Vol.2012-ARC-202 No.3
Vol.2012-HPC-137 No.3
2012/12/13

# TSUBAME2.0におけるMulti-rail InfiniBandネットワークの性能評価

野村 哲弘[1,2,a)]　遠藤 敏夫[1]　松岡 聡[1]

**概要**：TSUBAME2.0 のネットワークは Fat tree トポロジであるものの，大規模実行時に集団通信性能が劣化することが観測されている．本稿では想定される原因としてスイッチ間リンクにおけるパケット衝突とスイッチ間リンクの性能劣化に着目し，それぞれの問題を緩和するネットワーク設定を提示し，バンド幅および集団通信性能への影響を示す．ネットワーク設定の改善の結果，通信の確率的な遅延の発生をほぼなくすことができ，大規模実行時のインジェクションバンド幅において 16.0%〜39.5%の性能向上を確認した．

## 1. 背景

2010 年に東京工業大学に導入された TSUBAME2.0 スーパーコンピュータ [1] は，日本初の 1PFlops を超える計算機であり，4000 基以上の GPU を備える GPU スパコンという面で知られているが，2 つの独立した 2 段 Fat tree トポロジの InfiniBand QDR ネットワークで構成される Full-bisection Multi-rail ネットワークで 1408 台の計算ノードが接続されている点も特徴である．Fat tree ネットワークでは，Edge スイッチ–ノード間のリンクの本数と同数以上のリンクで Edge スイッチ–Core スイッチ間を接続することで，スイッチ間リンクが通信ボトルネックとなることを防いでいる．一方，Multi-rail ネットワークとは，全ノードが複数のネットワークに所属することで，ネットワークの数に比例した通信性能を得ることができるものであり，TSUBAME2.0 では I/O ノードへの接続の有無を除いて相似形である 2 つの InfiniBand QDR ネットワークを同時に使用することによって，任意の 2 ノード間で理論上は 80Gbps の通信性能を得ることができる．これらの性質から，TSUBAME2.0 は全体全通信やランダムな 1 対 1 通信が多く発生するアプリケーションにおいて Torus などの他のトポロジを持つコンピュータに比べて高い性能を示すことが期待される．

しかしながら，実際の TSUBAME2.0 における大規模実行時には，集団通信の性能が想定している理論性能よりも低くなる現象および，普段は 1 秒以下で終了するサイズの集団通信において確率的に 5 秒以上遅延する現象が経験的に知られており，通信性能がボトルネックとなっているアプリケーションにおいてネットワークが致命的な実効性能低下要因となっている．

[1] 東京工業大学 学術国際情報センター
Global Scientific Information and Computing Center, Tokyo Institute of Technology
[2] JST, CREST
[a)] nomura.a.ac@m.titech.ac.jp

## 2. TSUBAME ネットワークの通信性能評価

上記のように経験的に知られていた通信性能の劣化について，2 種類のマイクロベンチマークを用いて通信性能を計測することで，実際にどの程度の通信性能劣化が発生しているかを確認した．以下全ての実験は，**表 1** に示す環境 (TSUBAME2.0 の Thin Node) において，以下のような条件下で通信性能を評価した．

- 1 ノードあたり 1 プロセスを配置した．
- ランク番号の割り当てはノード番号順とした．同一ラック，同一 Edge スイッチ配下のノードのランク番号は連続している．
- 環境変数 `MV2_NUM_HCAS` を設定することで，MVAPICH2 に複数 rail を使った通信を行うよう指定した．

### 2.1 ランダムペアの Sendrecv 性能

1 つ目のベンチマークとして，`MPI_Alltoall` 関数の 1 フェーズの通信を模してバンド幅を計測するベンチマークを実行した．通信する全プロセスをランダムに組み合わせ，各ペアにおいて一定メッセージサイズの `MPI_Sendrecv` を同時に行うことにより，バイセクション通信性能を測定した．各ノード間で通信の実行時間にばらつきが発生するが，集団通信をこのようなフェーズごとの通信として実装した場合には，フェーズ間では一番遅いプロセスの遅延が

