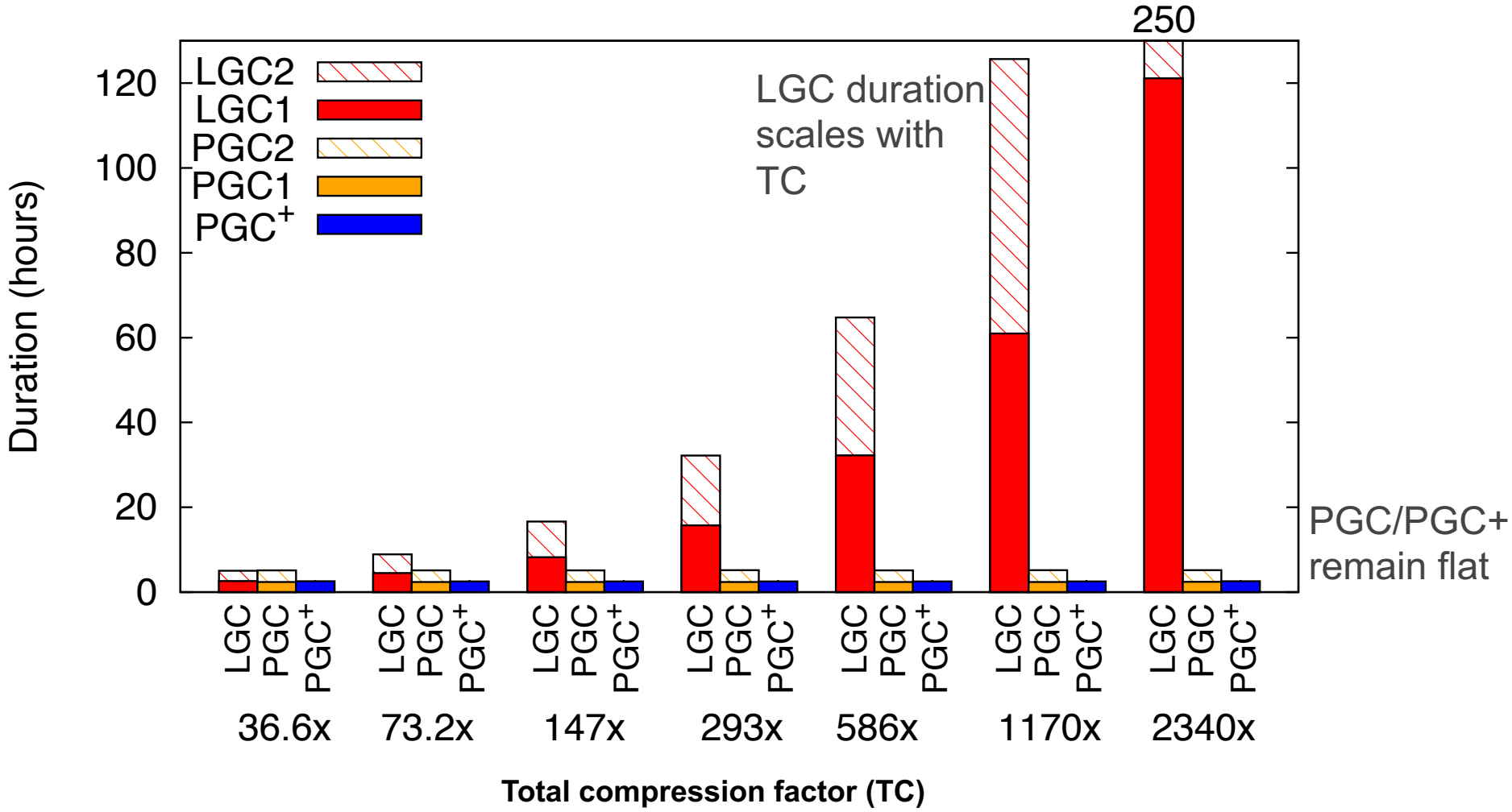
GC on Different Platforms (36.6x TC)

For this dedup, LGC2 is slightly better than PGC2
but PGC+ is better than LGC2/PGC2



