Hewlett Packard Enterprise

SEeSAW – Similarity Exploiting Storage for Accelerating Analytics Workflows

Kalapriya Kannan, **Suparna Bhattacharya**, Raj Kumar, Muthukumar Murugan, Doug Voigt

Hewlett Packard Enterprise

USENIX HotStorage, June 20, 2016

Digital data generation in an always-on world …

Got to see some great sand sculpting yesterday #sandsculptures #sandsculpting #beach #hamptonbeach #beachday

Beach day this weekend? Check out the **#HamptonBeach** sand sculptures

… inevitable when it is so easy to capture details

#HappyFathersDay to all the wonderful dads we know. Look at what's on **#HamptonBeach**! Enjoy this gorgeous day!

As a Dad, this is my favorite Sand Sculpture at the 2016 Hampton Beach Sand Sculpture Contest.

Informatics becomes the bottleneck as cost of gathering data goes down (mobile, social, IoT, surveillance, machine logs, …)

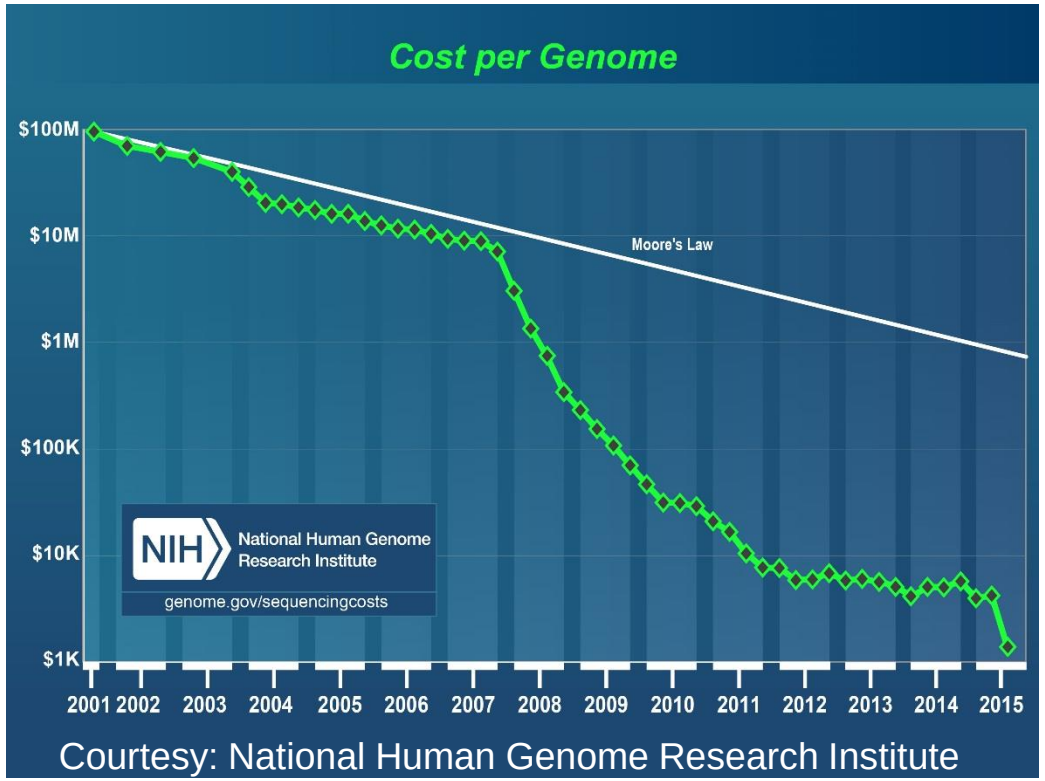
The Promise

Abundance of historical context to discover insights via analytics

The Dilemma

How to build capacity to absorb all information even if available?

