Benchmarking Grid Environments at the User Level

Rob F. Van der Wijngaart (Computer Sciences Corp.)

Michael Frumkin (NAS Division)

NASA Ames Research Center
Moffett Field, CA, USA

Philip Dormer Stanhope, Earl of Chesterfield (1746):
A job worth doing is worth doing well

Nerd's addendum:
A distributed job worth doing is worth doing well

Nerd's addendum's addendum:
A distributed job worth doing is worth measuring

Motivation

Current measurement technology for Grid performance:

- Simulation tools: WARMstones, Bricks, MicroGrid, …
- Probes: Globus HBM, NWS, NetLogger, …

Useful for:

- Users debugging Grid application performance
- Developers of Grid & communication software

But …

- Does not provide metric for comparisons of Grid performance delivered on actual distributed applications

Motivation cont'd

Considerations:

- Simulations: Idealized; Grid model assumptions unclear; require voluminous input; static;
- Probes: Superposition principle doesn't hold, in general;

Goal: Determine Grid *application performance* objectively

Definition: Performance =

$$\frac{\#\ \text{work units completed}}{\text{time} \times \text{Grid resources used}}$$

Motivation cont'd

Work unit:

- single unit: requires overall workload characterization
- use representative set of distributed apps

Time:

- wall clock
- cpu

Unit of Grid resource:

- how to sum heterogeneous resource?
- G$? US$?
- Stopgap: add resources to performance *narrative*

Grid Benchmarking Proposal

- Provide paper-and-pencil specifications of small set of complete, “representative” distributed apps
- For convenience, also provide reference implementations (Globus, Condor, Legion, Java, Grid Engine, ksh, ...)
- Focus on computational aspects of Grids
  - Use mesh-based NAS Parallel Benchmarks (NPB) as building blocks: well understood, well-calibrated, allow communication, deterministic, no input required, portable, parallel

NAS Parallel Benchmark codes

Kernels:

- EP Random-number generator
- IS Integer sort
- CG Conjugate gradient
- MG post-processing (data smoother)
- FT visualization (spectral analysis)

Pseudo apps:

- BT
- SP
- LU

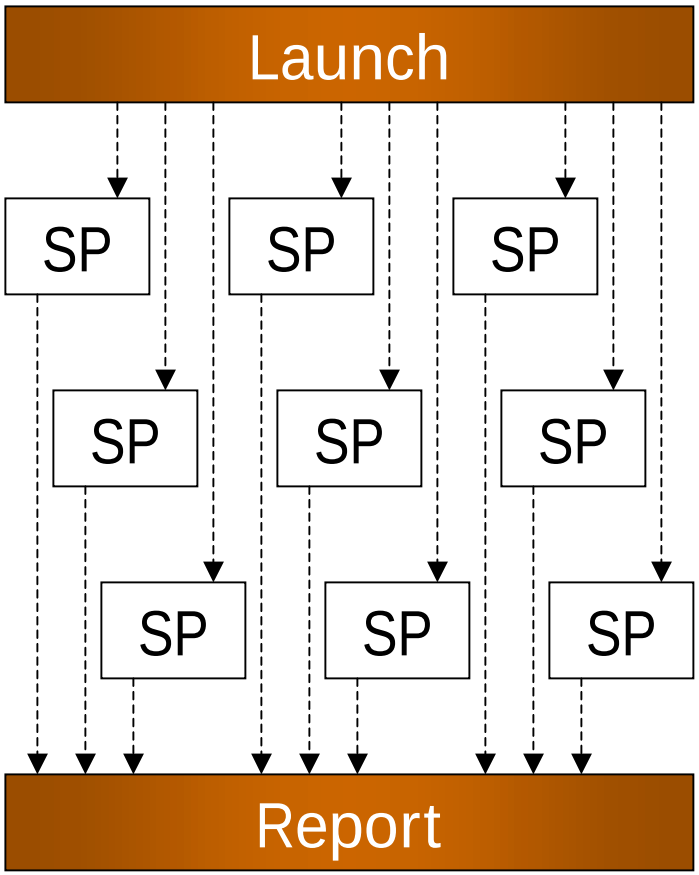
} scientific computations (flow solvers)

Grid benchmark construction

- Provide synthetic Grid app. for scientific computing:
  - Data Flow Graph (DFG) coupling scientific codes
- Specify:
  - abstract services: authenticate, create task, communicate
  - problem sizes (classes): S, A, B, C, …
- Do not specify:
  - Mapping, scheduling, fault tolerance, data security
- Measure and report turnaround time

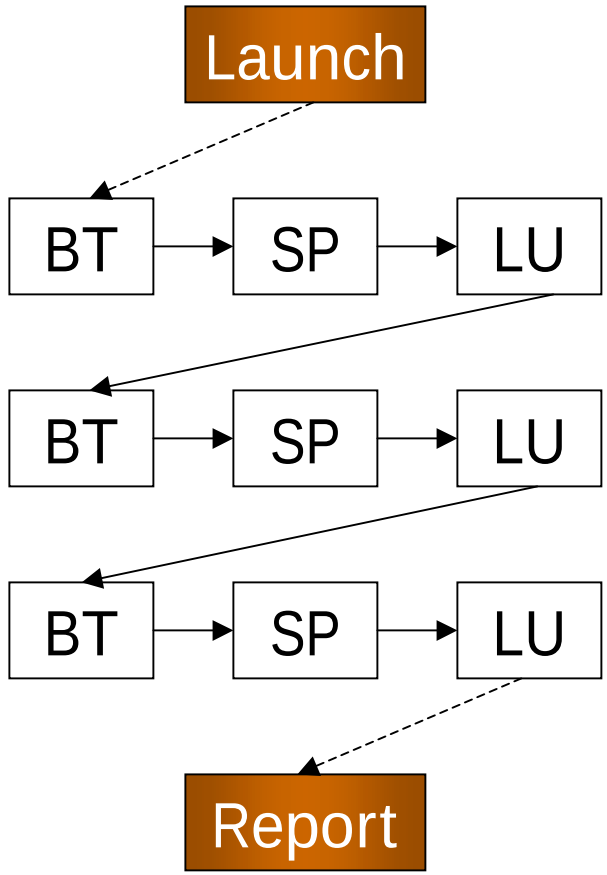
NGB Data Flow Graphs, Sample size

Embarrassingly Distributed (ED)