**表 1** 評価環境の緒元 (TSUBAME Thin ノード)

| | |
|---|---|
| ノード数 | 1408 (うち 1300 台を使用) |
| ネットワーク | Dual Rail InfiniBand 4x QDR (40Gbps x 2) |
| トポロジ | 2 段 Fat Tree |
| スイッチあたりのノード数 | 14 もしくは 16 |
| CPU | Intel Xeon E5670 x 2 (2.93GHz) |
| OS | SuSE Linux Enterprise Server |
| MPI ライブラリ | MVAPICH2 1.8 |
| OFED ドライバ | MLNX_OFED_LINUX-1.5.3-3.0.0 |
| サブネットマネージャ | OpenSM 3.3.9.MLNX_20111006_e52d5fc-0.1 |

各プロセスに伝播していき，最終的な集団通信の実行時間に対する支配項となるため，今回の実行においては一番実行に時間がかかったペアの実行時間を通信実行時間として定め，実行結果は各ノードのインジェクションバンド幅に換算した．なお，すべての実験において疑似乱数シードを固定することで同じノード数の実験における通信相手の組み合わせを固定し，通信相手の組み合わせによる性能への影響を排除している．

**図 1** にメッセージサイズを 512MiB としてプロセス数を変えたときの最低インジェクションバンド幅の推移を示す．バンド幅が高いほど通信性能が良いと言える．40 ノードまでの実行ではほぼ理論性能である 7.5GB/s の性能が出ているが，ノード数が増えるにつれて最良時の最低インジェクションバンド幅が 5.2GB/s, 3.9GB/s, 3.1GB/s, 2.6GB/s と段階的に低下している様子が観測される．これは，Edge スイッチをまたぐ通信が支配的になるにつれて，スイッチ間リンクにおける通信の衝突が発生しやすくなるためと思われる．性能低下が離散的である原因は，1 つのスイッチ間リンクの通信の多重度が離散的であるためと思われる．

参加ノード数が 600 を超えると，最低インジェクションバンド幅が 500MB/s を下回る試行が出現するようになり，ノード数が増えるにつれてその頻度は増加していることが観察された．本稿ではメッセージサイズ 512MiB の試行のみを図示しているが，他のメッセージサイズにおける実行時にも傾向は変化しなかった．実験時間が限られていたため，以降の本ベンチマークについてはメッセージサイズ 512MiB の場合のみを計測している．

### 2.2 Alltoall 性能

2 つめのベンチマークとして `MPI_Alltoall` の実行時間を測定した．**図 2** に 1 通信あたりのメッセージサイズを 1MiB としたときの `MPI_Alltoall` の実行時間の推移を示す．実行時間が短いほど通信性能が良いと言える．最低インジェクションバンド幅ベンチマークと同様に，参加ノード数が 600 ノードを超えると異常値 (通信が通常時の数倍かかるケース) が発生することが分かる．これらの異常値を排除しても (**図 3**)，通信性能にばらつきが大きいことがわかる．また，最低インジェクションバンド幅ベンチマークと同様にノード数が増えると実効バンド幅が低下していることが読み取れる．