Storage tends to be the long pole in analytics, *bypassed as far as possible!*

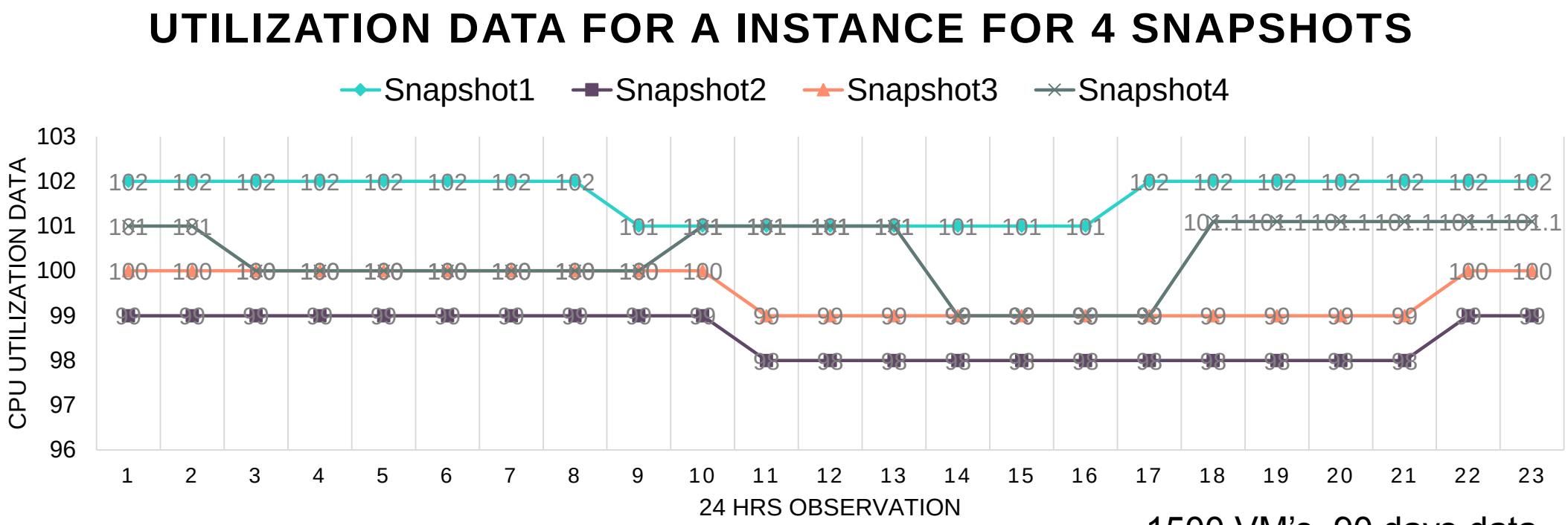


Courtesy: National Human Genome Research Institute

Instead, … could storage leverage characteristics of the data to address this dilemma?

Example: VM utilization data characteristics (Cloud Optimizer)

Historical utilization based forecasting (capacity planning, placement, migration)



1500 VM's, 90 days data
Snapshot = 3 days

Snapshots similar 98% of the time

Storage systems can accumulate much data with similar content

Especially true of data that is stored primarily for the purpose of analytics

Similar patterns arise naturally

Observations that show consistent repetitive variation

– Sensor, surveillance, geo-satellite data

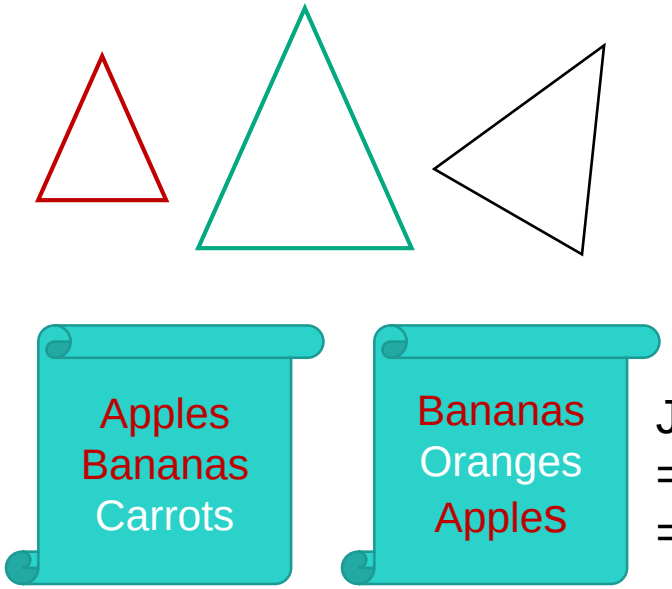
Data generated by similar sources in a similar context