Parameter study

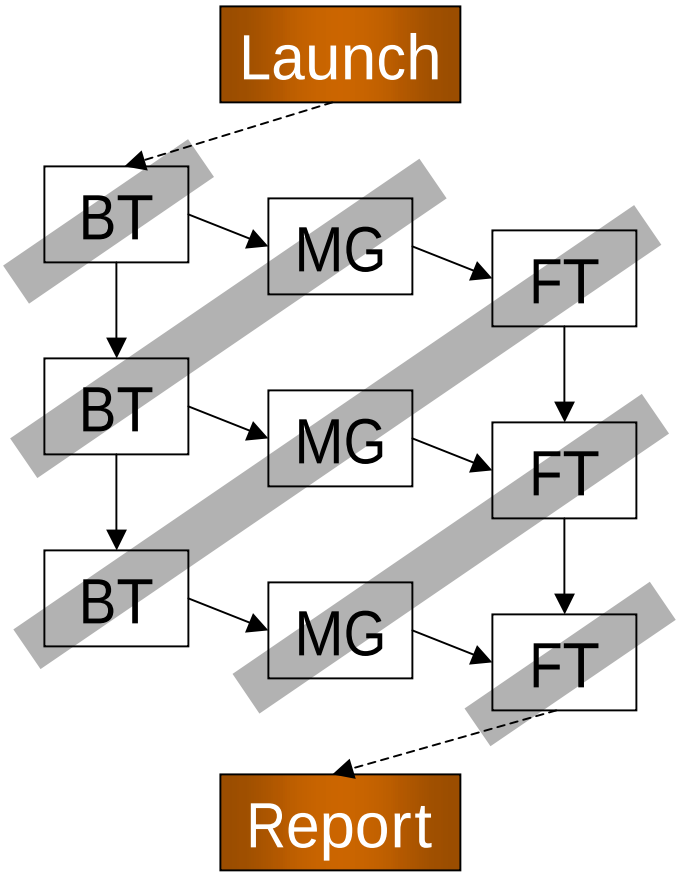
Helical Chain (HC)



Cyclic process (restart)

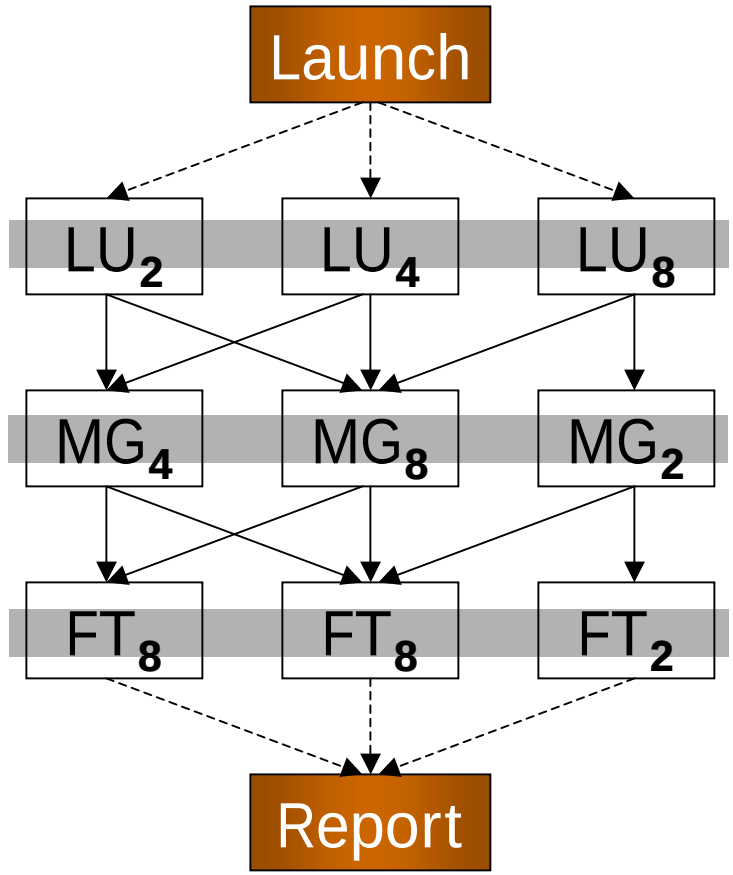
NGB Data Flow Graphs, Sample size

Visualization Pipe (VP)



Visualization cycle

Mixed Bag (MB)



Unbalanced chain

Grid Benchmarking Issues

- Are proposed DFG's representative?
- What other apps to use besides scientific computing?
- How to specify resources used in a uniform manner?
  - how to determine what resources were used?
- Is turnaround time best measure?
- How to scale up NGB problem sizes?
- Should NGB be called "benchmarks"?
- How to interpret results?
  - couple with system-level probes?

Interpretation of NGB results

Primitive Grid services

- Functionality
- Consistency among runs

Complete resources reservation

- Variation of single resource

Definition of uniform Grid currency (G$)

- NGB performance per G$

Definition of convertible Grid currency (US$)

- NGB performance per US$

Grid maturity increases