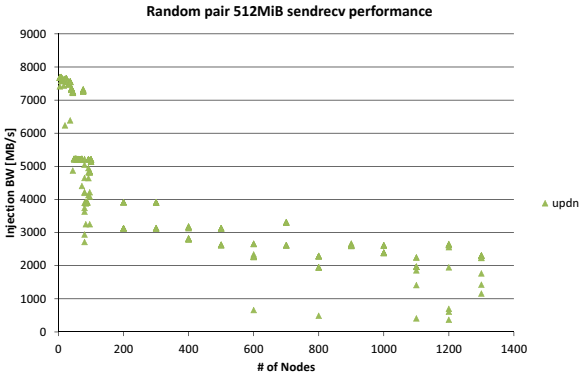


**図 1** TSUBAME2.0 におけるインジェクションバンド幅

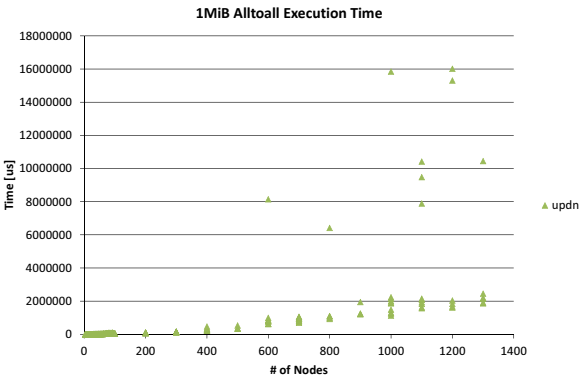


**図 2** TSUBAME2.0 における Alltoall 通信実行時間

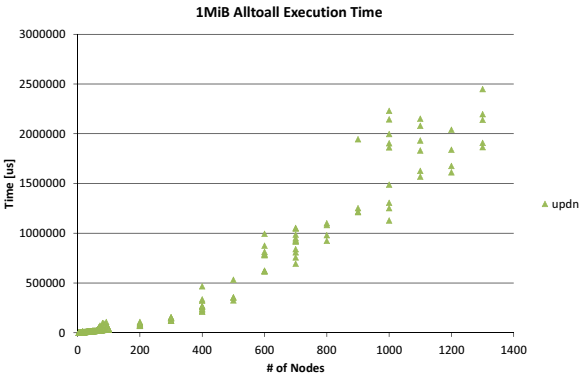


**図 3** 図 2 から異常値を除いたもの

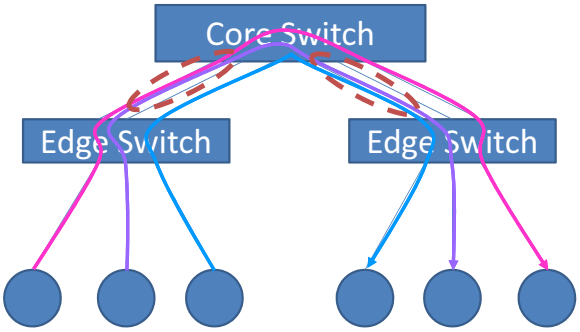


**図 4** Fat Tree におけるルーティングの衝突

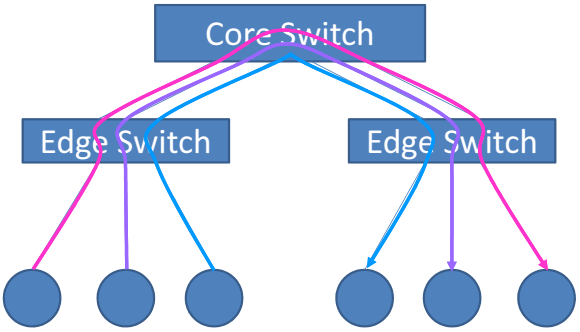


**図 5** Fat Tree における理想的なルーティング

## 3. 性能劣化の要因

今回，性能劣化の要因としてルーティング戦略によるものおよび，不調リンクの存在によるものの 2 つを想定した．以下にそのそれぞれについて性能が低下する原因を述べる．

### 3.1 ルーティング戦略

我々は，段階的な性能劣化の原因として，**図 4** に示すように，2 段 Fat Tree の Edge スイッチと Core スイッチ間のリンクが有効に使われておらず，通信の衝突が起こっていると推量した．理想的には**図 5** のように，各通信が上流のリンクの間で完全にバランスされてリンク速度を使い切る通信ができるのであるが，InfiniBand のルーティングはパケットの送信先ごとに次ホップのスイッチを固定する静的ルーティングであるため，すべての通信パターンにおいて通信が衝突しないルーティングを行うことは不可能である．また，集団通信に頻出する通信パターンにおいてのみルーティングを最適化することも考えられるが，実際の運用では故障ノードの発生によって歯抜けとなるノードが出現するため，この方法は大規模計算機環境では破綻する．実際に今回の実験中にも複数台の計算ノードがダウンして，実行対象のノードリストから取り除かれている．

InfiniBand でのルーティングテーブルはサブネットマネージャが管理しており，TSUBAME2.0 では OpenSM 3.3.9 の UpDn ルーティング戦略に基づいて決定されている．今回は，実験時に最新であった OpenSM 3.3.15 に附属する以下のルーティング戦略を用いて，ルーティング戦略による通信性能の変化を観察した．

