

Application BASTRI

Fiches Equipes

DATAMOVE (SR0718BR)

Data Aware Large Scale Computing

DATAMOVE □ (DATAMOVE (SR0799BR) , DATAMOVE (SR0799BR))

Statut: Terminée

Responsable : Bruno Raffin

Mots-clés de "A - Thèmes de recherche en Sciences du numérique - 2023" : *Aucun mot-clé.*

Mots-clés de "B - Autres sciences et domaines d'application - 2023" : *Aucun mot-clé.*

Domaine : Réseaux, systèmes et services, calcul distribué
Thème : Calcul distribué et à haute performance

Période : 01/01/2016 -> 31/10/2017

Dates d'évaluation :

Etablissement(s) de rattachement : CNRS, GRENOBLE INP, UGA
Laboratoire(s) partenaire(s) : LIG (UMR5217)

CRI : Centre Inria de l'Université Grenoble Alpes

Localisation : Laboratoire LIG- Bâtiment IMAG

Code structure Inria : 071125-0

Numéro RNSR : 201622038P

N° de structure Inria: SR0718BR

Présentation

Aujourd'hui les plus grands supercalculateurs (classement du Top500) sont composés de centaines de milliers de coeurs de calcul, atteignant des performances de l'ordre du PetaFlops. Déplacer des données sur de telles machines devient un goulet d'étranglement majeur. La situation devrait empirer avec les machines exaflopiques, les capacités de transferts de données augmentant moins vite que celles de calcul. Les unités de calcul disponibles seront très probablement sous-utilisées, limitées par les capacités de transferts. La hiérarchie mémoire et le stockage sur ces machines devrait changer significativement avec l'avènement des mémoires non volatiles (NVRAM), nécessitant de nouvelles approches pour la gestion des données. Les mouvements de données sont par ailleurs une source importante de consommation d'énergie, et donc une cible pertinente pour améliorer le rendement énergétique des machines.

L'équipe DataMove se consacre à ces enjeux, menant des recherches sur l'optimisation des mouvements de données pour le calcul intensif. DataMove travaille sur quatre axes de recherche:

- Intégration de l'analyses de données et du calcul intensif
- Ordonnancement par lot (batch) prenant en compte les mouvements de données
- Etude empirique des plateformes à grande échelle
- Prédiction de la disponibilité des ressources.

Le gestionnaire de tâches et de ressources est en charge de l'allocation des ressources lors des demandes d'exécutions par les utilisateurs (quand et où exécuter une application parallèle). L'augmentation du coût des mouvements de données nécessite des politiques d'ordonnancement adaptées capables de prendre en compte l'influence des communications internes à l'application, les I/O ainsi que la congestion liée au trafic généré par les applications concurrentes. Modéliser le comportement des applications, typiquement par des techniques d'apprentissage, pour anticiper l'usage effectif des ressources sur ces architectures est un autre enjeux critique pour améliorer les performances (temps, énergie). L'ordonnanceur doit aussi gérer efficacement les nouveaux types d'applications. Les plateformes haute performance doivent supporter de plus en plus des tâches de traitements intensifs de données en plus des traditionnels calculs de simulation numérique. En particulier, la masse toujours croissante de données générées par les simulations numériques motive une intégration plus poussée entre la simulation et l'analyse de résultats. L'objectif est de réduire le trafic de données et d'accélérer l'analyse des résultats en effectuant le traitement des résultats (compression, indexation, analyse, visualisation, etc.) au plus proche de là où elles sont créées. Cette approche,

Contact

- **Responsable :** Bruno Raffin
- **Tél :** + 33. 4. 5.6 .52. 7.1 .39
- **Secrétariat Tél :** + .33. 4. 7.6 .61. 5.5 .33

En savoir plus

- Site sur inria.fr
- Derniers Rapports d'Activité : [2016](#) , [2017](#) , [2018](#) , [2019](#) , [2020](#) , [2021](#) , [2022](#) , [2023](#)

Documents sur la structure

- [Intranet](#)
- [Privés](#)

Décisions

- [11298](#) (01/12/2015) : création
- [11914](#) (16/12/2016) : prolongation
- [12256](#) (16/06/2017) : prolongation

