

Gysela5D

5D GYrokinetic SEmi-LAgrangian Code for
Magnetically Confined Fusion Plasma Simulation

Plan

- ❖ Introduction
- ❖ Contexte
 - Gysela5D
 - FTI
- ❖ Objectifs
- ❖ Poincare
 - Description de la machine
 - Figure 1 : Weak Scaling Sans Checkpoint et Avec Checkpoint Diagnostics tous les 1 itérations
 - Figure 2 : Weak Scaling Sans Checkpoint et Avec Checkpoint Diagnostics tous les 8 itérations
 - Figure 3 : Temps de checkpoint avec diagnostics tous les 1 itérations vs tous les 8 itérations.
- ❖ Curie
 - Description de la machine
 - Figure 1 : Weak Scaling Sans Checkpoint et Avec Checkpoint Diagnostics tous les 1 itérations
 - Figure 2 :Weak Scaling Sans Checkpoint et Avec Checkpoint Diagnostics tous les 8 itérations
 - Figure 3 : Temps de checkpoint avec diagnostics tous les 1 itérations vs tous les 8 itérations.
- ❖ Conclusion & perspectives.

I- Introduction

Avec le déploiement d'une infrastructure matérielle pour le calcul intensif, aussi bien au niveau national avec la mise en place de GENCI qu'au niveau Européen dans le cadre du projet PRACE, les chercheurs ont maintenant accès à un parc de supercalculateurs de classe mondiale. De tels supercalculateurs devraient rendre possible des avancées majeures dans les différents domaines scientifiques mais aussi contribuer à des avancées importantes ayant des conséquences en terme de compétitivité industrielle. Atteindre ces objectifs impose de disposer d'applications conçues et optimisées pour un très fort degré de parallélisme. De plus, la compréhension des masses de données produites par les grandes simulations nécessite des outils de post-traitement et de visualisation adaptés. La complexité des architectures de systèmes de la classe Petaflop/s et celles des simulations visées représentent un véritable défi. S'y attaquer impose de constituer des équipes pluridisciplinaires comprenant des informaticiens du HPC (High Performance Computing), des spécialistes du traitement numérique, des mathématiciens appliqués et des experts des domaines applicatifs visés.

La maison de la simulation est un laboratoire regroupant cinq partenaires (CEA, CNRS, INRIA, Université d'Orsay et l'Université de Versailles St Quentin) qui a pour objectif d'accompagner, de soutenir et de stimuler les communautés scientifiques afin de tirer le meilleur parti des supercalculateurs, en particulier déployés dans le cadre de GENCI et du projet Européen PRACE.

II- Contexte

- **Gysela5D :**

C'est la première principale simulation de la fusion nucléaire du réacteur ITER, qui permet de simuler les perturbations qui se développent dans le Plasma. C'est pourquoi ça nécessite l'utilisation d'un super calculateur pour l'exécution de cette simulation.

L'utilisation d'un tel nombre de cœurs sur beaucoup de temps pose des problèmes de pannes de certains composants du calculateur même si chaque cœur de calcul est fiable.

Aujourd'hui, les plus gros calculateurs subissent des pannes que ce soit matérielles ou logicielles jusqu'à plusieurs fois par semaine.

Dans ce cas, l'exécution de grande simulation (plusieurs jours sur plusieurs dizaines de milliers de cœurs) est quasiment assuré d'être interrompue par une panne et qui risque de ne jamais pouvoir terminer.

Une solution à ce problème consiste à sauvegarder régulièrement l'état de simulation sur disque pour pouvoir la reprendre en cas de panne, on parle de checkpoint/restart.

Il existe 2 types de checkpoint (sauvegarde) qui sont déjà implémenté sur Gysela:

- ⑩ Synchrones : Arrêt momentanément pour faire du checkpoint, puis reprise les calculs.
- ⑩ Asynchrones : checkpoint sur la RAM, puis reprise du calcul et un thread écrit sur le PFS.

Cette sauvegarde pose des problèmes de performances liés à la grande taille des données à écrire (jusqu'à plusieurs téraoctets). L'augmentation de la taille des machines va rendre ce problème encore plus critique et le risque est de passer plus de temps à sauvegarder qu'à calculer.

Une solution est possible d'intégrer la bibliothèque FTI au code Gysela5D.

- **FTI : Fault Tolerance Interface**

FTI est une bibliothèque qui offre de multiples fonctionnalités de tolérance de panne grâce à un interface facile à utiliser son objectif de FTI c'est d'améliorer la fiabilité des supercalculateurs en s'appuyant sur les caractéristiques matérielles sous-jacentes de la manière la plus efficace, tout en offrant une interface très simple pour l'utilisateur. FTI est écrit en C et il cible les applications de calcul haute performance utilisant MPI.