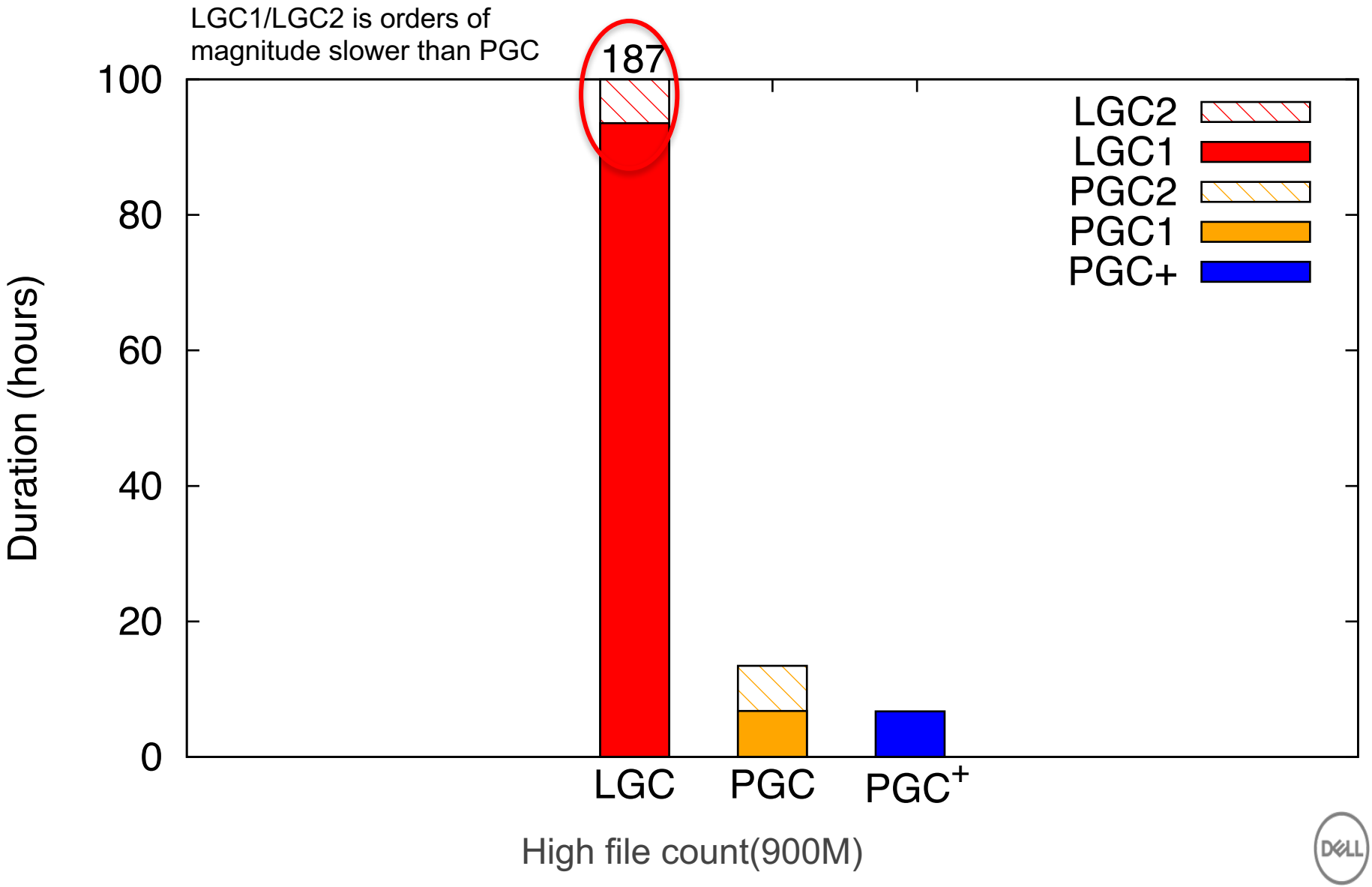
© Copyright 2017 Dell Inc.

High Total compression Workload



© Copyright 2017 Dell Inc.

High file Count Workload



© Copyright 2017 Dell Inc.

Conclusions

- Shift in workloads required moving from depth-first based mark phase to breadth-first based mark phase
- PGC works better than LGC for very high TC datasets and large number of small files
- Due to extra phases and performance constraints introduced in PGC, PGC is not uniformly faster than LGC
- PGC+ uses various optimizations to improve over PGC, primarily by avoiding multiple mark phases
- PGC+ is significantly faster than LGC when 2 mark phases are required and orders of magnitude faster for problematic workloads

© Copyright 2017 Dell Inc.

