JT's first slide

* I like Brian's first slide
* can't optimize failure, can optimize speed.

NIKHEF pdp

JT's second slide

The application itself can support such access via intelligent use of the DMS. An example mechanism is an experiment-specific program (or servlet) that dispenses events. This servlet would receive the results of the query, e.g. as a list of (LDN, objref)-tuples as input, and as output would dispense event data objects in a manner compatible with the experiment software. In the background, we expect the event dispenser to contact the Data Management System to discover access costs for the various DSs, and then start accessing them in an advantageous order, queuing the events for consumption as the bytes arrive. From the application's viewpoint, the servlet object provides data on demand, and the application need not be concerned with access optimization.

CDN Demo

* Use SRMs as is for origin server (http?)
* use Coral for proxy network as-is
* Root access via http to start with
* metrics
  * job failures due to DM at least 3 times better
  * access / eff not worse than now
  * steal others metrics

NIKHEF pdp

work

* deployment
* observe behavior, response of system
* what is real volume of accessed data
* what is access freq distribution?
* how much work are holes (security?)

NIKHEF pdp

participants

* Mike Freedman (Coral)
* Rene Brun (Root)
* BiG Grid (SARA/Nikhef) site, security eval?)
* Brian Bockelman (UNL, site)
* Dirk Duellemann
* other sites to run caches?

Miron is watching

Users? Everybody!!

NIKHEF pdp