A Cosine Similarity Methodology to Characterize Proxy-Parent Application Correspondence

Jeffery A. Kuehn (LANL)
Jeanine Cook (SNL)
Omar Aaziz (SNL)
Courtney Vaughan (SNL)
Jonathan Cook (NMSU)
Sept 1, 2020
# Why Proxy Applications?

<table>
<thead>
<tr>
<th>Real Applications</th>
<th>Proxy Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 KLOC – 1 MLOC</td>
<td>1 KLOC – 100 KLOC</td>
</tr>
<tr>
<td>Large number of library dependencies</td>
<td>Minimize library dependencies</td>
</tr>
<tr>
<td>High Code Complexity</td>
<td>Simpler - Captures key kernels</td>
</tr>
<tr>
<td>May contain proprietary information</td>
<td>No protected IP</td>
</tr>
<tr>
<td>Licensing / Export / Classification</td>
<td>Freely distributable by design</td>
</tr>
<tr>
<td>Exactly what is/will be run</td>
<td>An approximation to the real app</td>
</tr>
<tr>
<td>Staff intensive to work with</td>
<td>Easier to work with</td>
</tr>
</tbody>
</table>
DOE Proxy Applications

- Currently a set of ~60 proxy applications
- ~12.5 M Lines-of-Code (LOC)
- Mixed languages (C/C++/Fortran/Python)
- Used for
  - System acquisition benchmarks
  - Exemplars for collaborative research contracts (e.g. PATHFORWARD)
  - System Testing
  - Figure-of-Merit for comparison exercises
- Issues
  - Poorly understood correspondence between proxies and real app
  - Proxy may represent only selected features of the real app
  - A lot of code – down-select is required for most exercises
  - Current down-select process is heavily subjective
Pick Your Proxies Carefully

• Imagine two vendors offering the following alternatives to your current system:
  • Offer #1: has twice the peak floating point performance, but is otherwise similar to your current system
  • Offer #2: has twice the memory bandwidth, but is otherwise similar to your current system

• The benchmarks you choose to compare these alternatives will determine which system your purchase.
  • Dense linear algebra will select Offer #1 because it has a higher peakFP, but you already knew that
  • Streaming benchmarks will select Offer #2 because it has a higher memBW, but you already knew that
  • The wrong choice of benchmark could cost you 2x in capability
    • Performance-metric space is two-dimensional (peakFP & memBW)

• Which is more similar to your WORKLOAD?
How to down-select in a more data-driven way?

• Codes: Large in number, huge in size, byzantine in complexity
• Limited resources skilled in performing the analysis
• Deep analysis and simulation efforts are both time and labor intensive
  • These will still be needed but they need to be focused, preferably on a smaller amount of code
• How to quickly determine which proxies are most similar to the workload?
  • Insight: Think of “performance” as the interaction between a workload and a particular device’s unique set of resource constraints
    • The manner and proportion to which those resource constraints are exercised by a particular workload becomes a “fingerprint” for that workload
    • It follows that workloads with similar fingerprints will respond similarly to small relaxations of the resource constraints
      • e.g. similarly memory bandwidth intensive codes will respond similarly to a memory bandwidth change (one-dimensional performance-metric space)
Approach

• **We rely on two elements as the building-blocks/tools:**
  • The ability to collect "fingerprint" for a code
  • The ability to quantify a similarity comparison of two “fingerprints”

• **Desirable features for the component metrics and comparison method:**
  • Should be related to hardware constraints (limitations / rooflines / bottlenecks)
  • Should be automatically forgiving of extraneous, redundant, or missing characteristics
  • Should be raw metrics and minimum analysis
    • Help to *focus* the analyst’s time rather than merely *consuming* it

• **Both capabilities are relatively easy to provide**
  • Construct fingerprint from aggregation of characteristic metrics (e.g. processed hardware counters)
  • Comparison: treat metrics as components of a vector in a high dimensionality space (10’s to 100’s of metrics) and compare the angle between these vectors
  • Can extend to include add’l counters, different hardware, different compilers, etc
What is Cosine Similarity?