Avec FTI on n'a pas besoin de traiter les fichiers, les noms de fichiers, métadonnées, etc. FTI fait tout. Il faut juste donner un pointeur vers la structure de données qu'on souhaite sauvegarder et donner la taille des données.

Cette bibliothèque introduit un nouveau principe de sauvegarde multi-niveaux, quatre niveaux chaque un a un niveau de tolérance plus fiable avec un coût pour chaque niveaux. Mais un prérequis de cette bibliothèque est des disques SSD locaux pour les processus.

Il utilise un fichier de configuration pour définir toutes ses options. Par exemple, la fréquence des check-points chaque niveau, les emplacements des check-points, verbosité, etc le tout sans modifier le code source et / ou de recompiler l'application. Il suffit de modifier le fichier de configuration FTI et relancer l'application. Ce fichier de configuration est passé comme argument, donc on peut lancer plusieurs exécutions avec différents fichiers de configuration FTI pour réaliser des comportements différents.

Un processus est dédié à l' FTI sur chaque nœud qui s'occupe des check-points.

Finalement FTI efface automatiquement les points de contrôle précédents et leurs métadonnées après un poste de contrôle a été pris. Lorsque l'exécution est terminée, toutes les données des points de contrôle et les métadonnées sont nettoyées par FTI et il laisse les répertoires exactement dans le même état qu'il était avant le début de l'exécution.

Les niveau de FTI :

L1 : Dans ce niveau chaque processus écrit son check-point sur le SSD local donc la vitesse de l'écriture est constant quel que soit le nombre de nœuds

En cas de panne matériel on ne peut pas récupérer.

L2 : FTI détecte le nœud dans lequel chaque rang MPI est situé. Cela nous permet de sélectionner les processus partenaires et les groupes de codage situés dans les ganglions éloignés, ce qui augmente considérablement la fiabilité du groupe. Après FTI échange les check-points locaux entre deux partenaires. (FTI thread)

L3 : Échange des check-points entre les processus d'un même groupe avec un encodage Reed-Solomon. (FTI thread)

L4 : Écriture du check-point dans le PFS. (FTI thread)

III- Présentation des objectifs :

Evaluation de performances de trois approches pour les écritures de checkpoint dans Gysela.

- Checkpoint synchrone.
- Checkpoint asynchrone.
- Checkpoint avec FTI.

Analyse et modélisation du comportement de chacune des approches sur une variété de supercalculateur la machine Curie et Poincaré. Comporte des SSD et est donc bien adaptée à l'utilisation de la bibliothèque FTI.

Déterminer les critères pertinents pour le choix de l'approche à utiliser sur chaque machine disponible ainsi que sur celles à venir.

IV- Poincare

Description de la machine Poincare :

- Calculateur IBM, principalement composé de serveurs iDataPlex dx360 M4
- 92 Noeuds de calculs : 16coeurs par nœud, 32Go de mémoire par nœud.
- 4 Noeuds GPU : 4 processeurs, 32 Go par nœuds.
- GPFS (General parallel file system) : partagé par l'ensemble des nœuds
- /gpfs1l : 7 To, contient les $\{HOME\}$ ainsi que l'environnement logiciel partagé
- /gpfsdata : 37 To, contient un répertoire de travail pour les grosses données

Analyse des performances :