– System utilization data for similar workloads

– Human generated info on same viewpoint

Results of similar computation on similar data

Often semantically or statistically similar but not duplicates …



Jaccard Similarity
= |X ∩ Y| / |X U Y|
= 0.5

… unlike typical data/compute de-dup

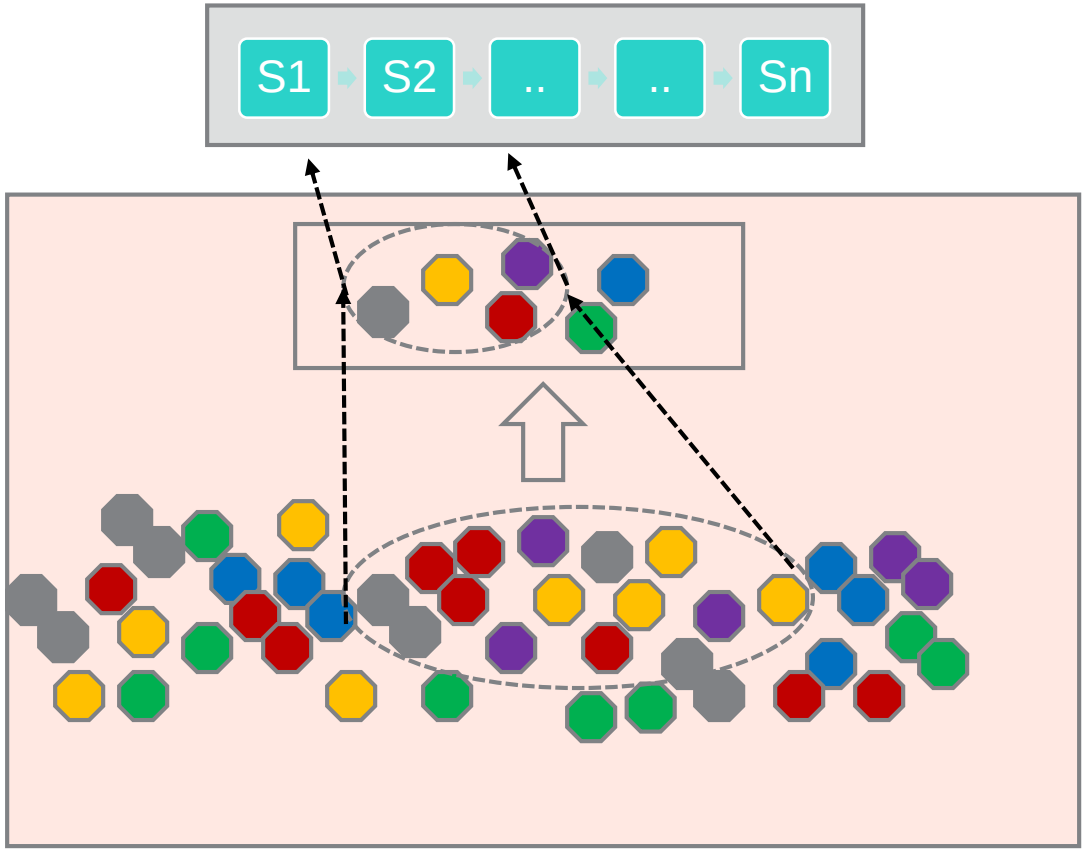
Should similarity detection be a fundamental storage primitive ?

Recognizing and exploiting this data property @ storage has several benefits

Advantages

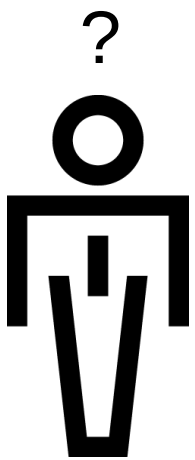
- **Accelerate analytics by skipping similar data sets**
- **Optimize storage by assigning differential value to data sets**
- **Prevent generation of nearly same data across workflows**
- **Analytical applications and frameworks operate without modification**
- **Reduce data movement, unlike similarity check at a higher layer**

Ex: Similarity based storage optimization



Provided we can get it right

Challenging Goals



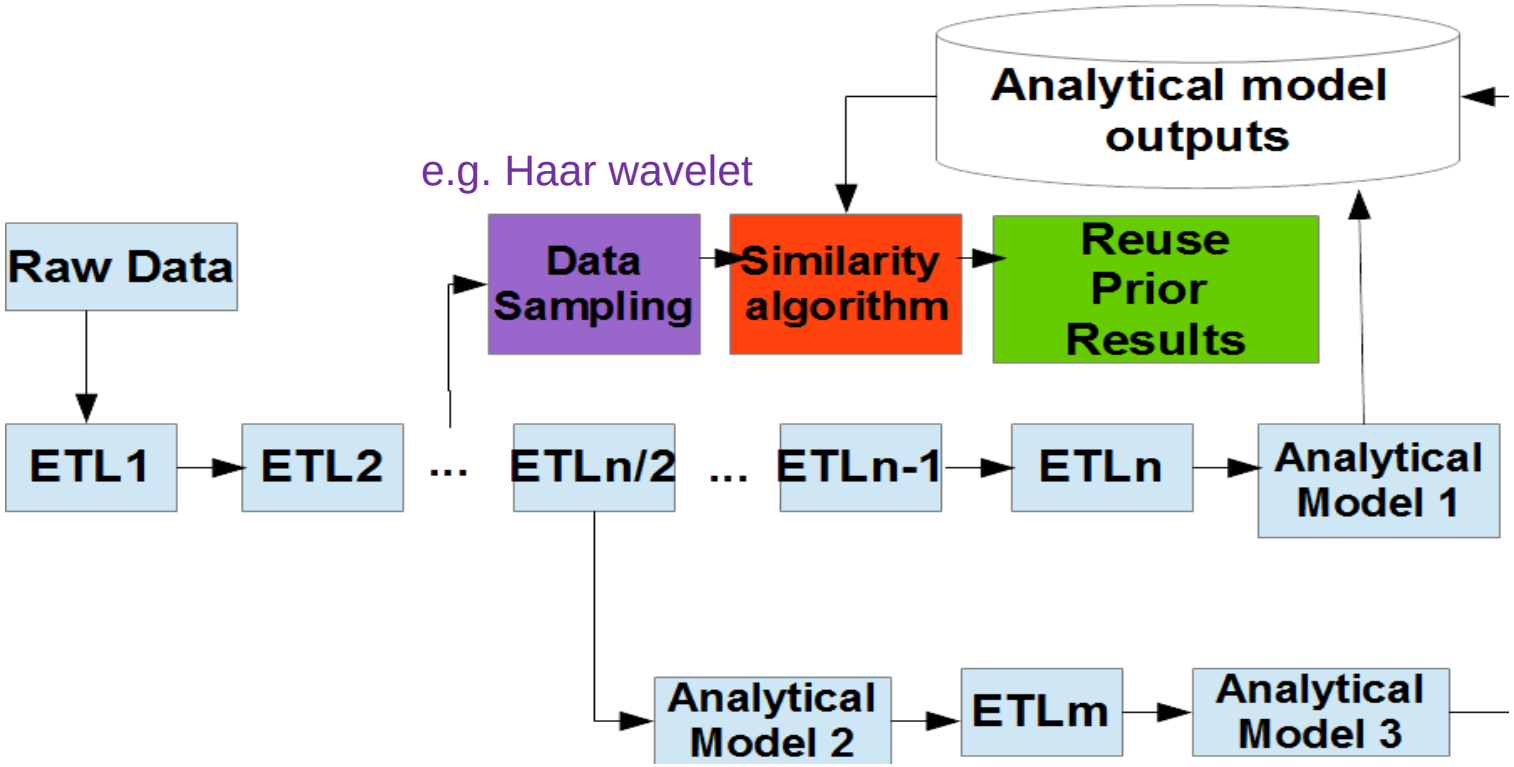
- **Similarity detection should not be heavy weight**
  - Rule out dissimilarity with minimal cost
  - State maintained with minimal storage overhead
- **Save resources IO, CPU, .. and time to get analytics results while preserving accuracy**
  - Similarity detection should reflect application relevant features
- **Seamlessly exploit similarity at the storage level**
  - Without application (or even framework) involvement

