HEP Benchmarks @ Nikhef

- Accounting
  - Standardised CPU usage for accounting records
  - External constraint
- Purchasing
  - Evaluating what we have purchased
  - Other peoples benchmarks inform future purchases
- Software
  - Comparing compilers
- Task Force measurments
  - Provide our benchmarks to others
  - View other peoples benchmarks

Nikhef

Batch System

- **Heterogeneous batch sysem**
  - **Regular, atleast yearly, purchases of new hardware**
  - **233 nodes, 10792 cores:**
    - **44 x AMD Epic 7702P**
    - **32 x AMD Epic 7H12**
    - **58 x Intel Xeon E5-2650 v4**
    - **15 x Inet Xeon Gold 6148**
    - **66 x AMD Epic 7751P**
    - **18 x IntelXeon E5-2680 v3**

Nikhef

Initial Usage

- **Sept 2021**
- **Running HEP-spec 2017 with various gcc versions**
  - **Confusion: run HEP-spec directly or via HEP-Benchmark-Suite**
  - **Used HEP-spec: only way (at time) to use different gcc versions**
- **None standard running was difficult**
  - **Not surprising, we tried to run before walking**

Nikhef

Task Force Measurements

- **Much easier**
  - **A little trial and error to get started**
  - **Supplied scripts were simple to customise and run**
  - **Easily repeatable for the different scripts**
  - **Work flow:**
    - **run all repetitions for a script with publish false**
    - **then upload afterwards**
- **Ran the HS06 benchmarks – straight forward**
  - **Create the HS06 tarball**
  - **Copy to benchmark machine**
  - **Run benchmark script**

Nikhef

Central DB

- **Access issues**
  - **Took a long time to resolve**
- **Primary use for us: track completed work flow**
- **Personal bias:**
  - **Kibana is often confusing**
  - **Find data is not the problem**
  - **Filtering down to what's needed can be difficult**

Nikhef