Solving the dilemma of big data: live storage constraints versus full scientific reach

J. Gonzalez[1], J. Lauret[2], Y. Ying[3], G. Van Buren[2], M. Burtscher[4], Ph. Canal[5], I . A. Cali[3], R. Nunez[1]

1. Accelogic LLC, Weston, Florida, USA; 2. Brookhaven National Laboratory, Upton, New York, USA; 3. MIT, Cambridge, Massachusetts, USA;
4. Texas State University, San Marcos, Texas, USA; 5. Fermi National Laboratory, Batavia, Illinois, USA

KEY LIVE STORAGE ISSUES TODAY:

- **Exabyte-scale datasets coming with potentially excess recorded bits beyond the instruments' (or measured, or needed) precision**
- **Expense of performant, facility-scale live storage**
- **Limited capacity of personal (or mobile) computing storage for remote work**
- **Impossibility to optimize data compression choices for all uses/users**

accelogic TECH FOR ROOT TTREE:

- ***BLAST*** **: finely tunable lossy compression for real number types, and strong lossless compression for integer types**
- ***Precision Cascade*** **: multi-tiered storage of data precision levels**
  - Simple, branch-wise configuration for writing
  - Tiers can extend to full precision (residual) retention as a fallback
  - Automatic reading based on precision tier files' presence

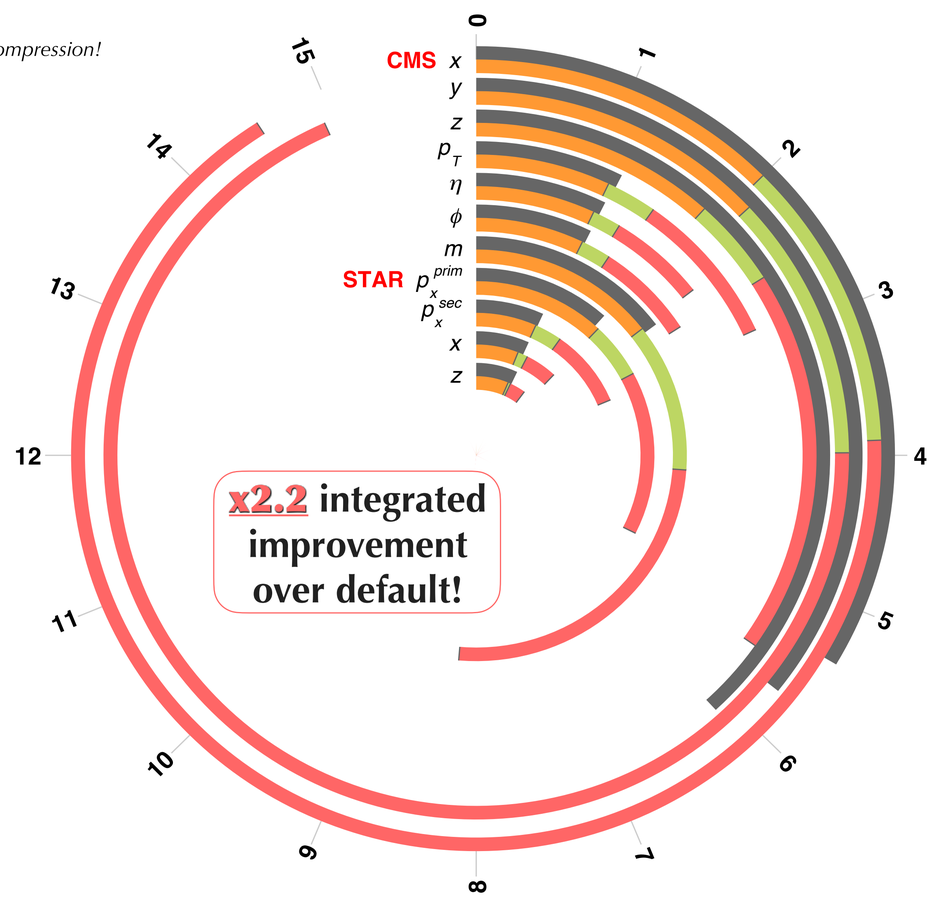
Findings for 11 data branches from **STAR** and **CMS** with:
- **default** (Zlib)
- BLAST + Precision Cascade in 3 tiers[1]:
  suggested compression[2] + **conservative** + **residual**

1. Shown compression speeds are BLAST alone except for residual.
*2. **STAR** and **CMS** test analyses see no science impact with suggested compression!*