Therefore ….

SEeSAW – Similarity Exploiting Storage

The Idea

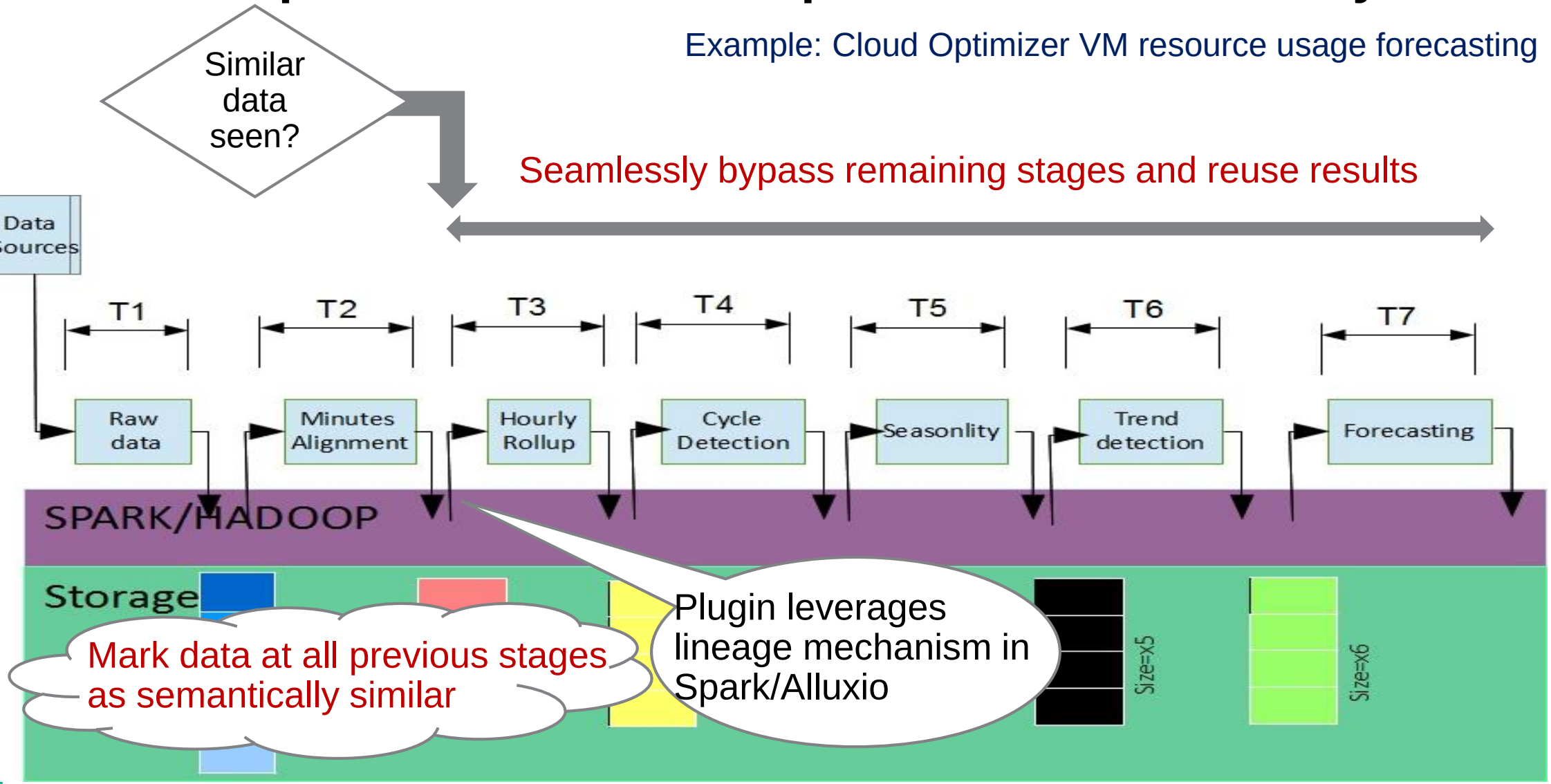
Identify similar data sets NOT JUST at the raw input but at intermediate transformation outputs (TO) to accelerate analytics workflows



