

TSUBAME 共同利用 平成 27 年度 学術利用 成果報告書

利用課題名 スペアノードを用いた実行継続手法の通信性能評価  
英文: Performance Evaluation of Spare Node Substitution

利用課題責任者 堀敦史  
First name Surname Atsushi Hori

所属 理化学研究所 計算科学研究機構  
Affiliation RIKEN Advanced Institute for Computational Science

邦文抄録(300 字程度)

ハードウェア規模の増大するエクサスケール環境では、故障発生率が増加するため、耐故障性の確保が重要である。我々はステンシル計算に対する故障復帰手法として、スペアノードを用いた手法の検討と評価を進めている。スペアノードを用いた実行継続では、アプリケーションプログラムへの大きな変更無しにロードバランスを維持した実行継続が可能。一方、通信経路が変化することにより、通信性能が低下することを確認している。これまで我々は Mesh/Torus ネットワークトポロジにおいて、スライド手法での実行継続後の通信性能の評価を行ってきたが、本課題では、Fat Tree ネットワークトポロジを持つ TSUBAME2.5 を用いて評価を行った。

英文抄録(100 words 程度)

With the increasing fault rate on high-end supercomputers, the topic of fault tolerance has been gathering attention. We proposed failure recovery method using spare node for stencil application named “sliding methods”. The sliding methods preserve load balancing and communication pattern, however, communication performance is degraded on Mesh/Torus network topology. In this project, we evaluated the sliding methods on TSUBAME2.5 having Fat-Tree network topology.

*Keywords:* 5つ程度

fault tolerance, fault mitigation, spare node, communication performance

背景と目的

エクサスケール環境では構成するハードウェアが増大するために故障発生率が増加し、耐故障性の確保が重要な課題となる。現在広く利用されているチェックポイント/リスタートによる実行継続は、規模の増大するエクサスケール環境では I/O 時間が長期化するために現実的ではないと言われており、ユーザレベルでの耐故障性の確保が注目されている。

ユーザレベルでの故障対策を実現するために、プログラム内での故障検知と挙動の定義が可能な ULFM のようなライブラリの研究が進められている。しかしその一方で、実際にその研究成果を用いてどのように実行を継続することが効率的であるかについては、あまり研究がなされていないというのが現状である。

そこで我々は、ULFM のようなユーザレベルでノード故障への対応を可能とするライブラリを用いた効率的な故障復帰手法として、ステンシル計算を対象とし、ス

ペアノードを利用したスライド手法の提案を進めてきた。スペアノードを利用することで、プログラムへの大きな変更をすることなくロードバランスと論理的な通信パターンを保った実行継続が可能となる。しかしその一方で、ネットワーク的に離れたスペアノードを利用することで、物理的な通信経路が変化し、通信衝突の発生による通信性能の低下が生まれることが判明している。これまでに我々は、Mesh/Torus ネットワークトポロジ上でのスライド手法について、その有効性を検討するために通信性能の評価を進めてきた。

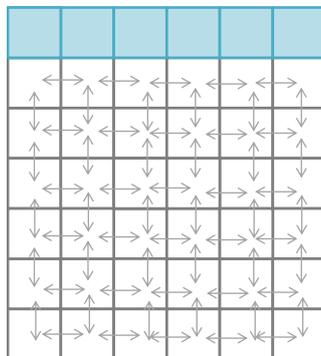
本プロジェクトでは、Mesh/Torus と並び並列計算機で利用されるネットワークトポロジである Fat Tree ネットワークトポロジを持つ TSUBAME2.5 を利用し、Fat Tree ネットワークトポロジでのスライド手法の有用性を検証するため、実行継続後の通信性能を評価した。Thin ノードを最大 343 ノード用い、スライド手法による実行継続後の通信性能を測定した結果、

TSUBAME2.5 ではどの手法を用いた場合にも故障前からの通信性能の低下は発生しないことを確認した。

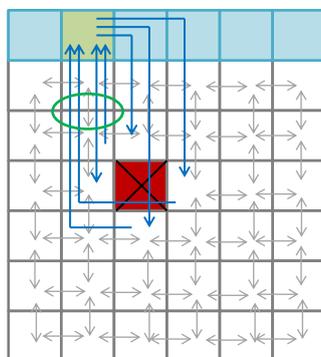
## 概要

本課題では、我々が検討と評価を進めているスペアノードを用いた実行継続手法について、Fat Tree ネットワークトポロジを持つ TSUBAME2.5 を用いて評価を行った。