- UpDn: TSUBAME で通常利用されている戦略である．ツリー状のトポロジを共通する祖先に到達するまで送信元および送信先から辿り，得られた経路のうち最短のものを採用する．
- MinHop: 送信元と送信先を結ぶ最短経路のうち任意の経路を次ホップの転送先として選択する
- Fat Tree: 完全 Fat Tree を仮定して通信の衝突を避けるように次ホップの転送先を送信先に応じて順に割り振ることで通信の衝突を回避する．
- DFSSSP(Deadlock Free Single-source Shortest Path)[2]: 全ホストで全対全の通信経路を作成した時に負荷が完全にバランスするように経路を構成する．

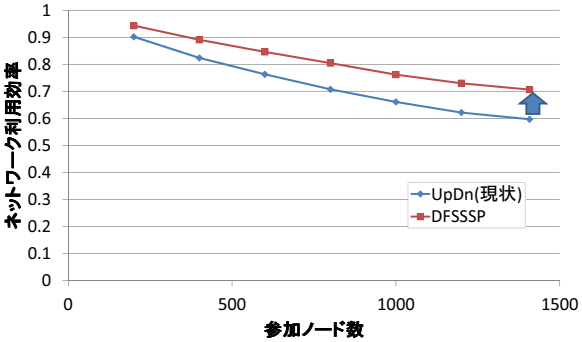


**図 6** UpDn および DFSSSP におけるネットワーク利用効率のシミュレーション結果

**図 6** はネットワークシミュレータ ORCS[3] による UpDn および DFSSSP におけるバイセクションバンド幅のシミュレーション結果である．この結果より，ルーティング戦略を変化させることで通信性能が向上することが期待できる．

### 3.2 不調リンク

TSUBAME2.0 の大規模ネットワークにおいてはスイッチ間リンクにケーブル異常やスイッチポート異常に起因する不調なリンクが発生していることが，`ibdiagnet` コマンドにおけるポートの速度およびパフォーマンスカウンタの値から判明した．これらのエラーは一時的なもの (一旦リンクを再起動することによって復帰する) と恒久的なものの両方があり，場合によっては全く通信できなくなるわけではなく，速度低下や大量のパケットロスを起こすものの通信できてしまうものがある．そのようなリンクがネットワーク上に存在すると，そのリンクを用いる通信だけが遅延することにより，全体の通信のボトルネックとなってしまうことが推察される．以下に TSUBAME2.0 で観察された主な不調リンクの症状を示す．

- 速度低下: TSUBAME2.0 のネットワークにおける 1 ポートあたりの通信速度は 4x QDR(QDR データレートのリンク 4 本分) の 40Gbps であるが，1x QDR や

表 2　ベンチマーク条件

| ラベル | ルーティング戦略 | 不調リンク無効化 |
|---|---|---|
| minhop | MinHop | N |
| updn | UpDn | N |
| ftree | Fat Tree | N |
| hetero | Fat Tree / DFSSSP | Y |
| updn2 | UpDn | Y |

4x SDR(いずれも 10Gbps) に縮退してしまっているリンクが発生した．

- 異常パケットの発生: TSUBAME2.0 のネットワーク全域で全く通信を行っていない状態においても，symbol_error_counter や port_rcv_errors のようにパケット破損が発生しているときに上昇するカウンタが秒間数百パケットのオーダで上昇していることを確認した．

後者については何らかの原因でパケットが無限ループしている，制御パケットが異常発生している，もしくはパケットを正常にエンコード・デコードできなくなったことなどが原因と考えられる．いずれにせよ，当該リンクを通過するパケットは (場合によっては確率的に) 遅延もしくは破損して正常に届かなくなると考えられる．そこでネットワーク中のこれらの以上リンクを個別に停止することで，Fat Tree のトポロジを多少犠牲にして安定したネットワークを構成した．