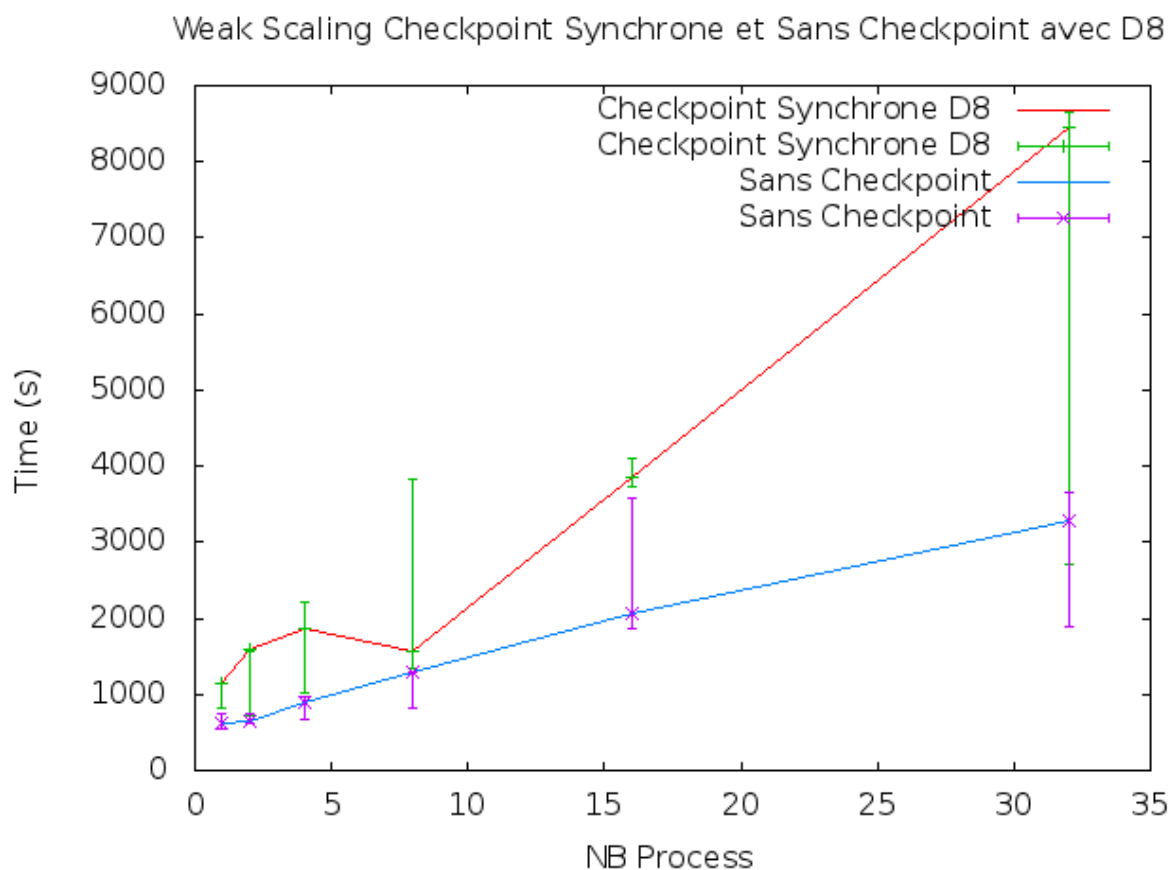


Figure : Weak Scaling Avec checkpoint synchrone et Weak Scaling sans checkpoint

La courbe en rouge : Weak Scaling avec checkpoint synchrone.

La courbe en vert : Weak Scaling Sans checkpoint.

L'expérience est faite pour voir la limite ou peut atteindre du PFS

On remarque que les 2 courbes croient en augmentant le nombre de processus, et on remarque aussi que l'intervalle d'erreur est très grand quand on est à 32 ou 8 processus, cela est dû à des perturbations qui s'est produite au calculateur notamment le cas où la bande passante du PFS est saturé...