スペアノードを用いた実行継続では、アプリケーションプログラムへの大きな変更無しにロードバランスを維持した実行継続が可能で、通信経路が変化することにより、通信性能が低下することを確認している。図 1 は、2D Mesh ネットワーク上での 5 点ステンシル計算プログラムの、スペアノードを利用した故障復帰例であり、最上行のノードがスペアノードである。故障前の状態(a)では、通信は衝突せずに行われているが、故障後にスペアノードを用いて実行継続した例の(b)では、通信経路が変更となり、丸で囲んだ部分において通信衝突が発生しており、通信性能が低下する可能性がある。



(a) 故障前の通信



(b) 故障復帰後の通信

図 1 スペアノードを利用した故障復帰

故障復帰後の通信衝突の発生は、スペアノードをど

のように利用し故障ノードと代替するか、その方式に大きく依存する。我々はスペアノードの利用方式として、スライド手法を提案している。以下に基本となる  $nD$  Sliding 手法と、その組み合わせである Hybrid Sliding 手法について簡単に示す。

(手法の図を入れる)

スペアノードを利用した故障復帰は、故障の前後でノードの物理的な配置が変更されることによって通信経路が変化する。Mesh/Torus ネットワークトポロジ上でのステンシル計算に適用した場合、故障前には存在しなかった通信経路の共有が発生し、メッセージ衝突により通信性能が低下する可能性がある。一方、スペアノードを利用したスライド手法を Fat Tree ネットワークトポロジに適用した場合、故障前後でノードの配置は変更されるが、トポロジの特性上通信経路の衝突は発生しない。そのため Mesh/Torus ネットワークトポロジ上でスライド手法を適用した場合とは異なり、スペアノードを利用した故障復帰後も通信性能は変化しないと考えられる。また、この特徴は代替先ノードの配置によらず保たれるため、どの Sliding 方式を用いたとしても性能に差は現れないと考えられる。そこで、TSUBAME2.5 上でスライド手法を適用した場合の故障復帰後の通信性能について、通信実験を行い評価した。

## 結果および考察

TSUBAME2.5 を用いて、スライド手法による実行継続後の通信性能評価を行った。評価には  $4 \times 4 \times 4$ 、 $5 \times 5 \times 5$ 、 $7 \times 7 \times 7$  の三次元形状のジョブを利用し、スペアノードは  $3D(2,1)$ 方式と呼んでいる、X または Y 座標が最大となる 2 平面に確保する形をとった。なお、ジョブの形状による差異がほぼ見られなかったため、以降の評価結果は  $5 \times 5 \times 5$  のものだけを示す。評価対象の通信パターンとして 7 点ステンシル計算における通信パターンと Barrier、Allreduce の 2 種類の集団通信を、実行継続手法として 0D 及び Hybrid の 2 手法を利用した。また、Mesh/Torus ネットワークトポロジとの比較を行うため、同様の実験を京でも行い比較した。

実験では、TSUBAME2.5 の予約キューを用い、連

続したノードを確保し利用した。また、ログインノードやファイルシステムの接続されている 1st Rail ネットワークは利用せず、2nd Rail のみを用いて実験を行った。なお、実際に故障を発生させることは不可能であるため、故障後のプロセス配置をシミュレーションにより求め、その配置を実環境上に再現し、通信性能の測定を行う。

京においては 3 次元ジョブを 6D メッシュ/トラスネットワークポロジに隣接関係を保持したランクマッピングとなる形で投入している。一方 TSUBAME2.5 ではそのようなマッピングはないため、ランク番号は京のものを踏襲し、X→Y→Z の順に座標を変化させたものを利用した。このため、TSUBAME2.5 ではスペアノードは不連続なノードに確保されることとなる。

図 2 は、4MiB の 7 点ステンシル通信の性能測定結果である。横軸は故障ノード数、縦軸は故障前の通信時間を基準とした通信時間比であり、故障数毎にシミュレーションで得た 20 パターンのランク配置で通信性能の測定を行い、その最悪値を示している。TSUBAME2.5 では故障後の通信時間の増加が最大でも 1.2 倍程度となり、京の 4.1 倍を大きく下回った。これは、京とは異なり衝突によるペナルティがほぼ発生しなかったためと考えられる。また、京では特に故障数が少ない場合に Hybrid 手法が高性能を示していたが、TSUBAME2.5 の場合はどちらの手法でもほぼ同性能を示しており、手法の違いによる性能差は無いと言える。