• Term is taken from the ML community, but really just a property of dot (inner) product in vector spaces in 2 or more dimensions
  • Think: “Projection of \( \mathbf{x} \) in the direction of \( \mathbf{y} \)”

• From the two complementary definitions:
  • Algebraic: \( \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i \)
  • Geometric: \( \mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}|| ||\mathbf{y}|| \cos \theta \)
  • \( \cos \theta = (\sum_{i=1}^{n} x_i y_i) / (||\mathbf{x}|| ||\mathbf{y}||) \)

• The included angle tells us about:
  • Similarity in the direction of the vectors
    • Not their magnitudes
  • \( \cos(\theta) = 0.0 \) – orthogonal (never…)
    • 1st “quadrant” – non-negative components
  • \( \cos(\theta) = 1.0 \) – same
  • Non-unique/Non-orthogonal basis
Advantages of Cosine Similarity

- Mathematically "forgiving" of missing, extraneous, or redundant characteristics
  - Non-distinguishing components are naturally suppressed (outside the plane of $\theta$)
  - Avoids the need for a “perfect” principle component analysis.
  - A wide net can be cast, capturing a broad variety of components without fear of corrupting the results
- Easily Extended to:
  - Different Kinds of Metrics (Time, memory, calls, samples, etc)
  - Multiple processors (mpi or openmp)
  - Different hardware
  - Different software stacks
  - Large numbers of component metrics
  - Assess similarity across different configurations of an application or proxy
Identify how proxies or apps cluster and which proxies best represent sets of apps