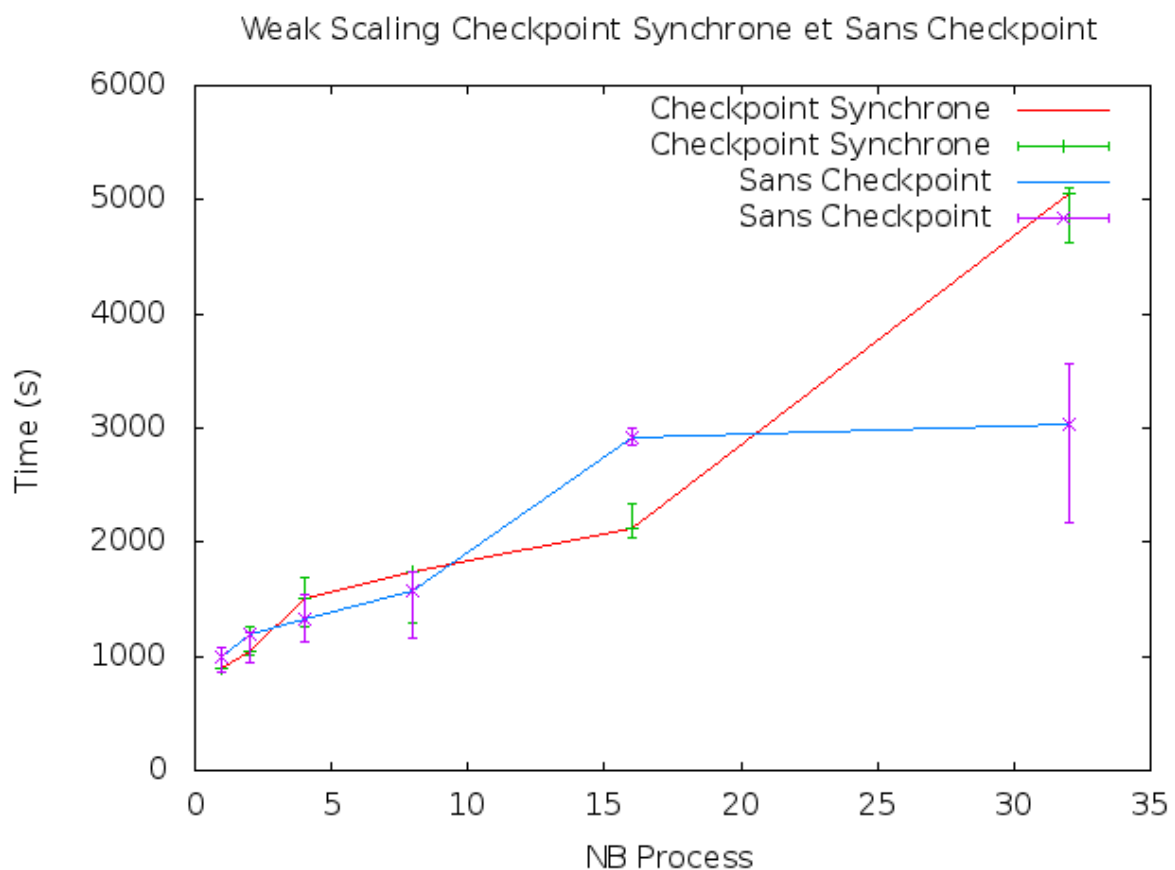


Figure : Weak Scaling avec checkpoint et weak scaling sans checkpoint avec plusieurs diagnostics.

Ces 2 courbes sont relatives au temps de weak scaling avec checkpoint et sans checkpoint avec plusieurs diagnostics sur la machine poicare.

En bleu : Weak Scaling Sans Checkpoint avec beaucoup de diagnostics.

En rouge : Weak Scaling avec Checkpoint avec beaucoup de diagnostics.

L'expérience est faite pour voir l'impact des diagnostics sur les checkpoint.

— On observe bien que les mesures sur poicare n'est pas stables, vu que la moitié des points présente un temps de checkpoint négatifs, donc dans ce cas on ne peut pas dire si les diagnostics ont influe sur les checkpoint ou non.

=> Ces perturbations sont dû à plusieurs raisons :

* Le nombre élevé des jobs soumis peuvent dégrader les performances.

* Le nombre des processus qui écrivent sur le PFS en même temps peuvent saturer la bande passante qui relie les processeurs et les PFS.

V- Curie

Description de la machine Curie

Le supercalculateur Curie, appartenant à Genci et hébergé dans les locaux du CEA au TGCC, est la première machine (Tier0) ouverte aux scientifiques européens dans le cadre de la participation française aux infrastructures de recherche PRACE.

CURIE propose 3 différents types de ressources de calcul basé sur une architecture x86-64, s'adressant à un large panel de challenges scientifiques, et offrant une puissance de calcul crête globale de 2Pflops.

Spécifications des Fat nodes :

— Machine

360 nœuds S6010 bullx : Pour chaque nœud 4 processeurs octo-cœurs Intel® Xeon®, 128 Go, 1 disque local de 2To. Ce qui nous donne 105 téraflops crête.

— Processeurs

1440 processeurs octo-cœurs Intel® Nehalem-EX X7560 cadencés à 2.26 GHz soit 11 520 cœurs.

Spécifications des noeuds hybrides :

16 châssis bullx B équipés chacun de 9 lames hybrides GPUs B505 2 Intel® Westmere® 2.66 GHz/ 2 Nvidia M2090 T20A, total de 288 processeurs Intel® + 288 processeurs Nvidia. ce qui nous donne plus de 192 téraflops crête.