Similarity at later stages reflects semantics of the model

How? Intercept transformation outputs to detect similarity

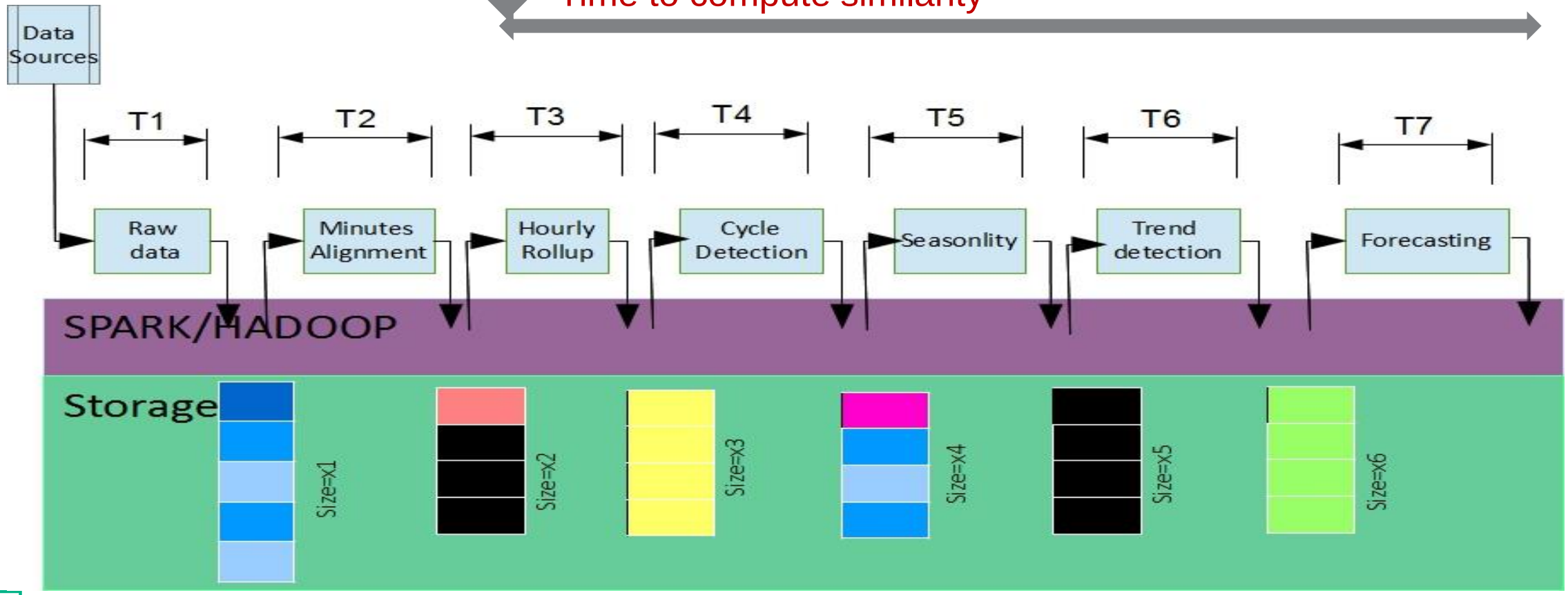
Example: Cloud Optimizer VM resource usage forecasting



