A Metadata Workload Generator for Data-Intensive File Systems

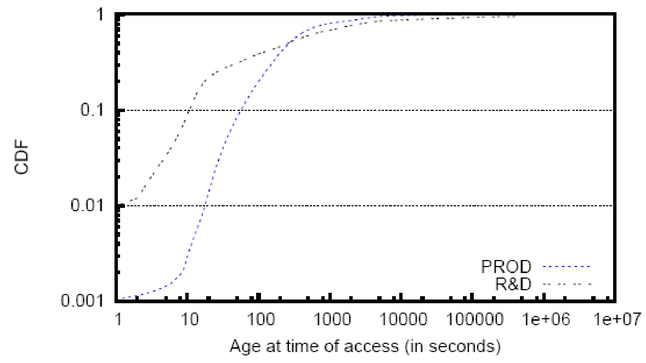
Cristina L. Abad*, Kihwal Lee^, Nathan Roberts^, Yi Lu+, Roy H. Campbell*

*** Department of Computer Science, *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign; ^ Yahoo!**

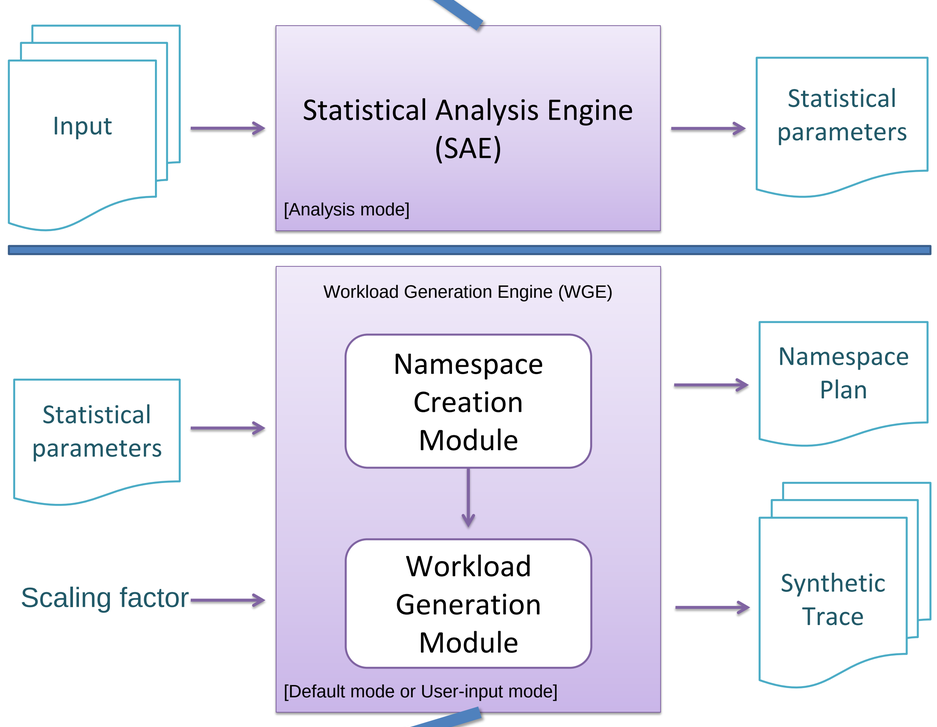
Statistical Analysis Engine (SAE)

- Analyzes namespace and metadata trace
- Extracts statistical parameters (model)
- Output format:
  - Parameters of distribution (if good fit)
  - Otherwise, empirical distribution percentiles
    - For continuous empirical distributions: Straight-line interpolation is used between the points (bins)

- Used to analyze to Yahoo! HDFS production traces from two clusters:
  - PROD → 4000+ nodes, production cluster
  - R&D → 1900+ nodes, research and development cluster
- Model of these two clusters included with tool, for default mode of operation



Example: File age at time of access

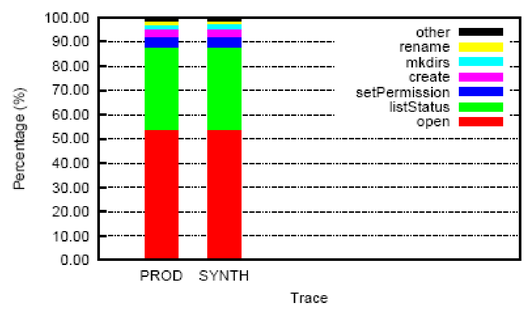


Namespace characterization	Number of directories
	Number of files
	Distribution of files at each depth in namespace hierarchy
	Distribution of directories at each depth in namespace
	Distribution of number of files per directory
	Distribution of number of subdirectories per directory
Access pattern characterization	Distribution of file sizes (in MB)
	Distribution of file age at time of access
	Distribution of file age at deletion (i.e., file life span)
Workload characterization	Percentiles of operation type in trace
	Interarrival rate distribution
	Percentiles of operations at each depth in namespace
	Distribution of files per depth in namespace
	Distribution of dirs. per depth in namespace
	Distribution of number of files per directory
	Distribution of number of subdirs. per dir.

Workload Generation Engine (WGE)

- Namespace creation module
  1. Create directory hierarchy; simultaneously satisfy:
     Distribution of directories at each and distribution of number of subdirectories per directory
  2. Create files
- Workload generation module: Discrete-event simulation
  1. Weighted selection of operation
  2. Choose target depth
  3. Choose a file, preserving temporal locality

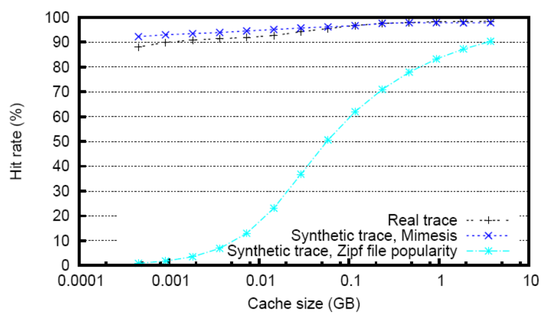
- **Validation:** Used WGE to generate synthetic trace modeled after PROD
- **Results:** WGE is highly accurate
  - Observed error < 5% for all statistical parameters extracted with SAE
- Example:



Types of metadata operations in trace

Case Study: HDFS Metadata Cache

- **Problem:** Namenode keeps all metadata in memory
- **Solution:** Use a metadata cache
  - We implemented an LRU metadata cache
  - Evaluation with our synthetic trace → Very close to real
  - Naïve evaluation with Zipf popularity synthetic trace → Poor



Summing it up:

- Massive data-intensive computing → new challenges for DFSs
- No standardized means of evaluation → reproduce? compare?
- Our approach: Mimesis, a synthetic workload generator
  - Produces representative and user-configurable traces on a realistic namespace
  - SAE: extracts parameters
  - WGE: generates statistically realistic traces
  - Parameters configurable to facilitate sensitivity analysis of designs

Future Work and Discussion

- Can we come up with a generative model that better explains namespace formation/evolution?
- Can we tune statistical parameters to create a suite of useful workloads?
- What are the limitations of the current scaling mechanism?
- Would it make sense to extend Mimesis to include a few micro-models that model specific applications (e.g., map-reduce)?
- Massively scalable trace replay is not trivial!

ILLINOIS YAHOO! AFRL AIR FORCE RESEARCH LABORATORY AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

cabad@illinois.edu