Localisation

- **Adresse postale :** 150 place du Torrent Bâtiment IMAG - Campus de Saint Martin d'Hères 38421 Saint Martin d'Hères
- **Coordonnées GPS :** 45.191, 5.765

appelée analyse in-situ, nécessite de revisiter le workflow traditionnel (calcul en batch puis analyse postmortem). L'application devient un tout incluant la simulation numérique, les traitements in-situ et les I/O, motivant le développement de stratégies d'allocation de ressources adaptées, de nouvelles structures de données et d'algorithmes d'analyse massivement parallèles pour entrelacer efficacement l'exécution des différents composants de l'application et globalement en améliorer les performances.

Pour traiter ces problèmes, nous combinons recherche théorique et développements pratiques en mode agile, pour concevoir des solutions polyvalentes et efficaces répondant aux besoins du domaine d'app

Axes de recherche

Integration of High Performance Computing and Data Analytics

The base idea behind in-situ processing is to perform data analytics as closely as possible to the running application, while ensuring a minimal impact on the simulation performance and the required modifications of its code. The analytics and the simulation thus need to share resources (compute units, memory, network). Today solutions are mostly ad-hoc and defined by the programmer, relying on resource isolation (helper core or staging node), or time sharing (in-lined analytics or running asynchronously in a separate thread). A first topic we will address will focus on developing resource allocation strategies and algorithms the programmer can rely on to ensure an efficient collaboration between the simulation and the analytics to optimize resource usage. Parallel algorithms tailored for in-situ analytics also need to be investigated. In-situ processings inherit from the parallelization scale and data distribution adopted by the simulation, and must execute with minimal perturbations on the simulation execution (whose actual resource usage is difficult to know a priori). This specific context calls for algorithms that rely on adaptive parallelization patterns and data structures. Cache oblivious or cache adaptive parallel data structures coupled with work stealing load balancing strategies are probably a sound basis. Also notice that the limited budgets of memory and data movements targeted by in-situ processing can be somehow compensated by an abundance of compute capabilities. As demonstrated by some early works, this balance can lead to develop in-situ efficient algorithms relying on original strategies that are not relevant in a classical context. In-situ creates a tighter loop between the scientist and her/his simulation. As such, an in-situ framework needs to be flexible to let the user define and deploy its own set of analysis. A manageable flexibility requires to favor simplicity and understandability, while still enabling an efficient use of parallel resources. We will further investigate these issues relying on the FlowVR framework that we designed with these goals in mind. Given the importance of users in this context, to validate our different developments we will tightly collaborate with scientists of some application domains, like molecular dynamics or fluid simulation, to design, develop, deploy and assess in-situ analytics scenarios. In-situ analytics is a specific workload, that needs to be scheduled very close to the simulation, but not necessarily active during the full extent of the simulation execution, and that may require to access data from previous runs. The scenarios we will develop will thus also be used to evaluate batch scheduling policies developed in the team.

Data Aware Batch Scheduling

The most common batch scheduling policy is First Come First Served (FCFS) with backfilling (BF). BF enables to fill idle spaces with smaller jobs while keeping the original order of FCFS. More advanced algorithms are seldom used on production platforms due to the gap between theoretical models and practical systems. In practice the job execution times depend on their allocation (due to communication interferences and heterogeneity in both computation and communication), while theoretical models of parallel jobs usually consider jobs as black boxes with a fixed execution time. Though interesting and powerful, the synchronous PRAM model, delay model, LogP model and their variants (such as hierarchical delay), are ill-suited to large scale parallelism on platforms where the cost of moving data is significant and non uniform. Recent studies are still refining these models to take into account communication contentions accurately while remaining tractable enough to provide a useful tool for algorithm design. When looking at theoretical scheduling problems, the generally accepted goal is to provide polynomial algorithms. However, with millions of processing cores where every process and data transfer have to be individually scheduled, polynomial algorithms are prohibitive as soon as the largest exponent is two. The model of parallel tasks simplifies this problem by bundling many threads and communications into single boxes, either rigid, rectangular or malleable. Yet these models are again ill-adapted to heterogeneous platforms, as the running time depends on more than simply the number of allotted resources, and some of the basic underlying assumptions on the speed-up functions (such as concavity) are not often valid in practice.

