Similarity Measurement for Proxy Application Fidelity

Si Chen si.chen2@emory.edu
Omar Aaziz oaaziz@sandia.gov
Jeanine Cook jeacook@sandia.gov
Avani Wildani avani@mathcs.emory.edu

Emory College of Arts and Sciences
Sandia National Laboratories

Motivation and Goal

- Quantitatively understand the correspondence of proxy to parent applications with respect to their underlying node and memory hardware behavior.
- Ensure the system co-design and procurement that proxy application are high fidelity models of the target applications
- Find the most efficient similarity measurement.

Experiment

Applications

- ECP Applications in scientific domain of cosmology, seismic modeling, molecular dynamics, and thermal transport.
- 16 apps in proxy/parent pairs and 5 unpaired.

System platform

- **Intel Skylake**
- **IBM Power9**

MPI-only mode, 128 ranks, 1 rank/core, on four nodes.

Data collection

- Collection infrastructure: LDMS
- PAPI sampler collect 700 hardware events performance for each application.
- Run same problem for each application:
  - Use the same input problem and/or parameters.

Methodology

1. Application representation
   - A vector with 700 dimensions (features), each of which is an average rank value for one hardware event.
2. Feature set
   - Overall (700 features)
   - Subset (subgroup features):
     - Cache-related, e.g., longest_lat_cache.miss
     - Instruction mix
     - Pipeline
3. Distance measurement
   - Euclidean distance:
     - Cosine similarity

$$\cos\theta = \frac{\sum_{i=1}^{n} x_i y_i}{\|x\|\|y\|}$$

   - Statistic distance:
     - Kullback–Leibler divergence

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

     - Jensen-Shannon divergence

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

     - Wasserstein distance

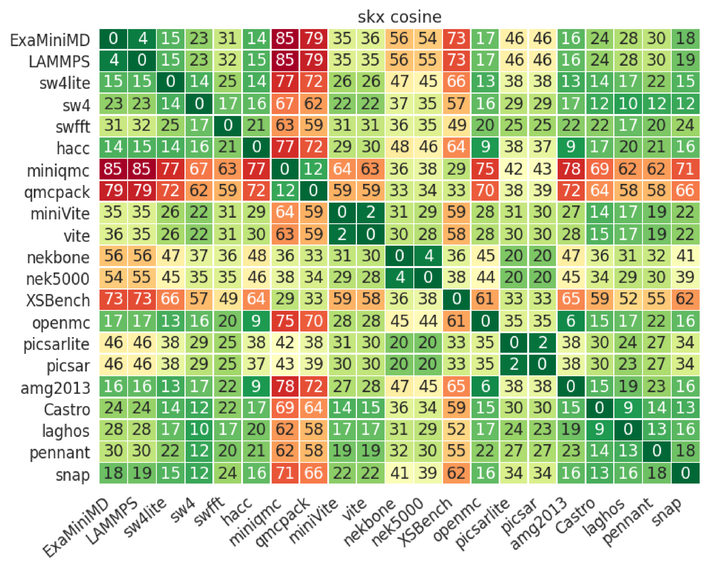
$$W_2(p,q) = \sqrt{\min_{P_{XY}} E_{P_{XY}}\left[\|x-y\|_2^2\right]} \text{ s.t. } P_X \sim p, P_Y \sim q$$

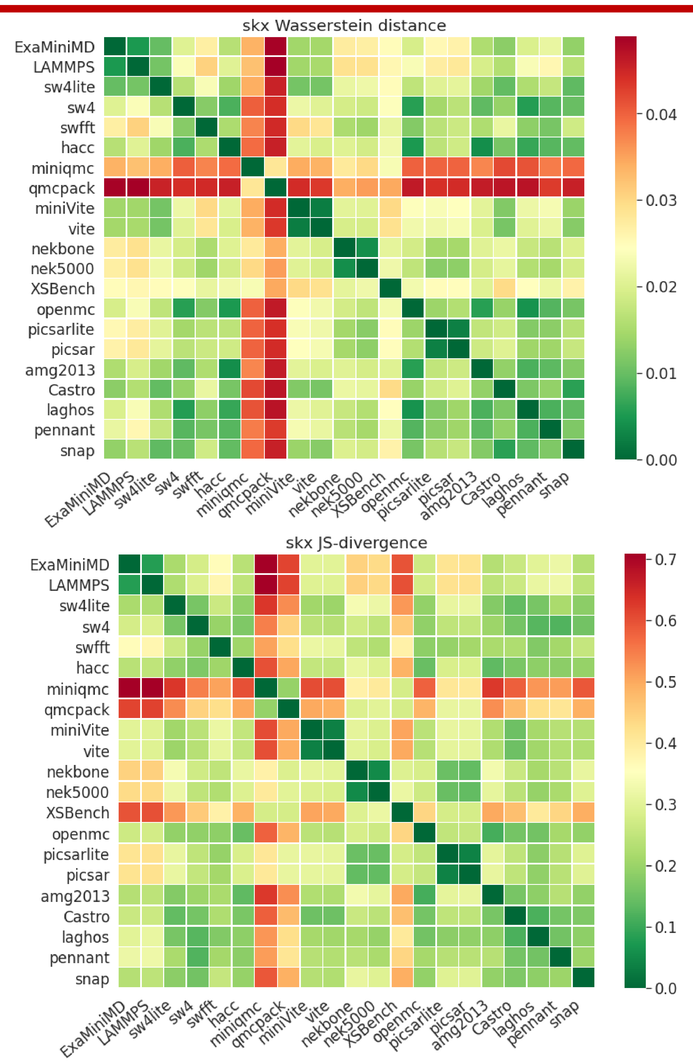
4. Cluster similar applications
   - Hierarchical Clustering
     - Get the similarity by tree branch height to rank applications.
5. Evaluation
   - Similarity ranking
   - Self similarity

ACKNOWLEDGEMENTS

Result

- Overall feature results of three distance methods in the Skylake system look alike.
- Three apps are generally different from the others.
- Three pairs are the most similar proxy/parent pairs.
- Two pairs (nek5000/nekbone and vite/miniVite) show high similar performance in cosine similarity, and relatively high similar performance in other two methods.
- The last five unpaired applications show relative similarity.
- QMCPack and MiniQMC are similar but differ from other applications.





Conclusion

- Overall ranking show similar result for three distance methods.
- Cross platform show different result because of different memory subsystem and SIMD width.
- Subset results: three subgroups contribute most to the overall similarity matrix.
- Most proxies are good representations of their parents, only with few divergences.
- Cosine similarity is simple, quick and interpretable by geometric angle
- Since other distance methods show similar result, cosine similarity is good enough to use for similarity measurement.

Future work

- Include domain expert evaluation for the choice of distance method.
- Calculate self-similarity for time series data with sliding windows to further validate the similarity methods.
- Validate similarity through measures of latency or some other environmental / real metric.