GridBench

The Road to M12

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Criteria for selecting benchmarks to be implemented

1) Source Code availability, licensing
2) Addresses issues that CrossGrid is currently facing
3) Simple Algorithm
4) Simple compilation and execution
5) Input/Output requirements

Criteria for selecting benchmarks to be implemented (continued)

6) Relation to CrossGrid applications

7) Well-known / Recognition

8) Micro-benchmark

9) Micro-kernel benchmark

10) Language and Middleware (Preferably C, MPI based)

11) D2.3 is
"Write first prototypes on a local cluster at the test site..." (Annex p61-62)

Candidates

- From the design document (D2.2) for GridBench:

	Micro-benchmarks	Micro-kernels	Application Kernels
CE/SE	gb_ce_lin gb_se_io		
SITE	gb_site_lmnet gb_site_mpbench	gb_site_hpl gb_site_nas_is gb_site_nas_ep gb_site_nas_ft gb_site_nas_bt gb_site_nas_sp gb_site_nas_lu	[Task1.1] Kernels [Task1.2] Kernels [Task1.3] Kernels [Task1.4] Kernels
GRID	gb_grid_net	gb_grid_lin gb_grid_npb	gb_grid_cg

Source Code Availability

- Most candidate benchmarks match this criterion which was a criterion for selection for inclusion in the Design Document in the first place
- Benchmarks based on CrossGrid application kernels do not satisfy this since they are at different stages of development (most are at very early stages)
  - Thus benchmarks based on X# application kernels will not be further discussed for inclusion in the M12 prototype.

Address main Grid issues

- Performance in CrossGrid and Grids in general is hugely dependent on networking (depending on type of application)
  - Networking is directly measured by *gb_site_lmnet* which is based on the LMBench benchmark. It measures p2p bandwidth.
  - Networking will also be measured directly by gb_grid_net (based on LMBench & NetPerf) but it design is still incomplete.
  - MPI benchmarks also measure networking but indireclty.

Address main Grid issues

- Performance of individual sites is a primary factor in grid performance (much like a CPU is to a regular machine)
  - The NAS kernels (*gb_site_nas_*)* are designed for measuring performance of parallel machines; in our case individual sites (clusters)
  - The High-Performance Linpack (*gb_site_hpl*) measures performance (using MPI) in a way that has come to be somewhat of a standard.

Simple Algorithm

- The gb_site_lmnet benchmark is based on simple and established methods of measuring latency and bandwidth.
- gb_site_hpl is based on a standard way of solving a set of linear equations (minor adjustments for efficient computation)

Simple Algorithm

- GridBench benchmarks based on NPB
  - gb_site_nas_is Integer Sort
  - gb_site_nas_ep Embarrassingly Parallel
  - gb_site_nas_ft Fast Fourier Transform
  - gb_site_nas_bt Block Tridiagonal solver
  - gb_site_nas_sp Pentadiagonal Solver
  - gb_site_nas_lu LU solver benchmark

Simple Algorithm

- NPB kernels measure:

	Math Functions	Network Bandwidth	Network Latency	Memory Bandwidth	Instruction Cache
EP	X				
IS		X		X	
CG		X		X	
MG				X	
FT		X		X	
BT		X			X
SP				X	
LU			X		X

Compilation and execution

- The NPB benchmarks are Fortran-based except the Integer Sort
- Compilation of both NPB and HPL is fairly straight forward

Input/Output requirements

- gb_site_lmnet takes only parameters as inputs (i.e. No input files).
- gb_site_lmnet Output is to standard out and is fairly short
- Output of HPL and NPB is fairly concise as well
- Optional output one NPB kernel can serve as input to another, thus enabling the use of Data Flow Graphs as in NAS Grid Benchmarks 1.0.
- NPB defines “classes” for problem sizes, HPL does something similar.