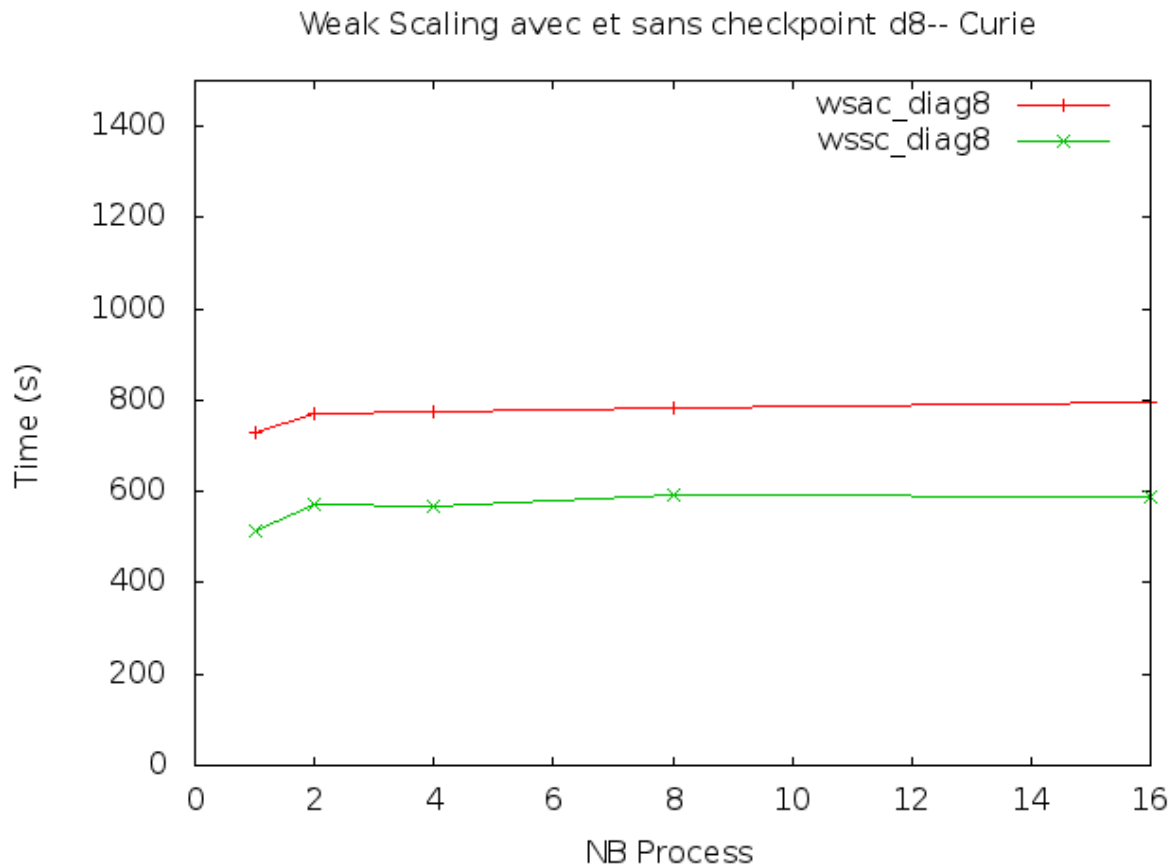
Spécifications de Curie Thin nodes

— Machine

5040 nœuds B510 bullx : Pour chaque nœud 2 processeurs octo-cœurs Intel® Sandy Bridge EP (E5-2680) 2.7 GHz, 64 Go, 1 disque local SSD.

— Processeurs

10080 processeurs octo-cœurs Intel® Xeon® nouvelle génération soit 80640 coeurs.



Analyse de performance :

Figure : Weak Scaling Avec et Sans Checkpoint sur la machine Curie.

La courbe en rouge représente le temps du Weak Scaling avec checkpoint synchrone.

La courbe en vert représente le temps du Weak Scaling sans checkpoint

Cette expérience faite pour voir la limite que le PFS peut supporter, on conclut que les 2 courbes sont stables jusqu'à 16 processus, donc jusqu'à 16 processus on n'est pas à la limite du pfs.

