TO: Distribution

FROM: R. E. Mullen

DATE December 13, 1974

SUBJECT: Backup Performance Gains

This note attempts to quantify the performance improvements made to the Backup/Dumper subsystem over the past several months. The final estimate is based on several different measurements and analyses which all point to the conclusion:

	VCPU CHARGED	MEMU CHARGED	PGFLTS
Backup as % of Total System	19.7%	27.9%	?
Backup reduced by factor	.52	.63	.65
Total System saved	10.2%	17.6%	?

If a single overall figure representing the improvement in total system throughput attributable to the new Backup is required, it would be between 12% and 14% depending upon the relative weights assigned to the cpu and memory usage improvements.

Many thanks are due Mr. Roger Roach for providing System and SysDaemon monthly resource usage totals, as well as the results of his own metering of the old and new versions of bound_dumper_.

BRIEF SUMMARY OF IMPROVEMENTS:

In the months of June and July only the old version of backup was used. In those two months Backup/Dumper's charged cpu time was 20.2% and 19.3% of total system charged time, and Backup/Dumper memory charges were 27.3% and 28.6% of charged memory usage. An interim version of the Backup procedures was installed during the month of August, and although figures are given for August, they have not been used in any estimates of improvement. It should be noted that the version of Backup which calls the new hcs_ primitives was not installed until the start of December. Figures for November seem to be inconsistent, and although favorable to the new Backup, have not been used.

The interim version of Backup is seen to reduce Backup's cpu usage to .83 of what it had been. The final version which calls the two new hcs_ entries (list_inacl_all and status_for_backup) proposed in MTB-069 is shown to use .58 as much cpu as the interim version, with more than half the gain being due to list_inacl_all. Thus the final vcpu usage saves 52% of the original usage.

Unexpected, and more significant, were improvements to Backup's paging behavior. MTB-069 was totally inaccurate in supposing that Backup was cpu-bound and alleging that there was supporting data. The truth was Backup should have been cpu bound. A strategy was developed for avoiding the bulk of the paging overhead involved in copying the user's segments before writing them to tape. When Backup remained memory intensive, in spite of all analysis and reasoning to the contrary, heavy use of the new tool cumulative_page_trace led to the discovery of bugs causing from 38 to 58 unnecessary page references per directory or segment dumped. The best estimate of the total reduction in memory usage is 63%.

NORMALIZATION BY QUANTITY OF PRINTOUT:

The prime assumption made for this analysis is that the number of lines of IO Daemon usage charged to Backup and the dumper is proportional to the quantity of useful work done by Backup, that is to the number of segments and directories actually dumped. Averaged over a month this is a fair assumption provided the method used to compute the number of lines printed is not changed and provided that the format of the dump maps did not change. This normalization does not take into account variations in either the configuration or the load on the system.

	CPU HRS	MEM*K	LINES*K DPRINTED	CPU/ LINES	MEM/ LINES
BACKUP					
June	27.6	1385.0	438.9	.0630	3.16
July	24.6	1351.8	380.9	.0646	3.55
*Aug	24.4	781.7	435.2	.0561	1.80
Sept	21.1	501.4	398.8	.0530	1.26
Oct	22.0	472.6	418.1	.0526	1.13
Interim Savings (Backup)				17.2%	64.4%
DUMPER					
June	25.4	1008.9	529.6	.0480	1.90
July	24.2	1239.9	483.6	.0500	2.56
*Aug	21.3	602.7	458.7	.0461	1.31
Sept	18.8	401.3	451.4	.0416	0.89
Oct	18.6	356.0	441.8	.0421	0.80
Interim Savings (Dumper)				14.6%	62.1%

If account is taken of the fact, supported below, that the final version is observed to use .58 of the cpu used by the interim version, then the following estimates are obtained:

Final Savings (average)	51.3%	63.5%

NORMALIZATION BY ALL OTHER USERS:

The assumption here is that the ratio of Backup's resource usage to that of all other users is relatively constant from month to month. That this is approximately true can be seen in the following data by comparing figures for June to those of July or those of September to those of October.

MEMORY UNITS

	OTHER USERS	DUMPER & BACKUP	RATIO
June	6391.9	2393.9	.375
July	6469.2	2591.7	.401
*Aug	5340.6	1384.4	.259
Sept	5749.7	902.7	.157
Oct	5924.9	828.6	.140
Final Savings			61.7%

VCPU TIME

	OTHER USERS	DUMPER & BACKUP	RATIO
June	208.8	53.0	.253
July	204.3	48.8	.240
*Aug	200.5	45.7	.228
Sept	191.1	39.9	.209
Oct	201.6	40.6	.201
Interim Savings			16.8%
Final Savings			51.7%

OBSERVATION OF BACKUP:

Because the final version of Backup has been installed for a short time, it is necessary to compute directly the savings of the final version over those of the interim version. Fortunately such a computation is not difficult: one need merely examine the running final version and compute how much more time would have been used by the interim version.

By looking in the apt entries for Backup and Dumper their per process vcpu usage can be determined. Use of the meter_gate command allows determination of the number of calls to the new entries list_inacl_all and status_for_backup. Each call to status_for_backup replaced 5 calls to procedures using a similar amount of vcpu per call. Each call to list_inacl_all replaced 16 calls to procedures taking similar amounts of vcpu per call.

These figures vary from day to day, and running totals are not available. The particular data given below is not atypical.

	CALLS MADE	CALLS SAVED	VCPU/ CALL	VCPU SAVED
status_for_backup	3191	12764	.011sec	140sec
list_inacl_all	1558	23370	.012sec	280sec

SAVED (from meter_gate)	420sec
FINAL (from apte.vcpu)	587sec
INTERIM (FINAL + SAVED)	1007sec
FINAL/INTRIM RATIO	.58

RUNS BY ROGER ROACH:

Runs were made with both the old version of bound_dumper_ and the new on a loaded service configuration at MIT. Several system libraries were dumped. Because the ratio of directories dumped to segments dumped was much lower than that seen in actual practice, this method will not show gains made possible by list_inacl_all.

	CPU	PAGEFAULTS
Savings	35%	65%

Because this test did not measure the increased speed in the dumping of directories, it is consistent with other indications of an approximately 50% reduction in vcpu usage.