Input/Output requirements

- Sample LMBench output (gb_site_lmnet will change this into an XML format)

...
UDP latency using localhost: 102.4115 microseconds
TCP latency using localhost: 152.6223 microseconds
RPC/tcp latency using localhost: 272.0483 microseconds
RPC/udp latency using localhost: 196.4605 microseconds
TCP/IP connection cost to localhost: 370.7143 microseconds
initial bandwidth measurement: move=10485760, usecs=281390: 37.26 MB/sec
move=286261248, XFERSIZE=65536
Socket bandwidth using localhost: 40.78 MB/sec
Avg xfer: 3.2KB, 41.8KB in 13.2600 millisecs, 3.15 MB/sec
[Networking remote to kottos: Linux kottos 2.4.7-10smp #1 SMP Thu Sep 6 17:09:31 EDT 2001 i686 unknown]
UDP latency using kottos: 387.4001 microseconds
TCP latency using kottos: 721.3188 microseconds
RPC/udp latency using kottos: 1283.4001 microseconds
kottos/tcp: RPC: Timed out
TCP/IP connection cost to kottos: 402.5385 microseconds
initial bandwidth measurement: move=10485760, usecs=1420074: 7.38 MB/sec
move=55050240, XFERSIZE=65536
Socket bandwidth using kottos: 7.01 MB/sec
...

Relation to X# applications

- Given the unavailability of most X# application kernels this is a difficult criterion to satisfy.
- The HEP kernel is available but it may require some effort to adapt into a benchmark.

Well-known / Recognized

- Applying well-known recognized kernels/ benchmarks is desirable
  - User's confidence
  - Comparison with other architectures
  - Verified and refined
- HPL and NAS satisfy this
- HPL is probably more well-known
- While LMBench is not as well-known it employs standard techniques

A micro-benchmark and a micro-kernel

- gb_site_lmnet is ofcourse a micro-benchmark that focuses on network
- HPL and the NPB kernels are micro kernels. (either hpl or **one** of the NPB)

Language and Middleware

- gb_site_lmnet is based on LMBench which is written in C and uses TCP/UDP for measurements
- HPL is written in C and MPI
- NPB kernels use MPI and fortran (except IS which uses C)

Targeted at local cluster (D2.3)

- Both HPL and NPB kernels are suited for clusters
- While the network aspect of LMBench (the base of gb_site_lmnet) involves only two hosts, gb_site_lmnet will be suited to characterize the local network at a site.
- gb_grid_net (based on LMBench and NetPerf) is not targeted at clusters but inter-site networking. Despite this it should probably be considered as it is useful during the testbed rollout.

Decision

- Micro-benchmark: gb_site_lmnet
- Micro-Kernel: gb_site_hpl
  - Selecting one out of the NPB kernels would be an arbitrary decision since each measures a slightly different aspect of a parallel machine
- Suggestions?

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Measurements and experiments

- The targets of D2.3 will be to measure processing power and networking at local sites.
- gb_site_lmnet will measure network bandwidth and latency (p2p, for TCP and UPD) in B/s and s respectively
  - Since measurements are inherenty point-to-point it would be of interest to measure the network performance of the cluster with concurrent transfers between several pairs of cluster nodes.

Measurements and experiments

- gb_site_hpl will measure the computational performance of a local cluster in solving a standard set of linear equations.
- Measurements will be given in flops for Rmax.
  - **Rmax** - Maximal LINPACK performance achieved
  - Nmax - Problem size for achieving Rmax
  - N1/2 - Problem size for achieving half of Rmax
  - Completion time in seconds.

Topics to be covered

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Submission/execution

- Benchmarking jobs will be submitted via RSL
  - It is still unclear if JDL matches the RSL capabilities
  - RSL is more generic and employed in many Grids
- Will probably use “fork” and mpirun
- Open issues with RSL:
  - Does using RSL (thus circumventing(?) the Resource Broker) create problems.
  - Controlled environment (i.e. No other jobs running on the same resources)

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Tools for measurement

- Measurement will be made by the (already) instrumented code of HPL and results will be delivered to standard output which will be delivered by the middleware to the user interface machine.
- Measurements by the instrumented gb_site_lmnet code are again delivered to standard output.
  - If using multiple concurrent measurements between pairs of hosts, the measurements will be aggregated.
- At this point there will be no *external* measurements

