A Scalable Numerical Algorithm for solving Tikhonov Regularization Problems in a large scale application

PPAM 2015 - Krakow, Poland September 6-9 2015

♦†♠ R. Arcucci, ♦† L. D'Amore, ♦S. Celestino, ♦G. Laccetti and †♣ A. Murli

{rossella.arcucci, luisa.damore, simone.celestino, giuliano laccetti, almerico.murli} @unina.it

♦ University of Naples Federico II, Naples, ITALY
♠ Imperial College London, London, UK
† CMCC, Euro-Mediterranean Center on Climate Change, Lecce, ITALY
♣ SPACI, Naples, ITALY

Imperial College London — collaboration — collaboration — Centro Euro-Mediterraneo sui Cambiamenti Climatici — collaboration — SPACI

We are concerned with large scale Tikhonov Regularization (TR) problems. TR is the most commonly used method of regularization for inverse and ill-posed problems. There are numerous examples of ill posed problems in computational mathematics and applications. The issue we face here is to solve large scale inverse ill posed problems efficiently. Efficiency is achieved by virtue of designing computational models specifically to take full advantage of massively parallel computers and General Purpose Graphics Processing Units (GPGPUs).

1. The Tikhonov-Regularized (TR) formulation

The TR problem can be described as following:

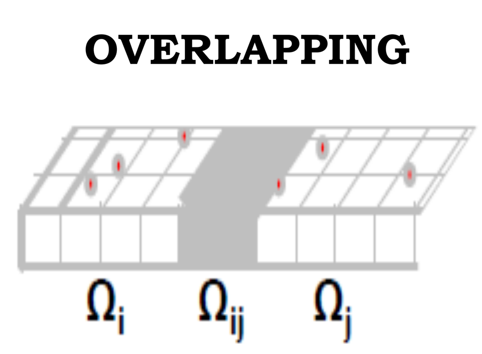
$$u(\lambda) = argmin_u J(u) = arming_u \left\{ \|\mathbf{Hu} - \mathbf{v}\|_R^2 + \lambda \|\mathbf{u} - \mathbf{u}^{\mathcal{M}}\|_B^2 \right\}$$

where $\|\cdot\|_R$ and $\|\cdot\|_B$ denote the weighted norms with respect to the error covariance matrices $\mathbf{B}$ and $\mathbf{R}$ and λ is the regularization parameter

2. The decomposed TR formulation

Let $\Omega \in R^3$ be the domain decomposed into a sequence of overlapping sub-domains $\Omega_i \in R^3$, such that:

$$\Omega = \bigcup_{i=1}^{p} \Omega_i \quad \Omega \subset R^3$$



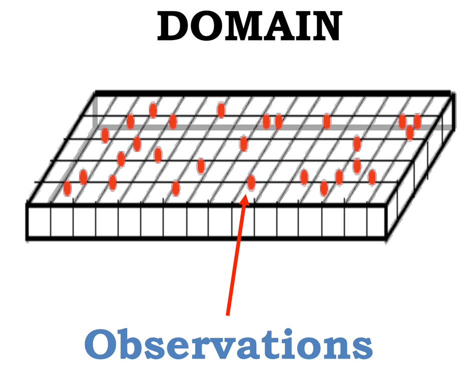
The decomposed TR formulation:

$$u_i^{DA} = argmin_{ui}\{J/\Omega_i + O/\Omega_{ij}\}$$

3. Performance Analysis – ScaleUp factor

If $p \in N$, and $p > 1$, the algorithm associated to the decomposition given is:

$$\mathcal{A}(\Omega, p) := \underbrace{\mathcal{A}(\Omega_1), \mathcal{A}(\Omega_2), \ldots, \mathcal{A}(\Omega_p)}_{\text{p times}}$$



Let $p1, p2 \in N$ and $p_1 < p_2$. Let $T(A(\omega, p_i)), i = 1, 2$ denote the time complexity of $A(\omega_i, p_i), i = 1, 2. \forall i \neq j$ we define the (relative) scale-up factor of $A(\omega, p_2)$, in going from p1 to p2, the following ratio:

$$S_{p2,p1}(N) = \frac{T(\mathcal{A}(\Omega, p1))}{(p2/p1)T(\mathcal{A}(\Omega, p2))}$$

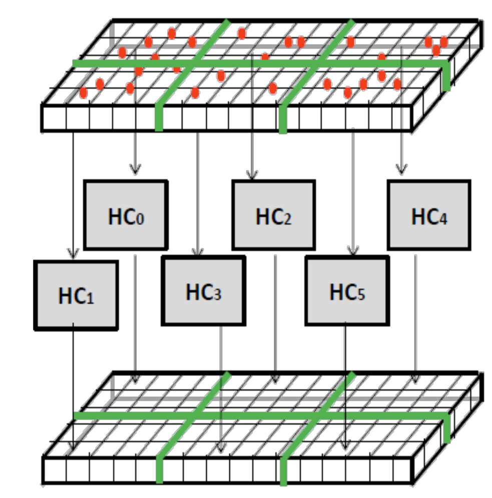
4. Case Study: Data Assimilation problem

