Community-wide benchmarking effort

High level remarks

- There are two types of benchmarks:
  - App like benchmarks that give estimates on app performance
  - Benchmarks that measure hardware feeds and speeds
- Benchmarks serve different goals
  - Measure speeds
  - Measure sizes
  - Probe behaviors (e.g. counters)

Requirements, Needs, Desires

- Approach should allow for both bench types and different use of the results.
- Benchmark runs should be reproducible.
- Benchmarks should be accessible through some public dashboard.
- Private copies of dashboard should be possible, in order to run on NDA hardware.
- Results (from non-NDA runs) should be able to be stored and be visible to the world (URL) so they can be referred.

Implications (1)

- Benchmark runs should be reproducible
  - Environment, settings, context should be captured and stored
  - Formal way to call the benchmarks and structured output so that harnesses can be written that automate the execution process
  - Ability to provide custom runners by benchmark author to allow for custom behavior (e.g., use MPI, disable cache prefetchers, etc)
  - Formal specification of requirements to build and run benchmark (e.g., requires MPI, requires PAPI, disable prefetchers)
  - CI integration

Implications (2)

- Results (from non-NDA runs) should be able to be stored and be visible to the world (URL) so they can be referred.
  - Benchmark output should be formally specified.
  - Output Visualization/Utilization left open-ended for future discussions.
  - There is interest for checking output correctness (instead of/in addition to performance) for obtaining metrics such as “goodput” as opposed to throughput.

Next Steps

- Todd will initiate a dashboard/CI/output storage as sub-part of an LLNL project.
- Reach out to people not present here, plan some future (virtual) meeting among those interested.
- Benchmark writers will start contributing benchmarks.
  - Maybe benchmarks need to be re-structured
- Iterate.