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Result archival

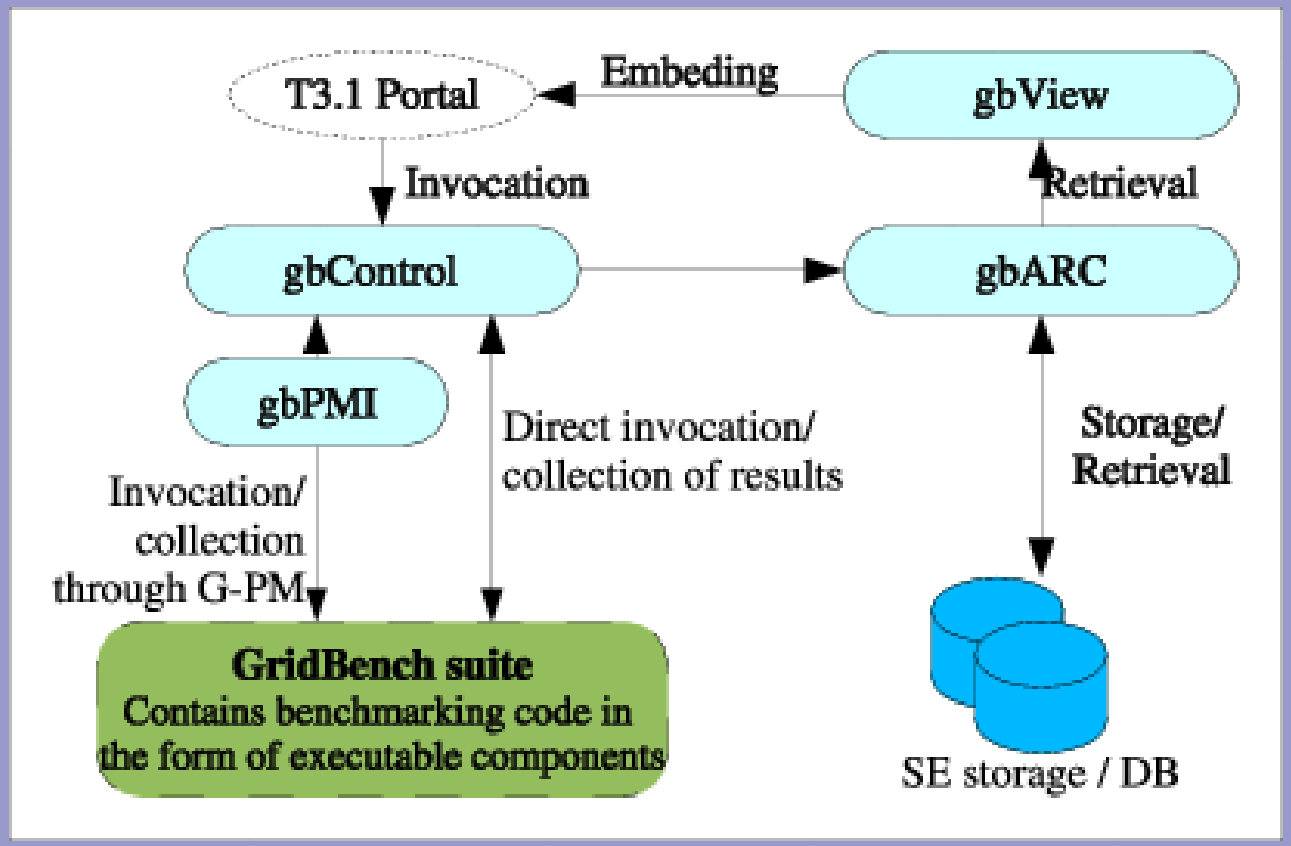
- Output delivered to the user's machine will be in XML format (to be specified)
- Archival of the results will be initially in a flat directory of XML files (i.e. No real archival, no central repository)
  - XML Archival/storage technologies (in RDMS?)
- Retrieval will be implemented after the archival method is specified.
  - In the end product, the user will have the ability o browse previous benchmark results

