

*Avoiding
MegaWord Memory Leaks
On a Cray T90*

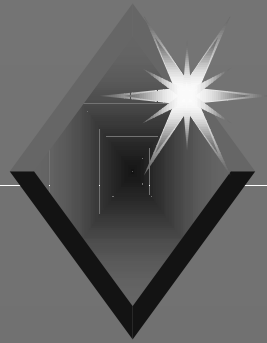
David J. Gigrich

May 18, 2000

Structural Analysis Computing

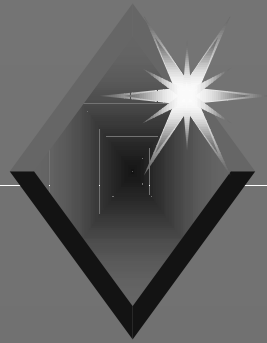
The Boeing Company

david.j.gigrich@boeing.com



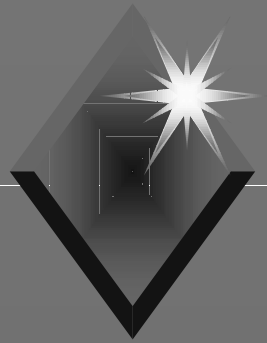
Topics of Discussion

- ◆ Introduction - Memory Management
- ◆ Memory Leaks
- ◆ Understanding the Nature of the Problem
- ◆ Solutions
- ◆ Results
- ◆ Conclusions



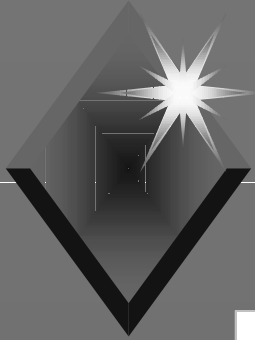
Memory Management

- ◆ Computer Memory
- ◆ Managers of Computer Memory
- ◆ Memory Management Allocation Schemes
 - ◆ Single User Contiguous
 - ◆ Fixed Partition
 - ◆ Dynamic Partition
 - ◆ Re-allocatable Dynamic Partitions
- ◆ Memory Fragmentation



Memory Leaks

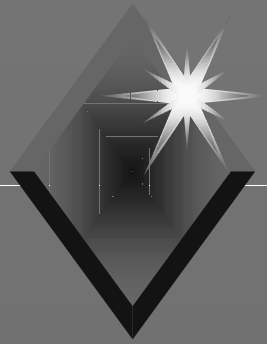
- ◆ Job Failure
- ◆ Problem Realization
 - ◆ Step Increase in Memory Requirements
 - ◆ Significant Increase in Memory Requirements
 - ◆ Example
- ◆ Identification / Verification



Example

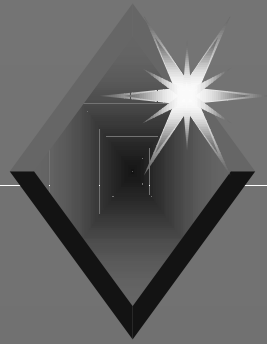
Program Area	
A	0.2MW
B	0.2MW
C	0.2MW
D	0.2MW
E	8.8MW
F	0.2MW
Heap area	

Program Area	
A	0.2MW
	0.2MW
C	0.2MW
D	0.2MW
G	0.1MW
	8.7MW
F	0.2MW
Heap area	



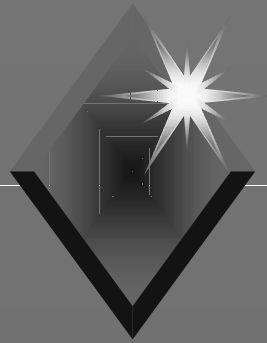
Understanding the Problem

- ◆ Operating System Memory Management
- ◆ Compilers
- ◆ Utility Routines
- ◆ The Application
 - ◆ Insufficient Computer Memory
 - ◆ Shared Resource



Solutions

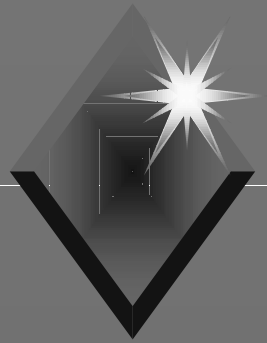
- ◆ Do Nothing - Leave as is
- ◆ Simple (Quick) Fix
- ◆ Software Redesign
(Comprehensive Solution)



Results

Job 1

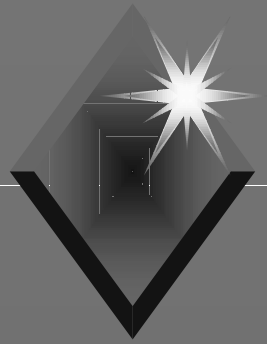
Software	Wall-Clock	CPU Time	Memory	Dollars
Original	10.8 hours	2680 sec.	46.6 MW	\$246
Simple Fix	3.1 hours	2924 sec.	17.5 MW	\$174
Redesign	2.4 hours	2949 sec.	13.3 MW	\$163



Results (Continued)

Job 2

Software	Wall-Clock	CPU Time	Memory	Dollars
Original	36.2 hours	2797 sec.	90.5 MW	\$379
Simple Fix	3.3 hours	3045 sec.	17.8 MW	\$180
Redesign	2.5 hours	3087 sec.	14.5 MW	\$173



Conclusions

- ◆ Better Computer Memory Utilization
- ◆ Software Developers Responsibility
- ◆ Over Management of Computer Memory