We aim at studying these problems, both theoretically and through simulations. We expect to improve on the existing models (on power for example) and design new approximation algorithms with different objectives such as stretch, reliability, throughput or energy consumption, while keeping in focus the need for a very low polynomial complexity. Realistic simulations are required to take into account the impact of allocations and assess the real behavior of algorithms.

Empirical Studies of Large Scale

Platforms

Experiments in realistic context is critical to bridge the gap between theoretical algorithms and practical solutions.

But to experiment resource allocation strategies on large scale platforms is extremely challenging due to their constrained availability and their complexity. To circumvent this pitfall, we need to develop tools and methodologies for alternative empirical studies, from trace analysis, to simulation and emulation:

- **Traces** are a base tool to capture the behaviour of applications on production platforms. But, it is still a challenge to instrument platforms, in particular to trace data movements at memory, network and I/O levels, with an acceptable performance impact, even if the granularity required for batch scheduling is rather coarse (seconds to hours time scale). Trace analysis is expected to give valuable insights to define models encompassing complex behaviours like network topology sensitivity, network congestion and resource interferences. Traces are also fundamental to calibrate simulations or emulations as well as for training algorithms relying on learning techniques.
- **Emulation and simulation:** Emulation techniques will also be considered to replay data movements and I/O transfers. The Grid'5000 infrastructure dedicated to experimentations plays a crucial role here, and we will pursue our participation in its developments. But experimenting on real systems is costly and time consuming. More over the number of platform configurations that can be evaluated is particularly limited. The obvious approach is to consider simulations. One challenge for simulators is to be able to simulate accurately resource behaviours upon different workloads directly generated from collected traces and coarse grain job models. We will in particular work on interfacing our production batch scheduler OAR with the SimGrid simulator developed by Polaris and Avalon.

Forecasting Resource Availability

It is well-known that optimization processes are strongly related to the (precise) knowledge of the problem parameters. However, the evolution of HPC architectures, applications and computing platforms leads to an increasing complexity. As a consequence, more and more data are produced (for monitoring CPU usage, I/O traffic, informations about energy consumption, etc.), by both the job management system (the characteristics of the jobs to be executed and those that have already been executed) and by analytics at the application level (parameters, results and temporary results). It is crucial to adapt the job management system to deal with the bad effects of uncertainties, which may be catastrophic in large scale heterogeneous HPC platforms.

More precisely, determining efficient allocation and scheduling strategies that can deal with complex systems and adapt to their evolutions is a strategic and difficult challenge. We propose to study new methods for a better prediction of the characteristics of the jobs and their execution to improve the optimization process. In particular, the methods studied in the field of big data (supervised Machine Learning, SVM, learning to rank techniques, etc.) could and must be used to improve job scheduling in the new HPC platforms. A preliminary study has been done with the target of predicting the job running times (SC'2015). We are interested in extending the panel of parameters (including for instance some indicators for energy consumption) and in developing new methods on stochastic optimization.

At the application level, it also appears as important to predict resource usage (compute units, memory, network), relying on the temporal and spatial coherency that most numerical simulations exhibit, to better schedule in-situ analytics, I/Os and more generally data movements.

Relations industrielles et internationales

Our transfer strategy is twofold: make most of our algorithms available to the community through open source software, and develop partnerships with private companies through direct contracts or collaborative projects funded by national or european agencies.

- **Large companies.** We have developed long term collaborations with large groups, like the HPC division of BULL/ATOS, or the R&D scientific visualization team at the EDF power company. They already funded several PhDs. We develop with them state of the art solutions adapted to their specific problems. In particular we are collaborating with BULL/ATOS to contribute to the SLURM batch scheduler. SLURM is an open source job manager, widely deployed on HPC platforms and strongly supported by BULL/ATOS. We develop with BULL plugins to make our scheduling algorithms available to the broad community of SLURM users.
- **Small and medium-sized enterprises (SMEs).** Optimizing performance is a challenging problem for many SMEs that develop solutions for distributed computing. We will develop tight collaborations with some of them (TeamTo, ReactivIP, Stimergy, etc.), helping them to push forward their competitive edge.