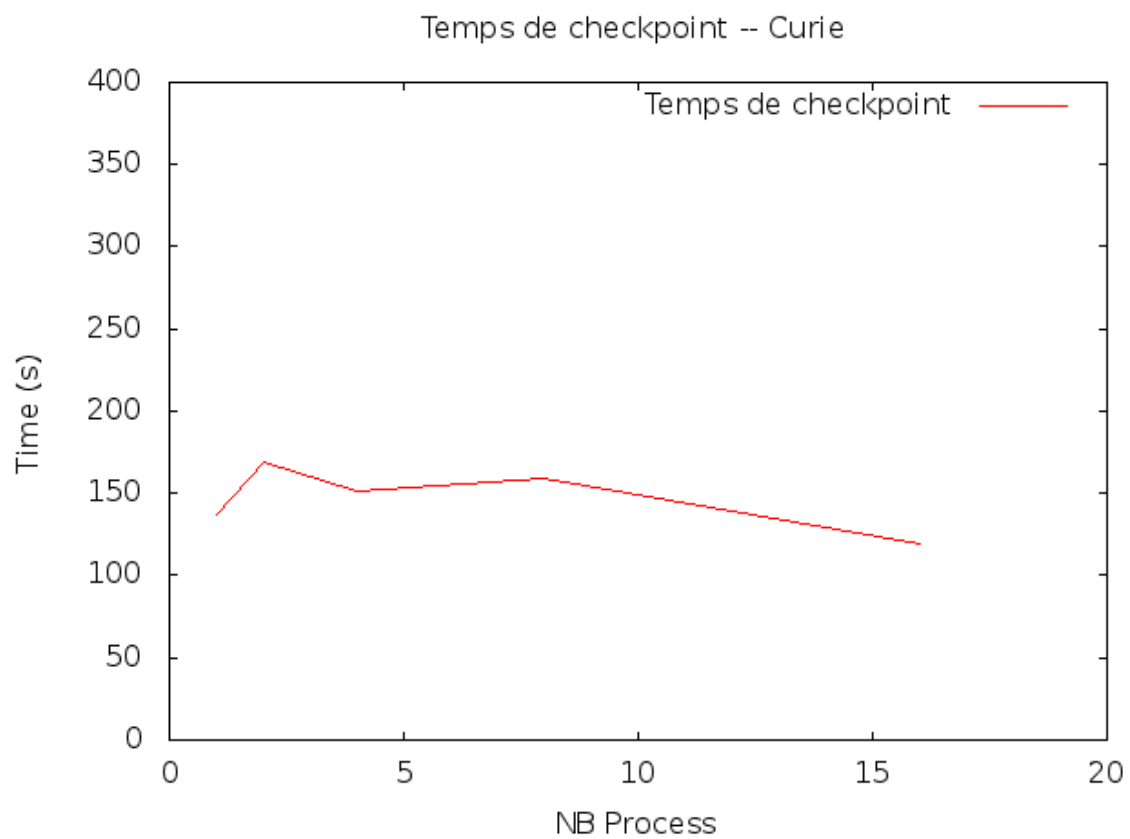


Figure : temps de checkpoint en fonction du nombre de processus.

Le temps de checkpoint est la différence entre le Weak Scaling avec checkpoint et le Weak Scaling sans checkpoint.

On remarque que la courbe est un peu stable sur les 16 premiers processus, et que le temps pour faire des checkpoint est de l'ordre de 150 seconde.

