Parallel I/O and Checkpointing Strategies for PDE-constrained Optimization

Hendryk Bockelmann and Vincent Heuveline

Numerical Methods and High Performance Computing (NUMHPC),
University of Karlsruhe (TH)
Zirkel 2, 76128 Karlsruhe, Germany
{hendryk.bockelmann,vincent.heuveline}@rz.uni-karlsruhe.de
http://numhpc.rz.uni-karlsruhe.de

Key words: PDE-constrained optimization, High Performance Computing, parallel computing, optimal control, optimization software, flow control

1 Introduction

In many applications, the ultimate goal of numerical simulation is related to the optimization or optimal control of the considered system or process. In that context the solution process is usually much more involved than a purely forward simulation leading to a descriptive or predictive view of the considered system. Typical examples are the optimal control of a thermal treatment in cancer therapy and the optimal shape design of an aircraft. Assuming partial differential equations (PDE) for the state equations, the numerical treatment of the associated large PDE-constrained optimization problems is typically highly CPU time demanding and relies on dedicated solution process, highly tuned algorithms and implementation.

As compared to a mere forward simulation, i.e. solving the underlying PDEs for a given parameter, the process of optimization involves its systematic variation with respect to a given objective function. Ideally the ratio between the computational effort for the optimization and the CPU costs associated to the forward simulation should be small and independent of the discretization level of the considered PDEs i.e.

$$\frac{\text{effort of optimization}}{\text{effort of simulation}} = \text{constant}.$$

Especially for large scale instationary flow control problems general purpose optimization procedures do not comply with this law due to increasing importance of data I/O. In that context we propose new numerical methods which combine checkpointing techniques with parallel I/O. A main emphasis is put on the adequate use of recent parallel I/O technologies associated to large HPC-platforms in the context of instationary flow control problems.

2 Numerical approach and arising difficulties

We consider problems of the following form

$$\min_{y \in Y, u \in U} J(y,u) \tag{1}$$

subject to

$$c(y,u) = 0,\ u \in U_{ad} \subset U,\ y \in Y_{ad} \subset Y \tag{2}$$

where y (resp. u) describes the state (resp. control) variable. Further Y (resp. U) describes the state (resp. control) space and we assume $J : Y \times U \longrightarrow \mathbb{R}$ for the objective function and $c : Y \times U \longrightarrow Z$ for the state equation.

As model problem we consider $c(y,u)$ to be the instationary Navier-Stokes equations on a given domain $\Omega \subset \mathbb{R}^d,\ d = 2,3$ and $J(y,u)$ is a costfunctional of tracking-type, where we impose the velocity field y to match a given field without vortices

$$J(y,u) = \frac{1}{2} \int_0^T \left(\int_{\Omega_c} |y - y_d|^2 \, dx + \lambda \int_{\Gamma_c} |\nabla u|^2 \, ds \right) dt. \tag{3}$$

By standard Lagrange-formulation one can derive the adjoint problem associated to (2)

$$\tilde{c}(\bar{y}, \bar{u}, \bar{z}) = c'_y(\bar{y}, \bar{u})\bar{z} + J'_y(\bar{y}, \bar{u}) = 0 \tag{4}$$

and the optimality condition

$$(J'_u(\bar{y}, \bar{u}) + c'_u(\bar{y}, \bar{u})\bar{z}, u - \bar{u})_{U^*,U} \geq 0, \qquad \forall u \in U_{ad} \tag{5}$$

Considering standard FEM discretization of the above continuous problems ($c(y,u)$ and $\tilde{c}(\bar{y}, \bar{u}, \bar{z})$) we arrive at the short forms

$$y_h^{i+1} = F(y_h^{i+1}, y_h^i, u_h^i) \qquad i = 0, 1, ..., nt - 1 \tag{6}$$

$$z_h^i = R(z_h^{i+1}, y_h^i, u_h^i) \qquad i = nt - 1, ..., 0 \tag{7}$$

where we assume y_h^i and z_h^i to be the discrete solution of the primal resp. adjoint equation at timestep i and $F(\cdot)$ (resp. $R(\cdot)$) to be the discrete solution operators associated to $c(\cdot)$ (resp. $\tilde{c}(\cdot)$).