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

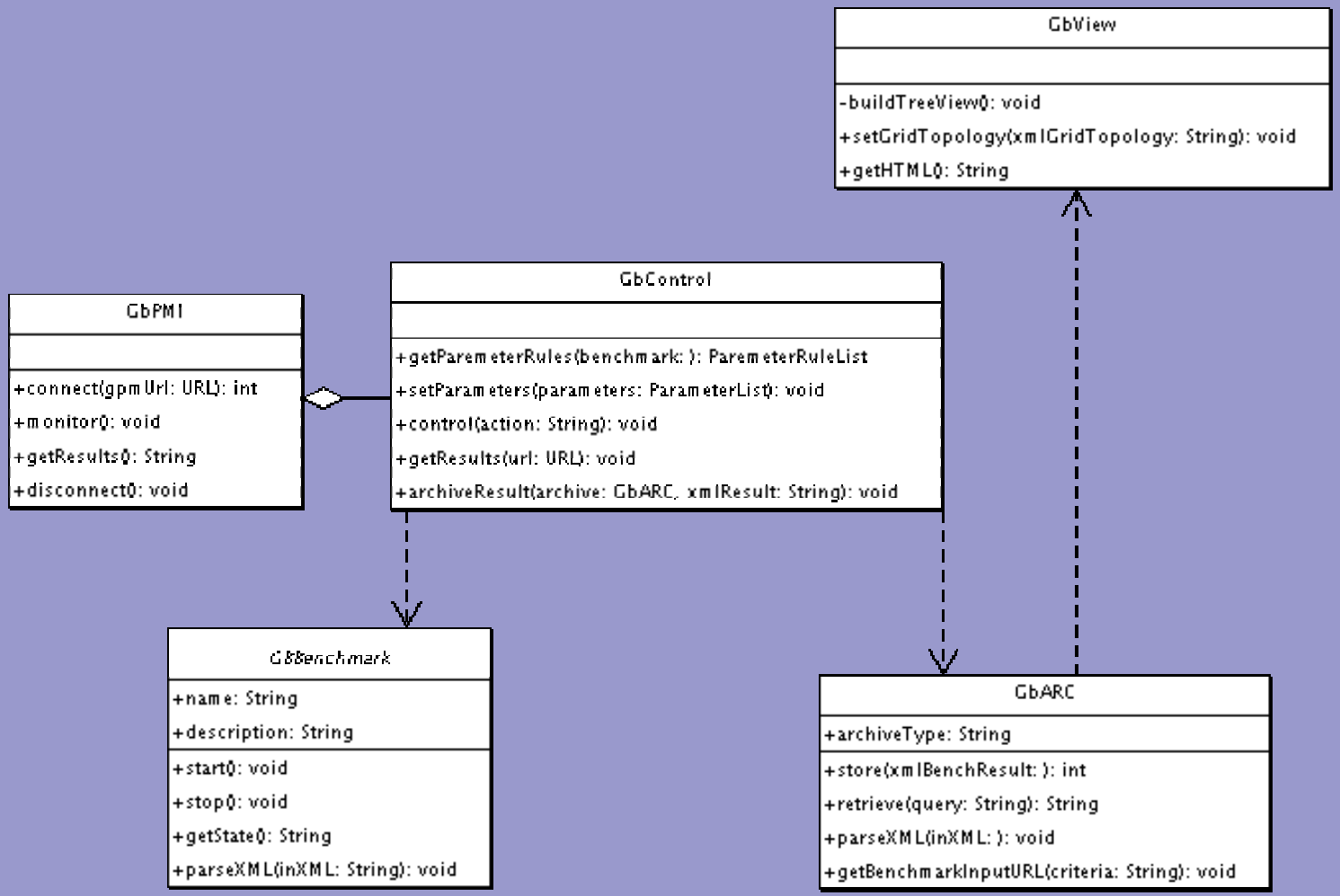
General About GridBench

- High-Level Diagrammatic View



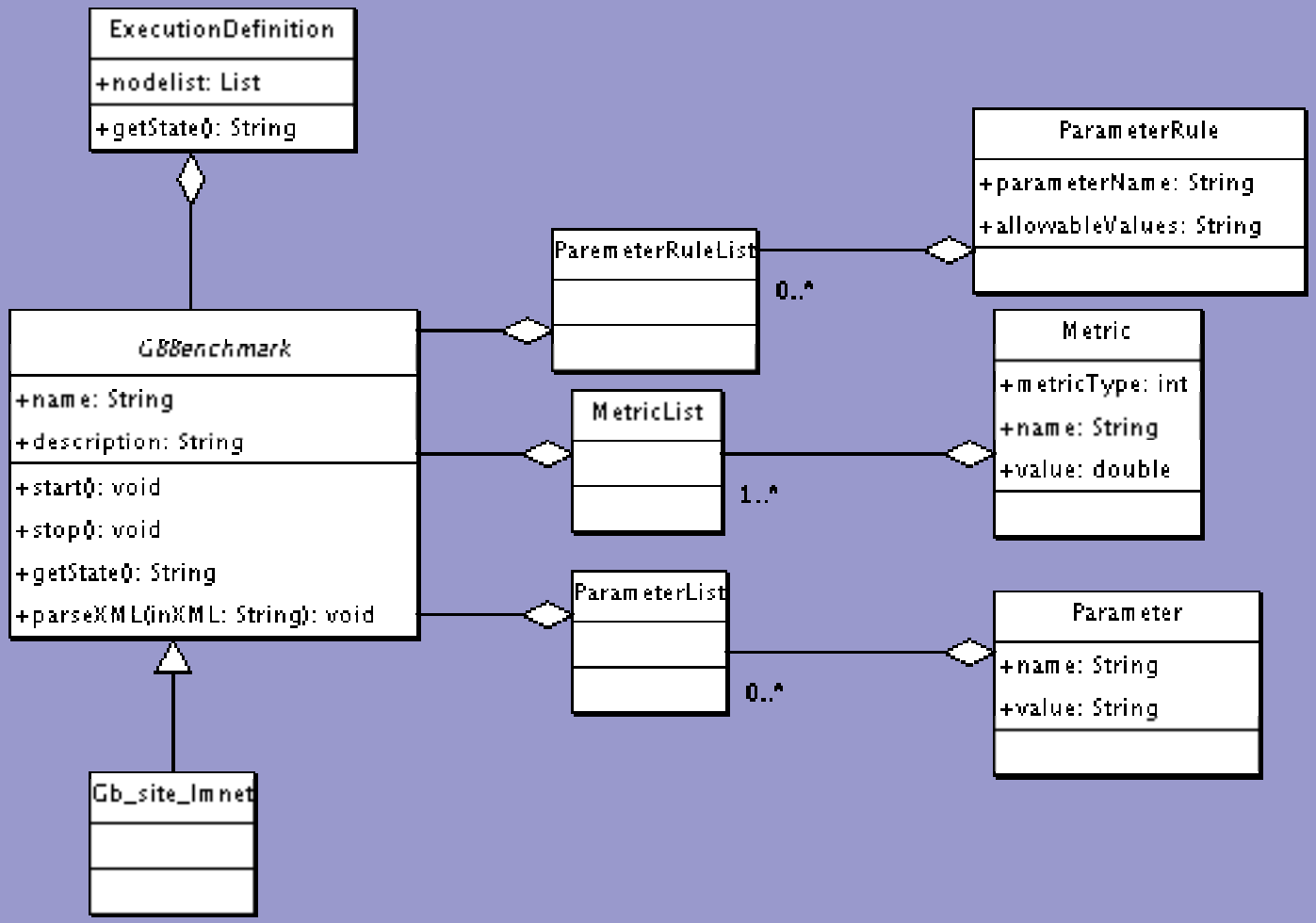
General About GridBench

- Utility Components



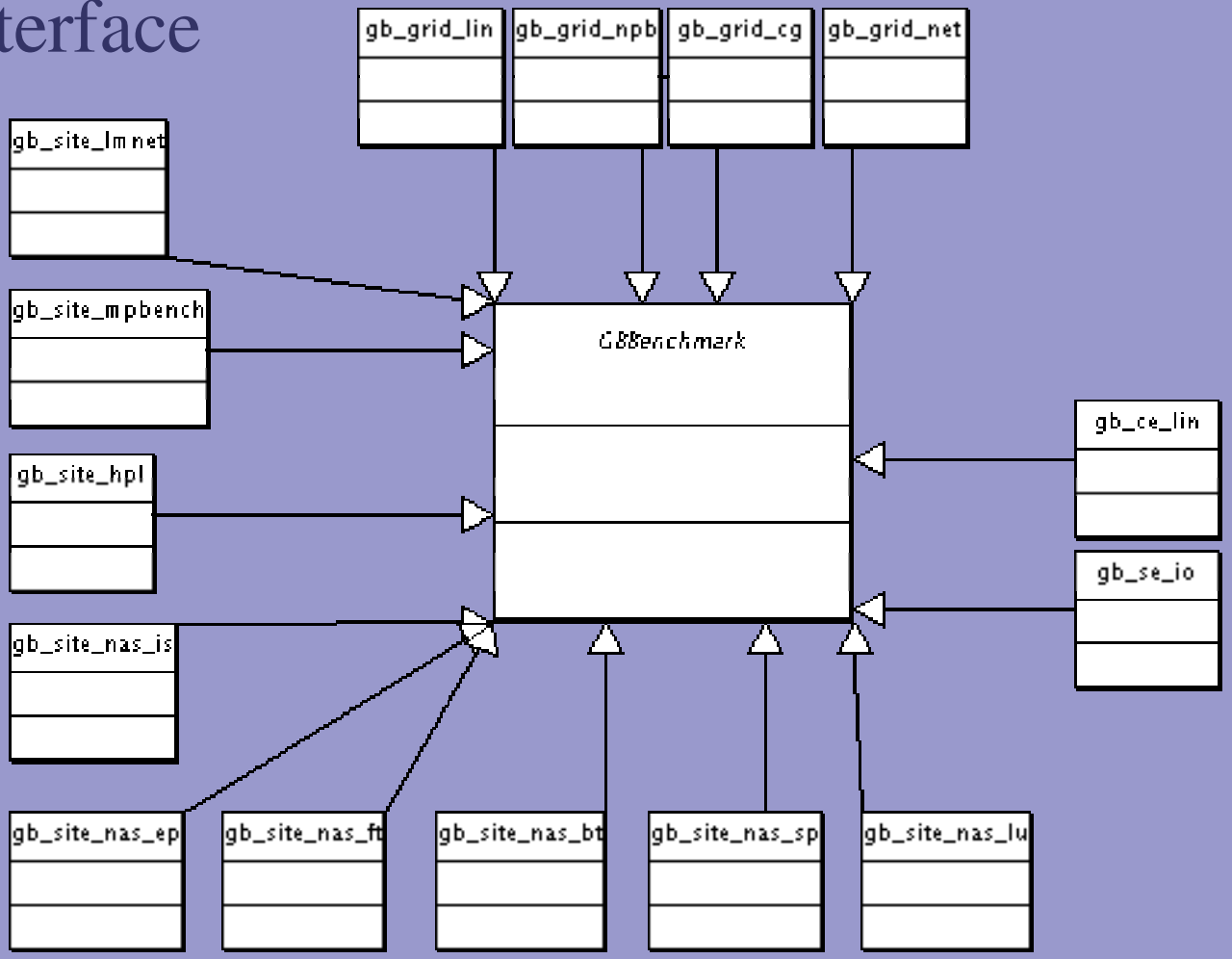
General About GridBench

- The common benchmark interface



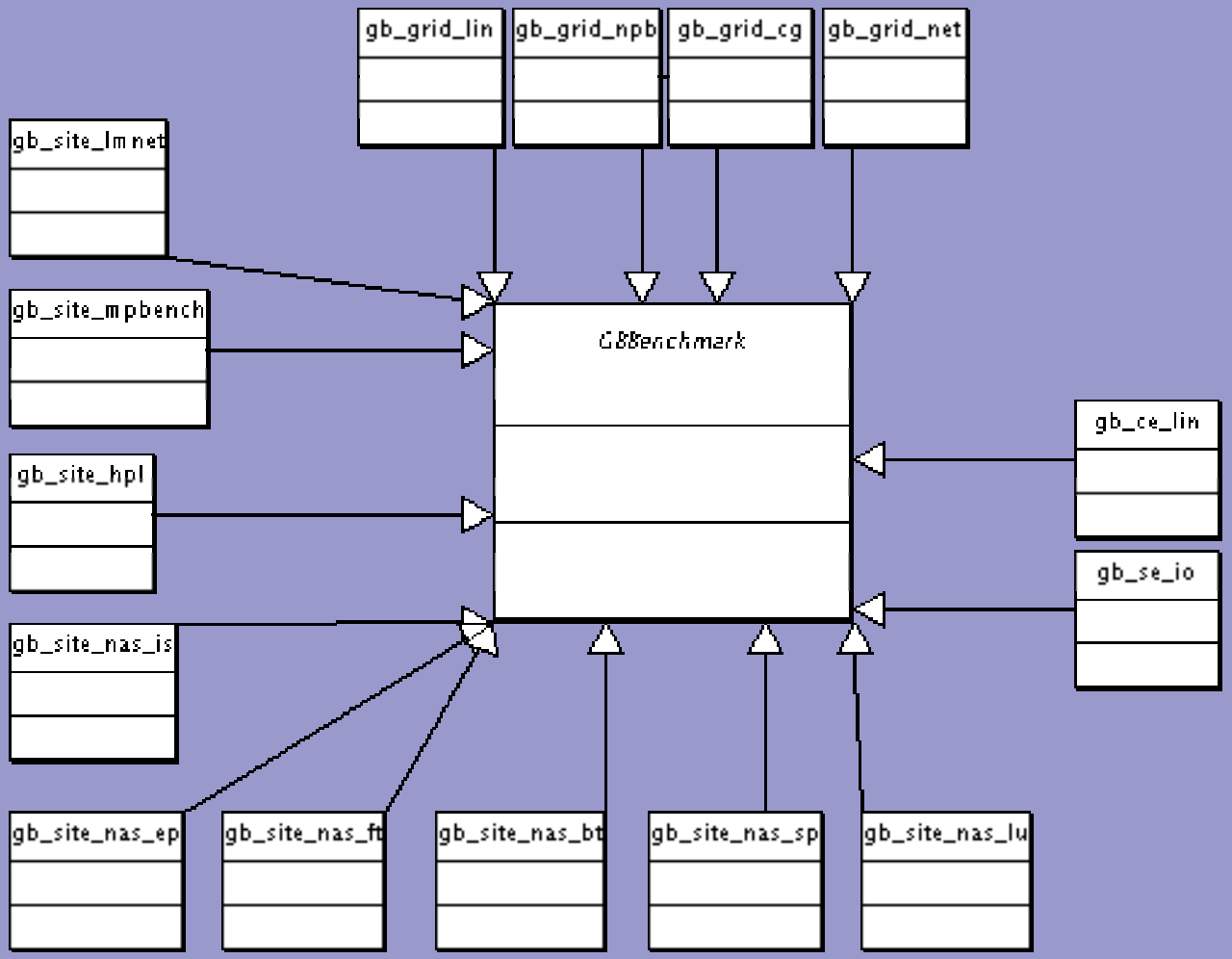
General About GridBench

- All benchmarks implement the GBBenchmark interface



General About GridBench

- High-level view of GridBench



Usecase: Determining site performance

- The user wishes to investigate the performance of a site