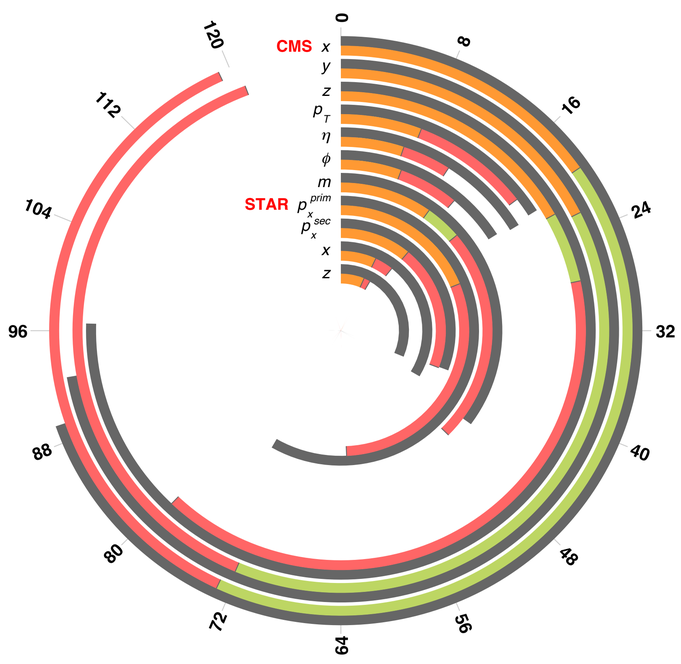
```
std::vector<Int_t> levels = { 51, 43 };
ROOT::PrecisionCascadeCompressionConfig targetConfig(
    ROOT::RCompressionSetting::EAlgorithm::kBLAST,
    levels,
    true /* Keep also the residual file */ );
...
lossy_branch->SetCompressionSettings(targetConfig);
```

This compression configuration is used for the shown results.

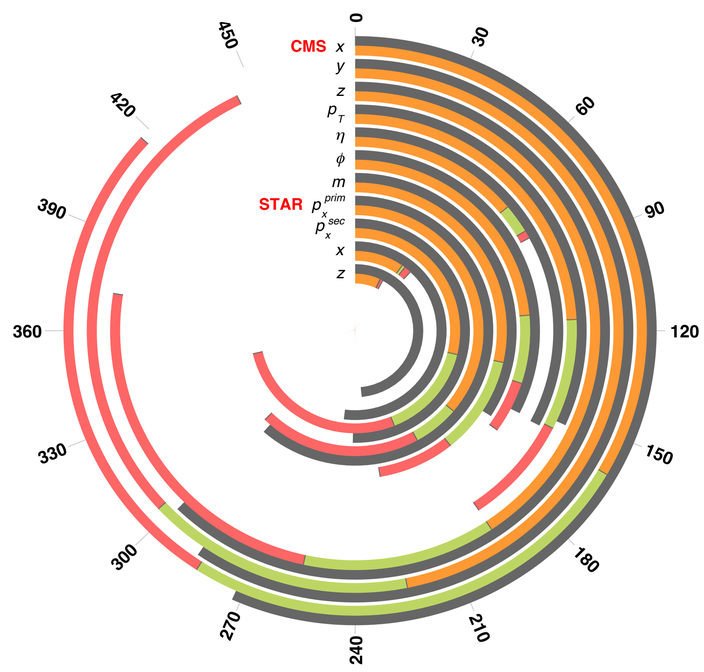
Compression Ratios



Compression Speeds [MB/s]



Decompression Speeds [MB/s]



PERFORMANCE SUMMARY ON REAL DATA:

- **Outperforms all tested existing encoders for integer types**
- **Better information retention at higher compression ratios than existing lossy types** *<Float16_t>*, *<Double32_t>*
- **Speeds depend on retained precision**
  - Currently slower than default (Zlib) for full precision
  - Compression can be competitive for BLAST alone
  - Decompression faster than default for precision that satisfies most analyses
- **Much better compression without science loss than any lossless encoders!**
  →This enables huge live storage savings (~x2) at no cost to science!!!

AVAILABILITY:

- **Accelogic LLC's BLAST encoder usage is being provided to the community under a modified BSD license**
- **Expecting public delivery in ROOT version 6.30/00 in just a few months!!!**

- Thanks to the **STAR** and **CMS** experiments for providing example data and analyses

Resources for more details & findings:
- **ACAT 2021:** Ph Canal et al 2023 *J. Phys.: Conf. Ser.* **2438** 012060
- **ACAT 2022:** Y Ying et al https://indico.cern.ch/event/1106990/contributions/4991262/