

Persistence is a widely used technique which allows the objects that represent the results of lengthy computations to outlive the process that creates it in order to considerably speed up subsequent program executions. We observe that conventional persistence techniques usually do not consider the application contexts of the persistence operations, where not all of the object states need to be persisted. Leveraging this observation, we have designed and implemented a framework called Pert, which first performs static program analysis to estimate the actual usage of the persisted object, given the context of its usage in the program. The Pert runtime uses the statically computed information to efficiently make tailoring decisions to prune the redundant and unused object states during the persistence operations. Our evaluation result shows that the Pert-based optimization can speed up the conventional persistence operations by 1 to 45 times. The amount of persisted data is also dramatically reduced, as the result of the application-aware tailoring.

