Chapter 7
Hierarchical Matrices:
A Context in Which New Features Are Vital

ABSTRACT

In this chapter, the authors present hierarchical matrices, which are a powerful numerical tool that allows reducing to a logarithmic order both the storage needs and computational time in exchange for a controlled accuracy loss, thanks to the compression of part of the original data to form low-rank blocks. This type of matrices presents certain particularities due to the storage layout, the different blocks configurations, and a hierarchically and nested partitioned structure of blocks; the presence of dense and low-rank blocks of various dimensions; and the recursive nature of the algorithms that compute the h-algebra operations. Thanks to the programming model OmpSs-2 and specifically to two novel features it incorporates, a fair parallel efficiency based on task-parallelism can be achieved in shared memory environments.

INTRODUCTION TO HIERARCHICAL MATRICES

Many engineering fields require powerful simulations of the systems they design, build, evaluate or analyze. Most of them do not need a fully accurate result and can afford precision losses in exchange for reduced computation times. That is the case, for example, of some of the applications on which rely

DOI: 10.4018/978-1-7998-7082-1.ch007

aeronautics simulations of pressure, temperature, tension, etc. that employ Boundary Element Methods (BEM).

In this chapter the authors will introduce Hierarchical Matrices (H-Matrices) (Hackbusch, 1999) as one of the numerical tools employed when solving BEM (Bebendorf, 2008) (Casenave, 2013), and then the authors will focus on how task-parallelism becomes the easiest strategy to exploit concurrent executions of operations involving H-Matrices, especially when leveraging the novelty features included in task-parallel programming models such as OmpSs-2.

LINEAR ALGEBRA BACKGROUND ON HIERARCHICAL MATRICES

From a mathematical perspective, hierarchical matrices are included in the set of compressed structures (Cheng et al., 2003). This is because they are built employing procedures that remove the "less representative" entries (which is understood as compression) until a pre-established ratio between accuracy loss and future computations acceleration is achieved. Conceptually, the compressed structures lay between dense and sparse linear algebra elements, as there is a presence of null elements in them, but not that high to consider them sparse matrices.

Now the authors present some mathematical definitions that are necessary for understanding the foundations of H-Matrices (Grasedyck et al., 2003) (Hackbusch et al., 2004) (Hackbusch, 2015). The first terms that need to be presented are the ones employed in the process of determining the "importance" of a certain matrix entry with respect to the others: eigenvalues, eigenvectors and singular values.

Definition 7.1. (Eigenvalues and eigenvectors) Let $A \in R^{n\times n}$, then if there exists a nonzero vector $\upsilon \in R^n$, and a nonzero scalar $\lambda \in \mathrm{R}$ *s*uch that $\mathrm{A}\upsilon = \lambda\upsilon$ has a nontrivial solution, then υ is an eigenv*ector of A,* and λ is an eigenva*lue of A*.

Definition 7.2. (Singular values) Let $A \in R^{m\times n}$, then let $\lambda_1, \lambda_2 \ldots \lambda \in R$ be the eigenvalues of A^TA (with repetitions). Order them in such a way that $\lambda_1 \geq \lambda 2 \geq \ldots \lambda_n \geq 0$ and let $\sigma_i = \sqrt{\lambda_i}$, so that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$. Then the scalars $\sigma_i \in R$ are the singular values of A.

With these two definitions in mind, prior to understanding how the compression of data is performed in H-Matrices and similar structures, two

Related Content