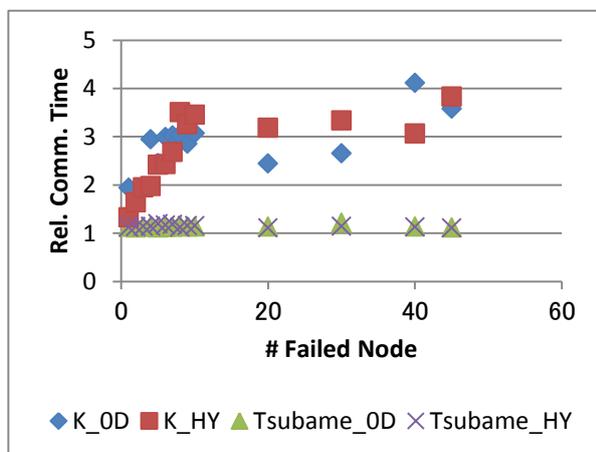


図 2 7 点ステンシル(4MiB)の性能

図 3 および図 4 は、それぞれ Barrier と Allreduce について同様に測定した結果である。こちらもステンシル通信の場合と同じく、どちらの手法を利用した場合も TSUBAME2.5 では故障復帰後の通信性能の低下はほぼ見られなかった。

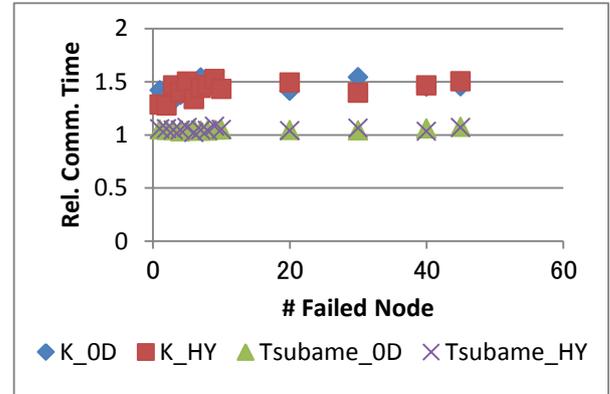


図 3 Barrier の性能

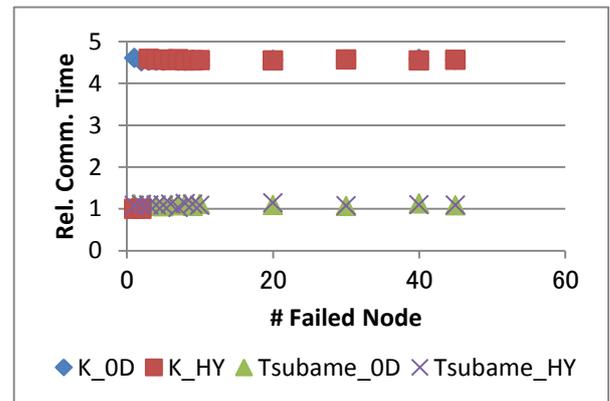


図 4 Allreduce(64KiB)の性能

ここまでの実験では、スペアノードを不連続で確保して測定をしたが、TSUBAME2.5 は Fat Tree 接続であるため、スペアノードをどこに確保した場合でも故障復帰後の性能に影響は無いと考えられる。そこで、スペアノードを連続する形でまとめて確保した場合の性能を測定し、比較と検証を行った。

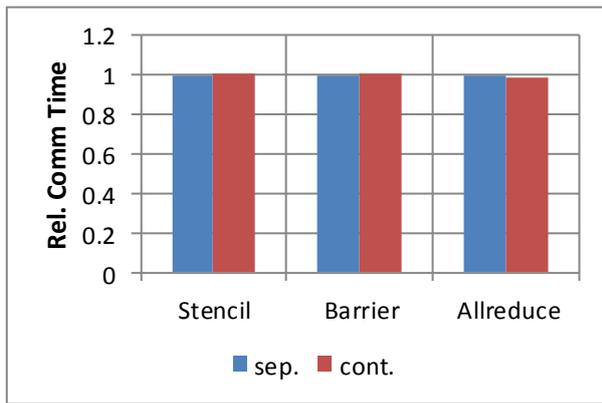


図 5 は、スペアノードの確保方式が非連続(sep.)の場合と連続(cont.)の場合とで、故障前の通信性能に差があるか実験した結果である。縦軸は非連続で確保した状態での通信性能を基準とした通信時間比である。結果から分かる通り、スペアノードの確保方式による性能差は見られなかった。