- The user chooses to run the gb_site_hpl benchmark (Linpack)
- The user logs into the portal and launches the benchmarking interface, GBView.
- The user selects gb_site_hpl and specifies the site.
  - Specification of the site depends on portal implementation

Usecase: Determining site performance

- The user will (in GBView)
  - specify the number of machines to run the benchmark on;
  - specify problem size;
  - Choose to launch the benchmark.
- GBControl will generate RSL (or JDL) and submit it
- The user has the option to cancel a benchmark job and not archive the result

Usecase: Determining site performance

- GBControl will acquire the output generated by the benchmark
- GBControl will reformat output to XML (if not already in XML)
- GBArc will parse the XML delivered by GBControl and archive it in a database (RDBMS).
- GBView will retrieve the results from GBArc and display them for the user

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Pointers to documentation

- Jack Dongarra, P.L. and A. Petitet, *The LINPACK Benchmark: Past, Present, and Future*. December, 2001.
- McVoy, L.W. and C. Staelin, *lmbench: Portable Tools for Performance Analysis*, in *USENIX Annual Technical Conference*. 1996. p. 279-294.
- David Bailey, T.H., William Saphir, Rob van der Wijngaart, Alex Woo, Maurice Yarrow. *The NAS Parallel Benchmarks 2.0*. in *The International Journal of Supercomputer Applications*. 1995.
- Rob F Van Der Wijngaart, Michael Frumkin. *The NAS Grid Benchmarks Version 1.0*. July,2002

Topics

- Benchmark Selection
- Measurements and experiments
- Submission/execution
- Tools for measurement
- Result archival
- Use-case
- Pointers to documentation
- Roadmap to M12

Roadmap to M12

- By end of M10
  - Develop a working GBControl prototype, and the GBBenchmark interface
  - Develop gb_site_hpl
- By end of M11
  - Specify and develop gb_site_lmnet