<table>
<thead>
<tr>
<th></th>
<th>Average App1&amp; App2</th>
<th>App1</th>
<th>App2</th>
<th>Proxy10</th>
<th>Proxy04</th>
<th>Proxy05</th>
<th>Proxy08</th>
<th>Proxy11</th>
<th>Proxy01</th>
<th>Proxy02</th>
<th>Proxy07</th>
<th>Proxy09</th>
<th>Proxy03</th>
<th>Proxy06</th>
<th>Proxy12</th>
</tr>
</thead>
<tbody>
<tr>
<td>App1</td>
<td>0.97</td>
<td>1.00</td>
<td>0.94</td>
<td>0.98</td>
<td>0.95</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
<td>0.94</td>
<td>0.91</td>
<td>0.78</td>
<td>0.72</td>
<td>0.67</td>
</tr>
<tr>
<td>App2</td>
<td>0.97</td>
<td>0.94</td>
<td>1.00</td>
<td>0.88</td>
<td>0.89</td>
<td>0.85</td>
<td>0.85</td>
<td>0.84</td>
<td>0.84</td>
<td>0.88</td>
<td>0.82</td>
<td>0.78</td>
<td>0.81</td>
<td>0.53</td>
<td>0.48</td>
</tr>
<tr>
<td>Proxy10</td>
<td>0.93</td>
<td>0.98</td>
<td>0.88</td>
<td>1.00</td>
<td>0.98</td>
<td>0.96</td>
<td>0.94</td>
<td>0.95</td>
<td>0.94</td>
<td>0.91</td>
<td>0.93</td>
<td>0.90</td>
<td>0.73</td>
<td>0.72</td>
<td>0.68</td>
</tr>
<tr>
<td>Proxy04</td>
<td>0.92</td>
<td>0.95</td>
<td>0.89</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.96</td>
<td>0.83</td>
<td>0.80</td>
<td>0.76</td>
<td>0.58</td>
<td>0.51</td>
<td></td>
</tr>
<tr>
<td>Proxy05</td>
<td>0.89</td>
<td>0.92</td>
<td>0.85</td>
<td>0.96</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>0.80</td>
<td>0.76</td>
<td>0.72</td>
<td>0.55</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td>Proxy08</td>
<td>0.88</td>
<td>0.91</td>
<td>0.85</td>
<td>0.94</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>0.77</td>
<td>0.74</td>
<td>0.73</td>
<td>0.50</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td>Proxy11</td>
<td>0.88</td>
<td>0.91</td>
<td>0.84</td>
<td>0.95</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>0.78</td>
<td>0.75</td>
<td>0.72</td>
<td>0.53</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>Proxy01</td>
<td>0.87</td>
<td>0.89</td>
<td>0.84</td>
<td>0.94</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>0.76</td>
<td>0.72</td>
<td>0.72</td>
<td>0.49</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>Proxy02</td>
<td>0.89</td>
<td>0.90</td>
<td>0.88</td>
<td>0.91</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.95</td>
<td>0.96</td>
<td>1.00</td>
<td>0.73</td>
<td>0.69</td>
<td>0.88</td>
<td>0.42</td>
<td>0.35</td>
</tr>
<tr>
<td>Proxy07</td>
<td>0.88</td>
<td>0.94</td>
<td>0.82</td>
<td>0.93</td>
<td>0.83</td>
<td>0.80</td>
<td>0.77</td>
<td>0.78</td>
<td>0.76</td>
<td>0.73</td>
<td>1.00</td>
<td>0.99</td>
<td>0.62</td>
<td>0.90</td>
<td>0.86</td>
</tr>
<tr>
<td>Proxy09</td>
<td>0.85</td>
<td>0.91</td>
<td>0.78</td>
<td>0.90</td>
<td>0.80</td>
<td>0.76</td>
<td>0.74</td>
<td>0.75</td>
<td>0.72</td>
<td>0.69</td>
<td>0.99</td>
<td>1.00</td>
<td>0.59</td>
<td>0.92</td>
<td>0.89</td>
</tr>
<tr>
<td>Proxy03</td>
<td>0.80</td>
<td>0.78</td>
<td>0.81</td>
<td>0.73</td>
<td>0.76</td>
<td>0.72</td>
<td>0.73</td>
<td>0.72</td>
<td>0.72</td>
<td>0.88</td>
<td>0.62</td>
<td>0.59</td>
<td>1.00</td>
<td>0.33</td>
<td>0.28</td>
</tr>
<tr>
<td>Proxy06</td>
<td>0.63</td>
<td>0.72</td>
<td>0.53</td>
<td>0.72</td>
<td>0.58</td>
<td>0.55</td>
<td>0.50</td>
<td>0.53</td>
<td>0.49</td>
<td>0.42</td>
<td>0.90</td>
<td>0.92</td>
<td>0.33</td>
<td>1.00</td>
<td>0.96</td>
</tr>
<tr>
<td>Proxy12</td>
<td>0.57</td>
<td>0.67</td>
<td>0.48</td>
<td>0.68</td>
<td>0.51</td>
<td>0.47</td>
<td>0.43</td>
<td>0.46</td>
<td>0.41</td>
<td>0.35</td>
<td>0.86</td>
<td>0.89</td>
<td>0.28</td>
<td>0.96</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Clustering the Results:

- None of the proxies exhibit high similarity to App2
  - **REPRESENTATION GAP**
- But several of the proxies exhibit high similarity to each other
  - 4, 5, 8, 11, 1 (&10, 2)
  - 7, 9
  - **REDUNDANCY**
- For these Apps in this basis set
  - 1 (2?) proxy was “useful”
  - 4-6 proxies, not 12.
- More isn’t better
  - **Significant down-select possible**
- For reference:
  - \( \arccos 0.28 \approx 74^\circ \)
  - \( \arccos 0.90 \approx 25^\circ \)
  - \( \arccos 0.98 \approx 12^\circ \)
  - \( \arccos 0.99 \approx 8^\circ \)
## Proxy – Parent Correspondence