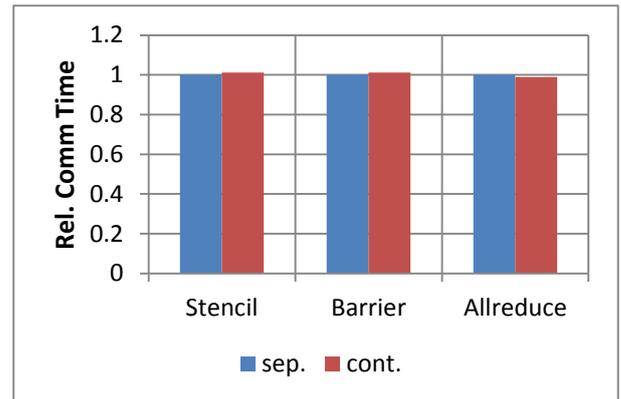


図 5 スペアノード確保方式による性能差

また、

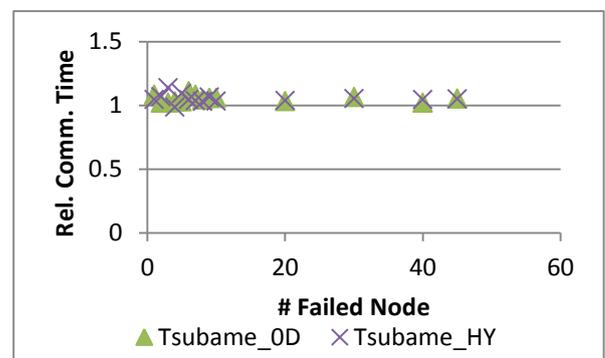
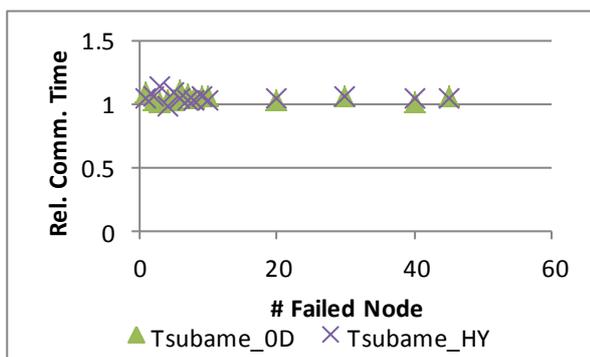


図 6 7点ステンシル(4MiB)の性能・スペア連続確保

図 6 に、スペアノードを連続で確保した上で、7点ステンシル通信パターンに対してスライド手法による故障復帰を適用した場合の通信性能を示す。結果を見て分かる通り、非連続でスペアノードを確保していた

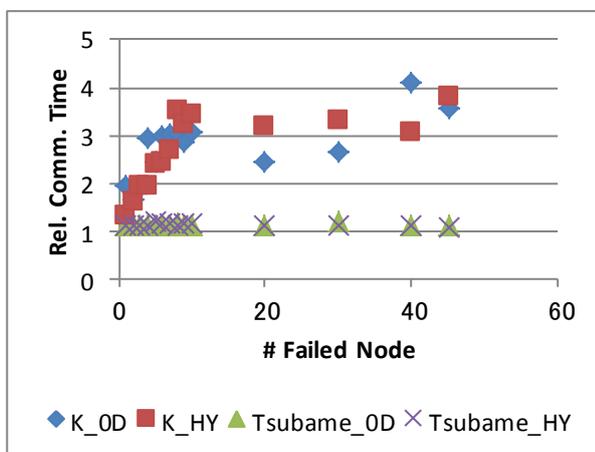


図 2 の場合とほぼ同様の性能を示しており、スペアノードの確保方式による性能差は出なかった。

#### まとめ、今後の課題

本プロジェクトでは、故障復帰後の実行継続手法として我々が提案している、スペアノードを用いたスライド手法について、Fat Tree ネットワークトポロジをもつ Tsubame2.5 を用い、故障復帰後の通信性能の評価を行った。実験の結果、Tsubame2.5 ではスペアノードを利用した故障復帰後には Fat Tree ネットワークの特性上新たな通信衝突は発生せず、通信性能はほぼ低下しないことを確認した。また、この特徴はスペアノードの利用方式によらず、どのスライド手法を利用した場合でも差異の無いことを確認した。また、スペアノードを確保する位置による影響を調査するため、連続と不連続の二つのパターンでスペアノードを確保し、その差異を確認したところ、方式による性能差は見られなかった。

今後の並列大規模環境を見据え、様々なトポロジでの故障対策について考慮するために、今回は Mesh/Torus と同様に HPC 環境で広く用いられる Fat

Treeトポロジでのスライド手法の評価を行った。今後は Dragonfly 等のネットワークについても性能評価を進める予定である。