

AutoCheck: Automatically Identifying Variables for Checkpointing by Data Dependency Analysis (IEEE/ACM SC24[47])



Xiang Fu^{†1}, Weiping Zhang^{†1}, Xin Huang^{†1}, Wubiao Xu^{†1}, Shiman Meng^{†1}, Luanzheng Guo^{†2}, Kento Sato^{†3}

^{†1}: Nanchang Hongkong University, ^{†2}: PNNL, ^{†3}: RIKEN,



• Background

- Checkpoint-restart (C/R) is one of big I/O workloads in supercomputers
- To facilitate C/R, an application-level C/R tool (VeloC) is installed in Fugaku
- However, while we can specify only variables needed to be checkpointed, thereby, reduce the size of checkpoint data, finding all necessary variables to restart is challenging in large programs

• Approach

- To remove such manual coding, we developed AutoCheck which automatically detect variables to be checkpointed by static and dynamic data dependency analysis

• Results

- We successfully detects all the necessary variables for checkpointing in micro-benchmarks and proxy apps
- AutoCheck significantly reduces checkpoint size compared to a system-level C/R tool, BLCR

- Github: <https://github.com/zRollman/AutoCheck>

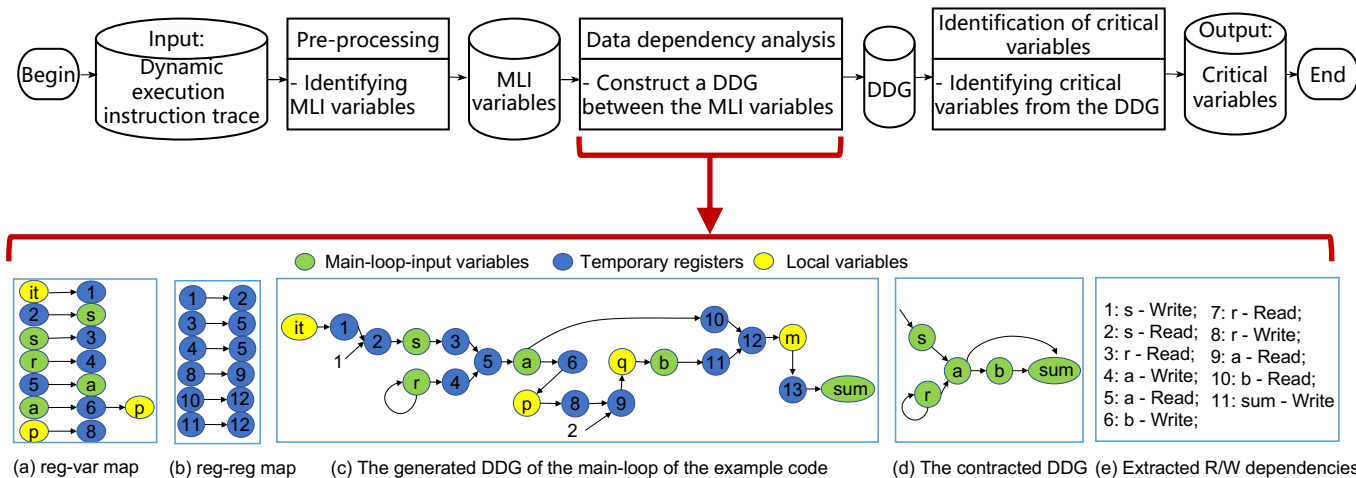


TABLE IV
STORAGE COST FOR CHECKPOINTING.

Name	Input size	BLCR [9] (MBs)	AutoCheck (MBs)
Himeno	129 × 65 × 65	32550.76	2.53
HPCCG	64 × 64 × 64	452202.50	610.9
CG	S	16569.47	0.16
MG	S	3220.39	2.84
FT	S	53616.26	24.6
SP	S	20068.88	7.81
EP	S	50061.67	0.03
IS	S	952.85	2.53
BT	S	34042.18	4.69
LU	S	17263.79	9.33
CoMD	-x 8 -y 8 -z 8	375798.50	241.71
miniAMR	-nx 8 -ny 8 -nz 8 -max_blocks 8	30310.90	0.09
AMG	-problem 2 -n 40 40 40	647577.68	58.83
HACC	-N 8 -t 16 × 16 × 16	837533.14	334.93