It is important to note that assuming a nonlinear problem the adjoint solution is directly coupled to the corresponding step of the primal solution. This means that the solution of the adjoint equation relies on the knowledge of the complete forward solution. To reduce the amount of data, checkpointing schemes have been introduced like

- uniform checkpoint distribution (two- or multilevel approaches) [1]
- binomial checkpoint distribution [2]
- adaptive (online) checkpointing [3][4]

Typically for 3D instationary problems the backup of the states easily grows up to some TByte. Further on HPC platforms one has to face the problem that the storage both for writing and reading the checkpoints can usually only be handled on a unique master node.

The needed gather and scatter routines are very expensive on typical clusters due to the fact that the performance of the network decreases dramatically the more processors you use in a collective operation (see Figure 1).

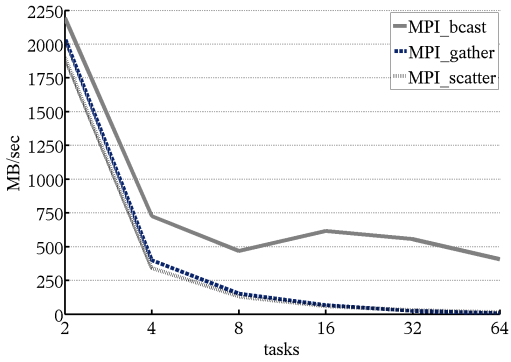

Fig. 1. Benchmarking of collective MPI-routines on HP XC6000 (University of Karlsruhe (TH)).

3 Solutions on HPC-platform

With respect to the previously described problems and order to solve the adjoint problem (4), we consider an hybrid approaches combining the following two approaches

- checkpointing-strategies: store some dedicated states in time instead of a complete backup and recompute missing states for the backward problem when missing
- parallel I/O: allows every processor in a cluster to write simultaneously on a harddisk

These two approaches result in different benefits. By an adequate choice of the time steps of the state solution by means of a checkpointing-strategy one can

- save up to 90 % of memory-costs
- restrict the computation overhead needed to recompute the missing time steps of the forward solution by a factor of two.

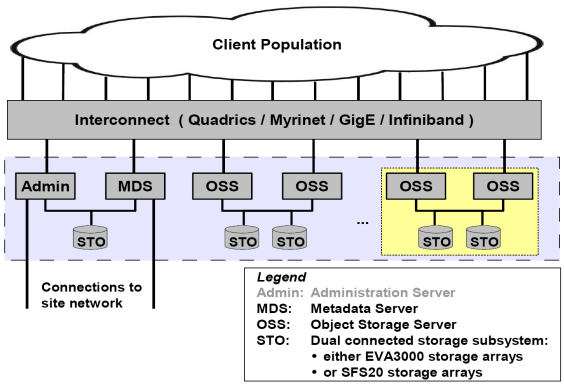

Fig. 2. System architecture of parallel data-system HP SFS (lustre).

A different benefit is given by the parallel I/O technique. These techniques allow every processor to access a dedicated storage in parallel (see Figure 2 and [5]).

For saving of a distributed vector (for example the i-th timestep of a forward solution) this results in

- no more need to gather the data on a master-processor
- reduction of the communication costs by generally 90 %.

In the proposed approach we define a strategy to find the right balance between both techniques in order to obtain the optimal performance on given HPC-platforms.

References

1. I. Charpentier. Checkpointing schemes for adjoint codes: Application to the meteorological model Meso-NH. *SIAM J. Sci. Comput.*, 22(6):2135–2151, 2001.
2. Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans. Math. Softw.*, 26(1):19–45, 2000.
3. V. Heuveline and A. Walther. Online checkpointing for parallel adjoint computation in pdes: Application to goal-oriented adaptivity and flow control. In W. et al. Nagel, editor, *Proceedings of Euro-Par 2006*, volume 4128 of *LNCS*, pages 689–699. Springer, 2006.
4. Michael Hinze and Julia Sternberg. A-revolve: an adaptive memory-reduced procedure for calculating adjoints; with an application to computing adjoints of the instationary Navier-Stokes system. *Optim. Methods Softw.*, 20(6):645–663, 2005.
5. Roland Laifer. Experiences with hp sfs / lustre in hpc production, isc2005 heidelberg, 2005. `http://www.isc2005.org/download/cp/laifer.pdf`.