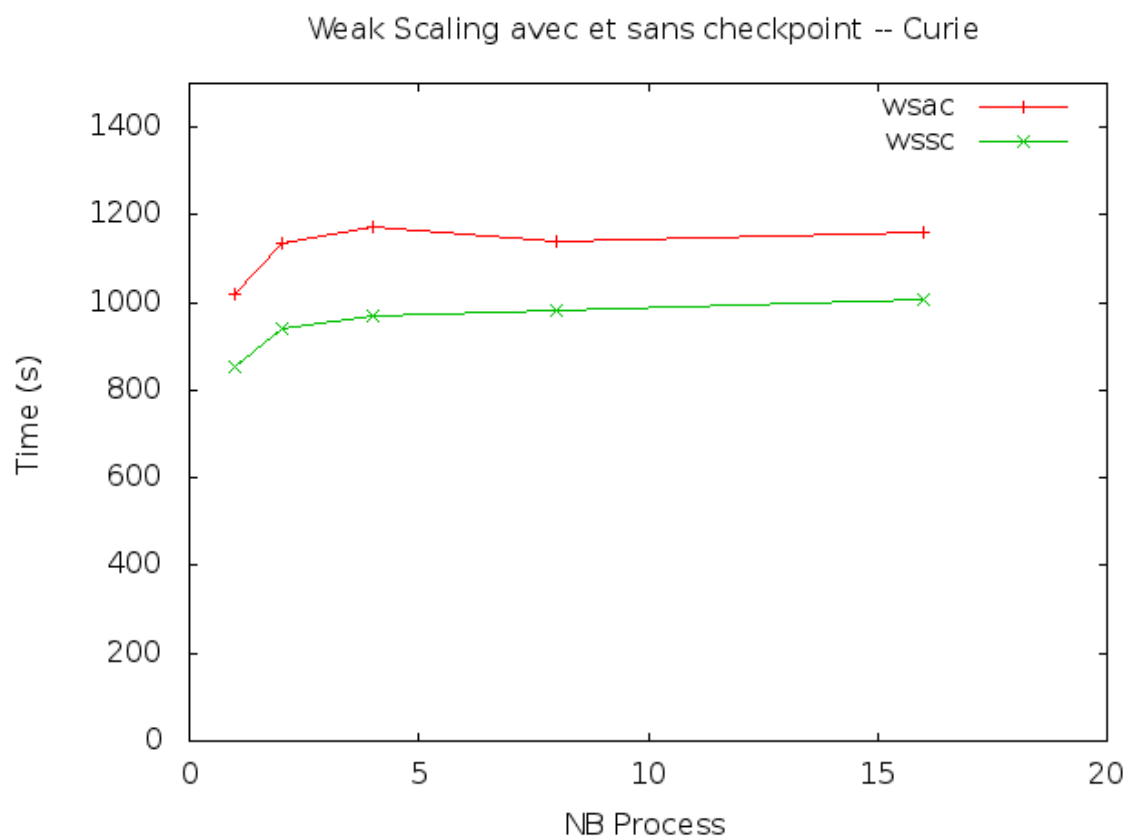


Figure : Weak Scaling avec checkpoint synchrone et Weak Scaling sans checkpoint avec plusieurs diagnostics.

Courbe en vert : Weak Scaling sans checkpoint et avec plusieurs diagnostics.

Courbe en rouge : Weak Scaling avec checkpoint et avec plusieurs diagnostics

Ici on a augmenté le nombre des diagnostics pour voir leurs impacts sur les checkpoints.

Les 2 courbes sont stables jusqu'à 16 processus.

Comparant cet expérience par rapport à celle sur poicare, on peut conclure que sur Curie jusqu'à 16 processus on n'atteint pas la limite du PFS alors que sur poicare, on ne peut rien conclure vu les perturbations sur le calculateur.

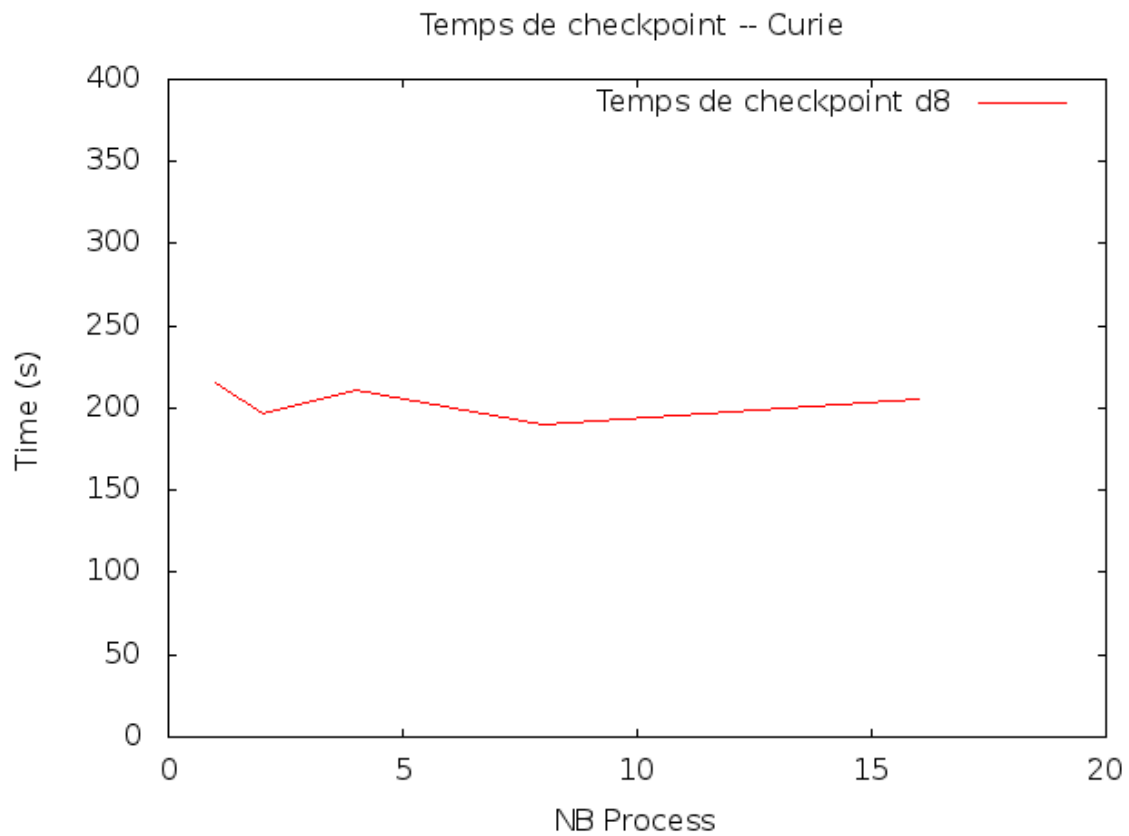


Figure : Temps de checkpoint avec plusieurs diagnostics.