<table>
<thead>
<tr>
<th></th>
<th>ExaMINI-MD</th>
<th>LAMMPS</th>
<th>MINIQMC</th>
<th>QMCPACK</th>
<th>SW4LITE</th>
<th>SW4</th>
<th>SWFFT</th>
<th>HACC</th>
<th>PENNANT</th>
<th>SNAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExaMINI-MD</td>
<td>0.00</td>
<td>0.29</td>
<td>58.53</td>
<td>20.78</td>
<td>8.33</td>
<td>8.19</td>
<td>6.30</td>
<td>7.66</td>
<td>8.39</td>
<td>3.67</td>
</tr>
<tr>
<td>LAMMPS</td>
<td>0.29</td>
<td>0.00</td>
<td>58.51</td>
<td>20.76</td>
<td>8.36</td>
<td>8.23</td>
<td>6.14</td>
<td>7.52</td>
<td>8.38</td>
<td>3.59</td>
</tr>
<tr>
<td>MINIQMC</td>
<td>58.53</td>
<td>58.51</td>
<td>0.00</td>
<td>39.30</td>
<td>51.41</td>
<td>51.60</td>
<td>56.63</td>
<td>54.44</td>
<td>51.33</td>
<td>55.20</td>
</tr>
<tr>
<td>QMCPACK</td>
<td>20.78</td>
<td>20.76</td>
<td>39.30</td>
<td>0.00</td>
<td>15.18</td>
<td>15.33</td>
<td>19.19</td>
<td>17.91</td>
<td>14.56</td>
<td>17.76</td>
</tr>
<tr>
<td>SW4LITE</td>
<td>8.33</td>
<td>8.36</td>
<td>51.41</td>
<td>15.18</td>
<td>0.00</td>
<td>0.39</td>
<td>10.47</td>
<td>9.66</td>
<td>3.89</td>
<td>6.04</td>
</tr>
<tr>
<td>SW4</td>
<td>8.19</td>
<td>8.23</td>
<td>51.60</td>
<td>15.33</td>
<td>0.39</td>
<td>0.00</td>
<td>10.52</td>
<td>9.78</td>
<td>3.76</td>
<td>6.00</td>
</tr>
<tr>
<td>SWFFT</td>
<td>6.30</td>
<td>6.14</td>
<td>56.63</td>
<td>19.19</td>
<td>10.47</td>
<td>10.52</td>
<td>0.00</td>
<td>3.14</td>
<td>10.42</td>
<td>5.18</td>
</tr>
<tr>
<td>HACC</td>
<td>7.66</td>
<td>7.52</td>
<td>54.44</td>
<td>17.91</td>
<td>9.66</td>
<td>9.78</td>
<td>3.14</td>
<td>9.82</td>
<td>9.82</td>
<td>5.26</td>
</tr>
<tr>
<td>PENNANT</td>
<td>8.39</td>
<td>8.39</td>
<td>51.33</td>
<td>14.56</td>
<td>3.89</td>
<td>3.76</td>
<td>10.42</td>
<td>9.82</td>
<td>0.00</td>
<td>6.01</td>
</tr>
<tr>
<td>SNAP</td>
<td>3.67</td>
<td>3.59</td>
<td>55.20</td>
<td>17.76</td>
<td>6.04</td>
<td>6.00</td>
<td>5.18</td>
<td>5.26</td>
<td>6.01</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- Most proxies of high fidelity representation of parent app
  - Exception: MiniQMC is weaker overall, particularly on cache behavior
- Exception: SWFFT is weaker overall, but represents cache behavior
- Gap: representation of HACC & QMCPack is weak, particularly QMCPack’s cache behavior
- Redundancy: All proxies have similar cache behavior to parent EXCEPT MiniQMC/QMCPACK
- Working set: MD and QMC are fairly different from the rest of the tested DOE apps/miniapps
  - Be careful using MD/QMC to characterize a system for other apps
Impacts

• Identify Redundancies in the Proxy App Suite / Benchmark Suites
  • ➔ Lower software maintenance burden
  • ➔ Easier to work with
  • ➔ Better understanding of what aspect(s) of the parent workload the proxy represents

• Deliver proxies that are more representative of real app behaviors to vendors
  • ➔ systems better optimized for our apps

• Define minimal proxy suite that covers all parent behavior
  • ➔ faster design-space exploration

• Can be broadly used to understand performance differences in compiler and application optimizations, application inputs/problems, kernels and systems, etc