Identify the Optimal stage and thresholds during first few runs

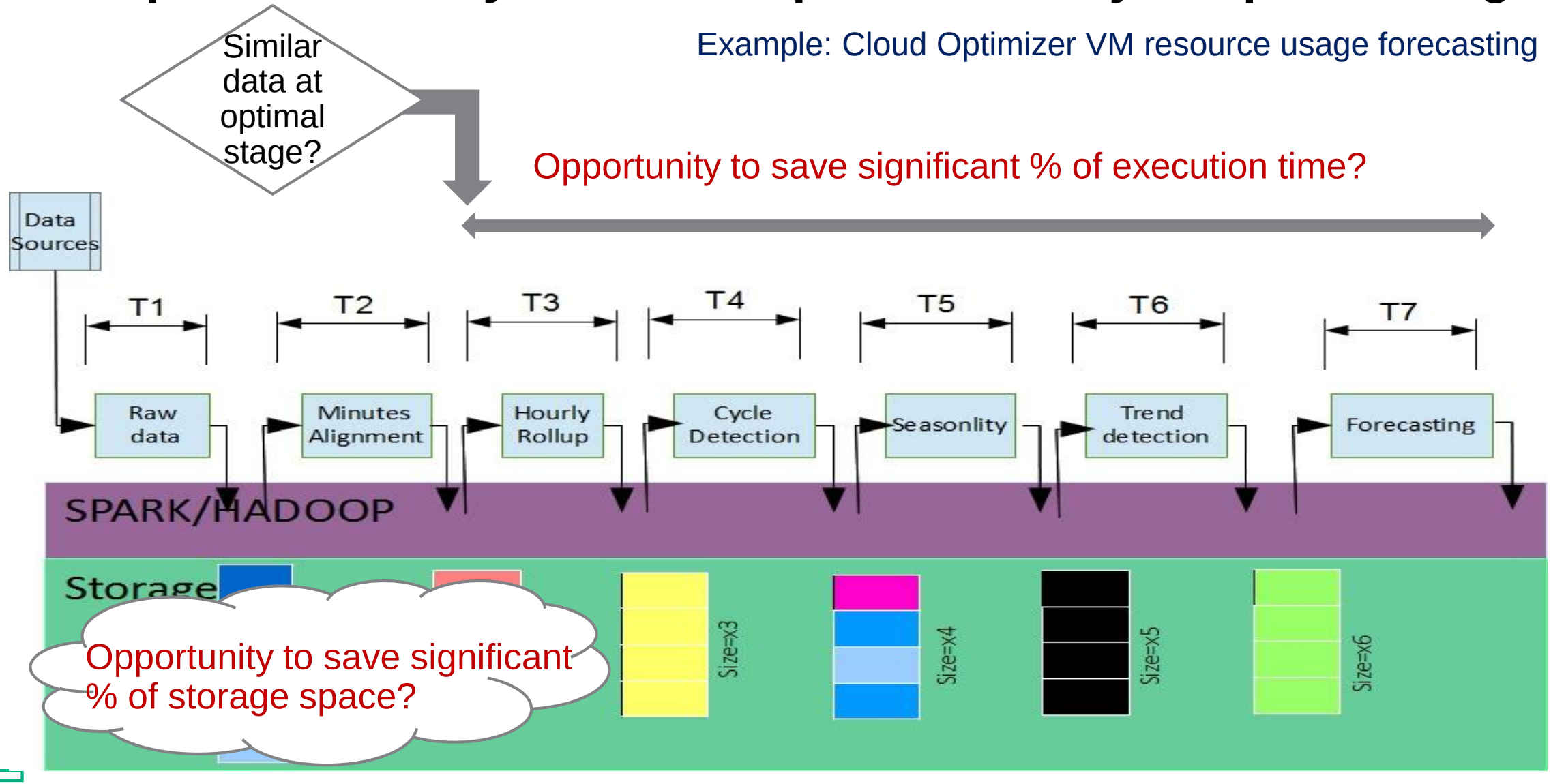
Pick optimal stage
to test for similarity

What is the time/space saved ?
vs
Time to compute similarity



Subsequent runs only test and exploit similarity at optimal stage

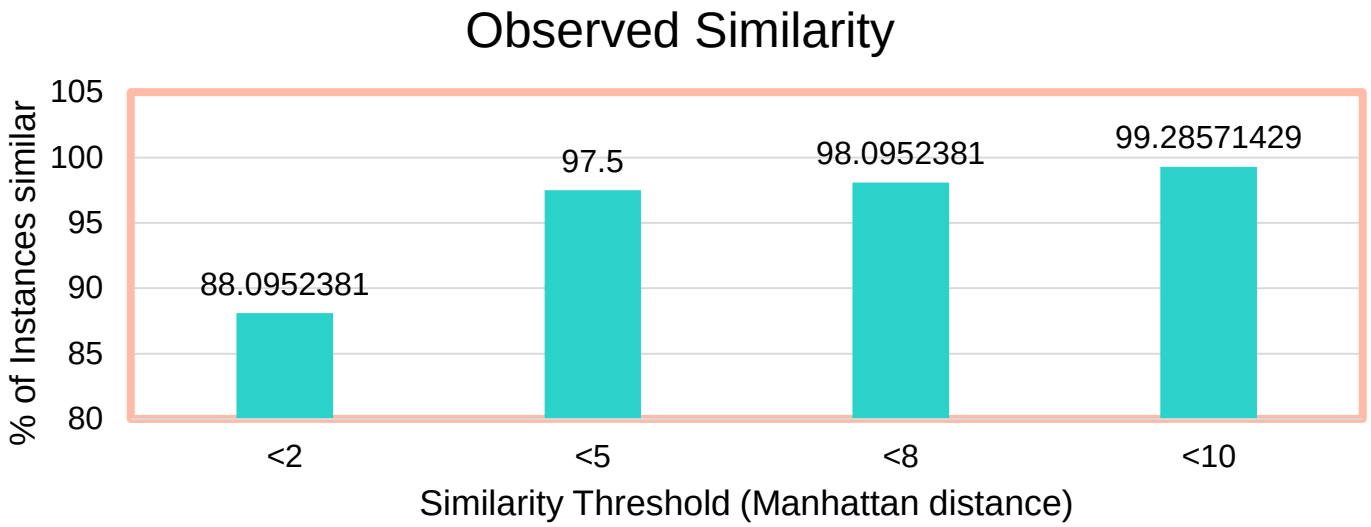
Example: Cloud Optimizer VM resource usage forecasting