Let $t \in [0, T]$ denote the time variable. Let $u^{true}(t, x)$ be the evolution state of a predictive system governed by the mathematical model M with $u^{true}(t_0, x), t_0 = 0$ as initial condition. Here we consider a 3D shallow water model. Let $v(t, x) = H(u_{true}(t, x))$ denote the observations mapping, where H is a given nonlinear operator which includes transformations and grid interpolations

5. Results

We consider two hybrid architectures: HA1 is a 288 CPU-multicores, HA2 is a GPU+CPU architecture

n	nproc	$T^{nproc}(N)$	$measuredS_{nproc,8}$	$S_{nproc,8}$
$O(10^6)$	8	2.0545e+02	1.0	1
	16	6.3316e+01	3.25	4
	32	2.0005e+01	10.27	16
	64	8.7835e+00	23.39	64
n	**nproc**	**$T^{nproc}(N)$**	**$measuredS_{nproc,8}$**	**$S_{nproc,8}$**
$O(10^7)$	8	-	-	-
	16	3.9091e+03	1.0	1
	32	9.9952e+02	3.91	4
	64	2.7584e+02	14.17	16



Results obtained on HA1 for a problem size $O(10^6)$ and $O(10^7)$ by using # thread-blocks = 2p

N	p	nproc	$T_{flop}^{GPU}(N)$
$O(10^7)$	1	2	0.144
	2	4	0.044
	4	8	0.025
	8	16	0.024

Values of execution time of algorithm running on GPU for a problem size $O(10^7)$

N	p	$T_{flop}^{GPU}(N)$	$measuredS_{nproc,2}$	$S_{nproc,2}$
$O(10^7)$	1	0.127	-	-
	2	0.027	4.7	4
	4	0.008	15.9	8
	8	0.007	18.1	16

Results on HA2: Values of T_{flop}^{GPU} and measured scale-up factor compared with theoretical once

6. Discussion

We now discuss scalability results shown in the tables. To this end, we introduce

$$s_{nproc}^{loc} = \frac{T_{flop}(N/p)}{T_{nproc}(N/p)}$$

which denotes the speed up of the (local) algorithm $\mathcal{A}(D_N(\Omega_i), N/p)$ for solving the local problem on subdomain $D_N(\Omega_i)$. Let us express the measured scale up factor in terms of s_{nproc}^{loc} . We have:

$$S_{1,nproc}^{measured} = \frac{T_{flop}(N)}{\frac{pT_{flop}(N/p)}{s_{nproc}^{loc}} + pT_{oh}(N/p)}$$

it comes out that

$$S_{1,nproc}^{measured} = \alpha S_{1,nproc} \qquad \alpha := \frac{s_{loc}^{nproc}}{1 + \frac{s_{nproc}^{loc} T_{oh}(N/p)}{T_{flop}(N/p)}}$$

Finally, it is worth noting that in our experiments, on HA1, local DA problems are sequentially solved, then $s_{nproc}^{loc} = 1$ and $\alpha < 1$, while on HA2, local DA problems have been concurrently solved on the GPU device, so $s_{nproc}^{loc} > 1$ and $\alpha > 1$,thus the above analysis validates the experimental results

References

1. R. Arcucci, L.D'Amore, S.Celestino,G.Scotti, G.Laccetti, A Parallel Approach for 3D-Variational Data Assimilation on GPUs in Ocean Circulation Models, International Conference on Scientific Computing, World Academy of Science, Engineering and Technology Proceeding, Vol. 2(5), ISSN 1307-6892, Paris, 2015.
2. . L. Carracciuolo, D'Amore, L. , Murli, A. Towards a parallel component for imaging in PETSc programming environment: A case study in 3-D echocardiography, Parallel Computing, Volume 32, Issue 1, January 2006, pp. 67-83.
3. L. D'Amore, R. Arcucci, L. Carracciuolo, A. Murli, A Scalable Approach to Variational Data Assimilation, Journal of Scientific Computing, Vol. 61, 2014
4. L. D'Amore, R. Arcucci, L. Carracciuolo, A. Murli -DD-OceanVar: a Domain Decomposition fully parallel Data Assimilation software in Mediterranean Sea - Procedia Computer Science 18, 2013, pp. 1235-1244.
5. L. D'Amore, R. Arcucci, L. Marcellino and A. Murli, HPC computation issues of the incremental 3D variational data assimilation scheme in OceanVar software - Journal of Numerical Analysis, Industrial and Applied Mathematics vol. 7, no. 3-4, 2012, pp. 91-105
6. L. D'Amore, D. Casaburi, A. Galletti, L. Marcellino, A. Murli – Integration of emerging computer technologies for an efficient image sequences analysis, Integrated Computer-Aided Engineering, vol. 18, Issue 4, 2011, pp. 365-378.