La figure représente le temps de checkpoint en Weak Scaling avec plusieurs diagnostics

On remarque que la courbe est un peu stable sur les 16 premiers processus, et que le temps pour faire des checkpoints est de l'ordre de 200 seconde.

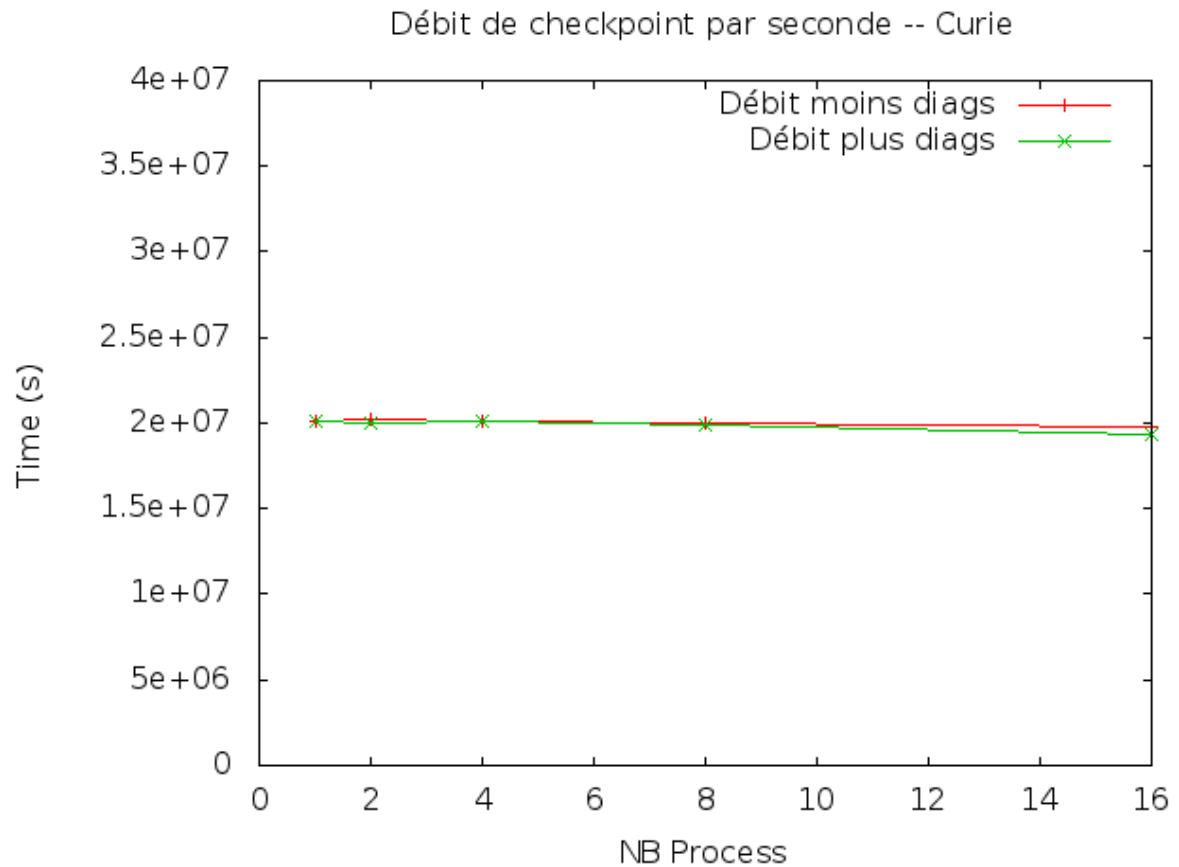


Figure : Débit d'écriture sur le PFS de la machine Curie.

Cette figure représente la vitesse d'écriture sur disque pour un processus par fichier.

Sur les x : nombre de processus

Sur les y : vitesse de checkpoint en octet/s

Vitesse d'écriture =

*(taille de fichier * nombre de fichiers restart) / Temps d'écriture.*

Curie utilise des "routeurs Lustre" pour accéder au système de fichier global Lustre au travers d'un réseau infiniband de 200Go/s de bande passante.

VI- Conclusion & Perspective

- Mesures sur poincare ne donne pas des résultats pertinents.
- Résultats sur Curie c'est très stable, et que jusqu'à 16 noeuds ce n'est pas suffisant pour atteindre la limite du PFS.
- Implémentation de FTI n'est pas encore débogué pour l'évaluer.
- Evaluer le mode asynchrone.