Early Results

How much similarity is observed?

Similarity at the optimal stage across snapshots taken 3 days

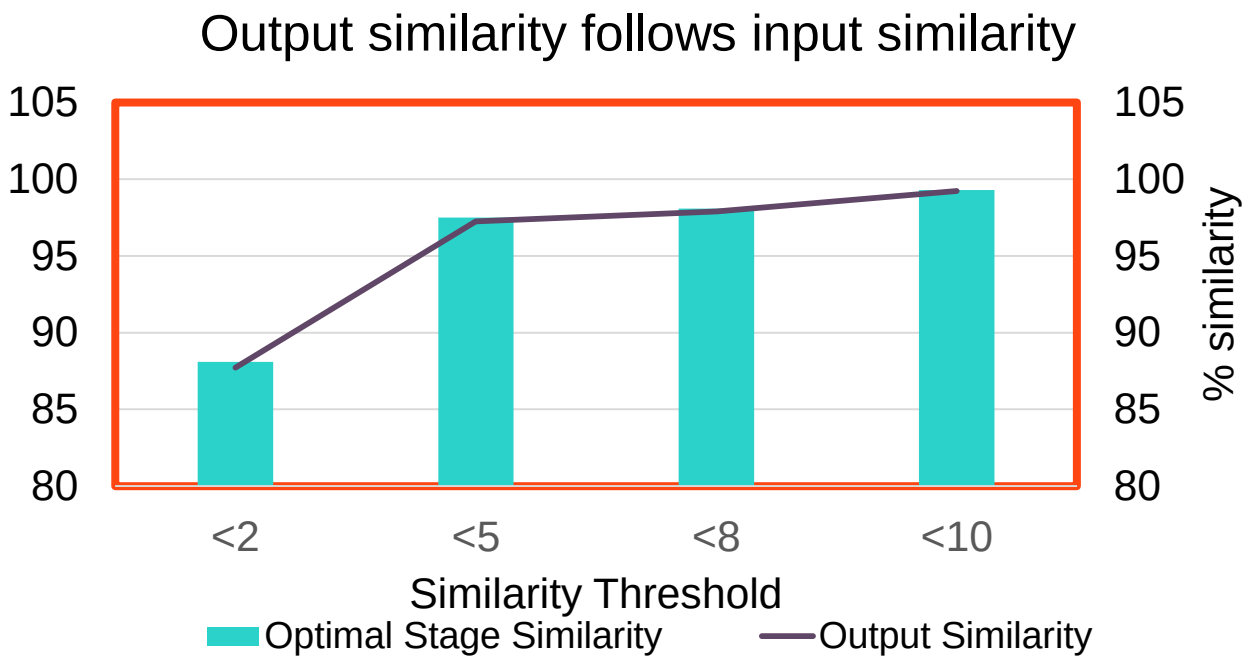


Snapshot 1 and 4

Similarity Threshold	<=5	<=8	<=10
Percentage Data set	97.5%	98%	99.28

Early Results

Do similar inputs at the optimal stage lead to similar analytics outputs?

