Poste d'ingénieur – 12 mois
Tolérance aux pannes dans la bibliothèque de communications réseau NewMadeleine

2025

Responsable:

- Alexandre DENIS (Inria, `Alexandre.Denis@inria.fr`),

Lieu: centre Inria de l'Université de Bordeaux, équipe Tadaam

Mots-clefs runtime, réseau, calcul haute-performance, tolérance aux pannes

Postuler en ligne:
`https://recrutement.inria.fr/public/classic/fr/offres/2025-08923`

Contexte

NUMPEX est le programme et équipement prioritaire de recherche (PEPR) pour l'Exascale en France. Cette initiative nationale vise à faire progresser la recherche scientifique et technologique dans le domaine du calcul à grande échelle et de l'informatique exascale

Ce poste s'inscrit dans le cadre du projet Exa-Soft du programme Numpex.

Dans le cadre du calcul haute-performance, les machines sont désormais très hétérogènes, équipées d'accélérateurs tels que les GPU ou les FPGA. Ces différentes unités se programment avec des paradigmes différents, et pour ajouter à la complexité, il faut gérer les transferts de données entre les unités, l'ordonnancement, et l'équilibrage de charge.

Pour tirer profit d'une telle plate-forme, l'équipe STORM a proposé le support d'exécution StarPU [4] qui gère ces problématiques de manière

générique et indépendante de l'application. Pour des grappes de calcul – plusieurs noeuds interconnectés par un réseau d'interconnexion – StarPU confie [3] l'exploitation du réseau à une bibliothèque de communication qui implémente l'interface MPI, d'utilisation standard en HPC.

Par ailleurs, l'équipe TADAAM développe la bibliothèque de communication NewMadeleine [1, 5], dont l'originalité est d'appliquer une stratégie d'optimisation à la volée sur les flux de communications issus des différents threads, en assurant une progression [8, 11] asynchrone en tâche de fond. Elle repose essentiellement sur le principe de programmation événementielle et de messages actifs, ce qui permet le déroulement des communications sans intervention de l'application.

Les besoins de StarPU en terme de communications ne sont pas les mêmes que ceux d'une application MPI classique, notamment en matière d'irrégularité, de réactivité, de multi-threading, de nombre de requêtes actives simultanément, alors qu'ils correspondent parfaitement au cahier des charges de NewMadeleine. Un portage se StarPU sur l'interface native de NewMadeleine a été réalisé [6] pour en exploiter au mieux les propriétés de progression, de passage à l'échelle [9], ainsi qu'une intégration spécifique [10] des opérations collectives.

Sur les supercalculateurs, le temps moyen entre pannes (MTBF) est typiquement de moins d'un jour. Les applications qui tournent à grande échelle doivent donc prendre en compte l'éventualité de fautes ou la perte d'un noeud. Les stratégies classiques pour gérer la perte d'un noeud est le *checkpoint*, qui est désormais disponible dans StarPU. Cependant, la gestion de la perte d'un noeud n'est pas encore supportée dans NewMadeleine.

Objectif

L'objectif de ce poste d'ingénieur est d'ajouter le support de la tolérance aux pannes dans la bibliothèque de communication NewMadeleine. Concrètement, il s'agira d'implémenter dans NewMadeleine la norme ULFM [2, 7].

Activités

Concrètement, les activités à mener seront les suivantes:

Détection des fautes. Il s'agira dans un premier temps d'ajouter une vraie détection des fautes dans les drivers réseau de NewMadeleine. Dans l'actuelle, cette détection est partielle, et prend souvent la forme d'un abandon plutôt que d'une remontée complète de la situation d'erreur.

État dégradé. Après avoir détecté les fautes, il faudra mettre en place les mécanismes nécessaires pour que NewMadeleine puisse continuer à fonctionner correctement sans utiliser les liens fautifs. Il s'agit essentiellement d'adapter les stratégies de polling et d'adapter les conditions de soumission de nouvelles requêtes.

Remontées des fautes. Ensuite, il sera nécessaire de remonter à l'utilisateur les situations d'erreur, en implémentant les codes d'erreurs et les nouvelles fonctions proposées par ULFM.

Récupération. Enfin, pour récupérer totalement après une faute, il faut être capable de lancer de nouveaux processus et de les connecter à la session courante au travers de l'appel MPI `MPI_Comm_spawn`. Cette fonction n'est actuellement pas implémentée dans NewMadeleine. Il sera donc nécessaire d'en proposer une implémentation.

Tests avec StarPU. Une fois que ces mécanismes seront en place, les tests se feront avec le cas d'usage envisagé dans Exa-Soft, à savoir au travers du runtime StarPU.

Commentaires

Les développements à réaliser se feront dans la bibliothèque NewMadeleine disponible en open source et se font en langage C. Il serait souhaitable qu'ils soient réalisés par une personne à l'aise en programmation réseau et système.

References

[1] Newmadeleine: An optimizing communication library for high-performance networks. `https://pm2.gitlabpages.inria.fr/newmadeleine/`.

[2] User level failure mitigation. `https://fault-tolerance.org/`.

[3] Emmanuel Agullo, Olivier Aumage, Mathieu Faverge, Nathalie Furmento, Florent Pruvost, Marc Sergent, and Samuel Thibault. Achieving High Performance on Supercomputers with a Sequential Task-based Programming Model. *IEEE Transactions on Parallel and Distributed Systems*, 2017.

[4] Cédric Augonnet, Samuel Thibault, Raymond Namyst, and Pierre-André Wacrenier. StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. *Concurrency and Computation: Practice and Experience, Special Issue: Euro-Par 2009*, 23:187–198, February 2011.

[5] Olivier Aumage, Elisabeth Brunet, Nathalie Furmento, and Raymond Namyst. NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks. In *Workshop on Communication Architecture for Clusters (CAC 2007), workshop held in conjunction with IPDPS 2007*, Long Beach, California, United States, March 2007.

[6] Guillaume Beauchamp. Portage de StarPU sur la bibliothèque de communication NewMadeleine. Master's thesis, Universite Bordeaux, September 2017.

[7] Wesley Bland, Aurelien Bouteiller, Thomas Herault, George Bosilca, and Jack Dongarra. Post-failure recovery of mpi communication capability: Design and rationale. *The International Journal of High Performance Computing Applications*, 27(3):244–254, 2013.

[8] Alexandre Denis. pioman: a pthread-based Multithreaded Communication Engine. In *Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Turku, Finland, March 2015.

[9] Alexandre Denis. Scalability of the NewMadeleine Communication Library for Large Numbers of MPI Point-to-Point Requests. In *CCGrid 2019 - 19th Annual IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing*, Larnaca, Cyprus, May 2019.

[10] Alexandre Denis, Emmanuel Jeannot, Philippe Swartvagher, and Samuel Thibault. Using Dynamic Broadcasts to improve Task-Based Runtime Performances. In *Euro-Par 2020*, Euro-Par 2020, Warsaw, Poland, August 2020. Rzadca and Malawski, Springer.

[11] Alexandre Denis and François Trahay. MPI Overlap: Benchmark and Analysis. In *International Conference on Parallel Processing*, 45th International Conference on Parallel Processing, Philadelphia, United States, August 2016.