

Event Sizes In Objy

Jacek Becla

Stanford Linear Accelerator Center



Measured

	mh evt	aver evt	aver evt	extracted
	10.2.3i	10.2.3i	11.10.1	analysis-13
evshdr	399	1019	1389	274
evt	492	749	985	338
tag	687	742	880	638
col	81	143	287	21
aod	5491	3860	3363	2912
esd	15957	10652	10998	-
total nav	1659	2653	3541	1271
nav+aod	7150	6513	6904	4183
nav+esd	17616	13305	14539	-
nav+aod+esd	23107	17165	17902	-

- ◆ Each measurement slightly different
 - Different # events
 - Different releases
 - Rel 10: events that did not pass filters stored (empty headers), Rel 11: not stored
- ◆ Exercise:
 - Take one average event from recent release, break down to a byte level

	measured	detail breakdown
	11.10.1	12.2.0, with stateID
evshdr	1389*N	1458*N
evt	985*N	768*N
tag	880*N	576*N
col	287*N	8*N+51k
aod	3363*N	not avail
esd	10998*N	not avail
total nav	3541*N	2810*N+51k

- ◆ Detail breakdown - not captured:
 - Empty spaces at end of page
 - Object alignment
 - Internal objy overhead
(containers, databases, page tables)

Reducing size

- ◆ Reducing micro-level data size
 - Event store redesign
 - Optimizing AOD
- ◆ Compressing data
 - Server-side
 - Client-side

Event Store Redesign

- ◆ Not at expense of flexibility
 - All features will be preserved
 - Quite a few new important feature will be added
- ◆ Deadline: Dec'02 in test release
- ◆ Manpower: ~ 11 FTE-months
 - 1/4 + completed
- ◆ See BdbWebpage for details

Expected Sizes

	now	expected
evshdr	$1458*N$	0
evt	$768*N$	$446*N+14(?)k$
tag	$576*N$	$576*N$
col	$8*N+51 k$	$8*N+10(?)K$
aod	not avail	?
total nav	$2810*N+51k$	$1030*N+24(?)k$

- ◆ Evshdr+evt: 2.1 k \rightarrow 0.45 k (80% smaller)
- ◆ More reduction possible in tag
 - Need feedback from users

Optimizing AOD

- ◆ Visitor from Novosibirsk
 - 4 months, should arrive any day (waiting for visa)
 - No concrete estimates available
 - I'd dare to make a guess 3.4 kB → 2 kB feasible
 - After staring 30 min at random AOD database

Server-Side Compression

- ◆ Available
- ◆ turned off
 - requires starting new amses,
will restart during power outage (Aug 9-11)
- ◆ Does not scale (uses server CPU)

Client-Side Compression

- ◆ Increased priority/pushed 3 weeks ago
 - Objy will provide framework
 - ~5 weeks of work, cost (we pay) \$25K
 - ½ of what they usually charge, indeed reasonable price
 - Delayed release 7.1 by 3 weeks to include that feature for us
 - Delivery time: end of August'02
 - Bdb will provide shared library responsible for compression
 - Needed: packaging code that is now used on server side
 - ~2-3 weeks of work, includes testing → ~ 2 months of elapsed time
 - to be done by Andy Hanushevsky, lots travelling in August

How Does Our Data Compress

	%smaller
	gzip
evshdr	78
evt	80
tag	65
col	93
aod	51
Total	63

- ◆ Total = 60% if compressing each page independently

- ◆ New persistent classes may NOT compress that well
 - Very well “packed”
 - Very few repeated values
 - Still might compress tag
 - This is good (even if client-side compression available) - compression uses CPU
 - Example: new mini: only 27% smaller if compressed

Summary

- ◆ Micro with navigational components as seen in production on disk
 - Now ~ 7 kB
 - Expected ~ 3.2 kB by end of '02
- ◆ Client-side compression in objy
 - Available ~ end of Sep '02
 - Might not need when new event store in production