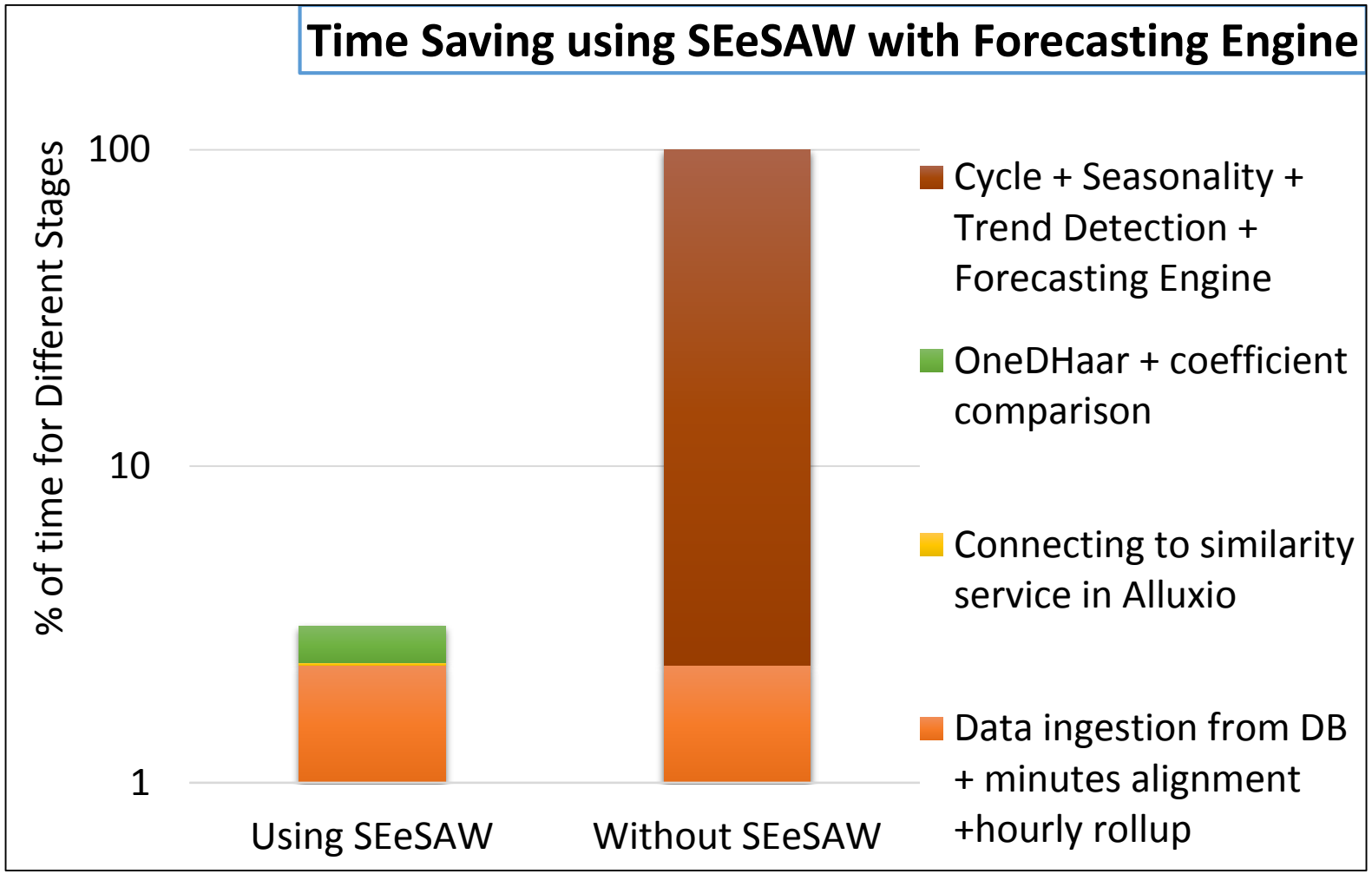


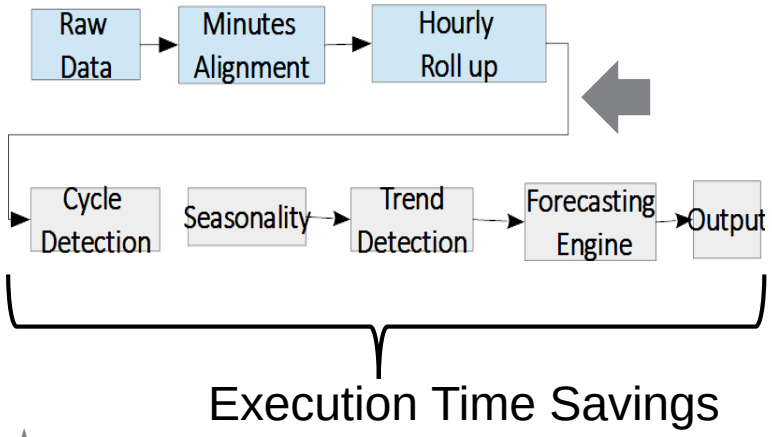
Snapshot 1-4

Variation < 0.5 %

Early Results

What was the performance benefit observed?





Speed-up 97%

Open challenges

Storage systems that exploit such data characteristics – worthwhile or too radical?

How should storage ensure ..

- Appropriate features and algorithms for measuring similarity?
- Similarity based decisions that preserve analytics accuracy?
- Enough history of prior executions is maintained for similarity detection and optimal stage analysis?

Are there other use cases for storage based similarity detection?

Are there other data characteristics storage systems could leverage?

Thank you

suparna.bhattacharya@hpe.com