## 4. 性能劣化の検証と改善

前節で述べた 2 つの原因および解決策を実際に実行して 2 節に述べたそれぞれのベンチマークにおける性能の変化を観察した．本来はそれぞれの解決策を切り分けて実行すべきであるが，実験時間の制約や，上記の結論に至った過程および不調リンクの無効化が不可逆な操作である点を理由に個別に切り分けた実験を行うことはできなかった．また，DFSSSP は TSUBAME2.0 ネットワークの 1st rail に適用した際に，正常な性能が発揮できず，全通信にかかる時間が 10 倍以上となりネットワークが不安定になってしまったため，1st rail に dfsssp を用いた実験は行っていない．**表 2** に今回実行した実験の条件を示す．

hetero においては，1st rail に Fat Tree 戦略，2nd rail に DFSSSP と，異なる戦略を採用した．TSUBAME2.0 の Multi-rail ネットワークは I/O ノードへの接続の有無を除いて相似であり，同じルーティング戦略を用いることで相似形のルーティングテーブルが作成され，性能低下する通信パターンも同様のものとなることが考えられる．そのため，各 rail におけるルーティング戦略を違えることによって弱点となる通信パターンが分散して相互に通信性能を補完しあうことが期待される．

全ベンチマークの実行結果を**図 7**，**図 9** および**図 10** に示す．ルーティング戦略の選択にかかわらず，3.2 節に示したような異常値が観察されていることがわかる．不調リンクの排除後の実験のみを示した**図 8**，**図 11** と比較すればわかるように，速度低下やカウンタ異常を起こしているリンクを排除することによってこれらの異常値のほとんどが発生しなくなり，図 3 と比較して異常値以外の部分においても性能が安定するようになった．

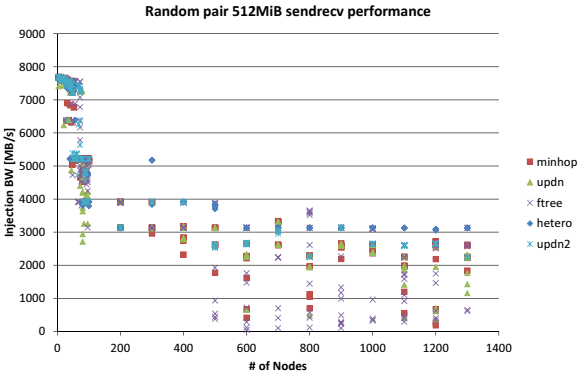


**図 7**　最低インジェクションバンド幅の分布

また，図 7 および図 8 に示す通り，hetero 戦略において他のルーティング戦略と比べて 16.0%〜39.5%最低インジェクションバンド幅の向上がみられた．グラフの形状を観察することにより，hetero 以外のトポロジにおいて，ノード数およびノードの組み合わせによって性能低下の幅が振動していたものが，hetero 戦略の採用によってその分の性能低下を防げるようになったと推論することができる．

図 10 は，`MPI_Alltoall` の性能比較である．Alltoall 通信においては通信相手を切り替えながら Sendrecv 相当の通信を (ノード数-1) 回行っている．そのため，不調リンク排除前の実験では通信時間のぶれが蓄積し，通信性能が安定していないことが分かる．他方，図 11 に示す不調リンク排除後の実験では，わずかな異常値を除いて通信性能は極めて安定していることがわかる．Alltoall の性能比較では，ルーティング戦略の選択に起因する性能の差は確認できなかった．この点について，何故最低インジェクションバンド幅と違う傾向が見られるか解明することは今後の課題である．なお，ノード数が 600 を超える部分での Alltoall のバンド幅 1.74GB/s であり，最低インジェクションバンド幅の約半分である．ノード数 200 付近の hetero における性能劣化を含めて，何故 600 ノード超で性能劣化が起こるかの解明も今後の課題である．

## 5. おわりに

我々は，TSUBAME2.0 における大規模実行時の通信性能低下がどの程度発生しているかを計測し，スイッチ間リンクの混雑と性能劣化に着目し，これらを解消することで通信速度の向上を図った．理論性能からの平均性能の乖離については rail ごとに異なるルーティング戦略を用いることでマイクロベンチマークにおいて最悪インジェクションバンド幅の平均値で 16.0%〜39.5%の性能向上を，確率的

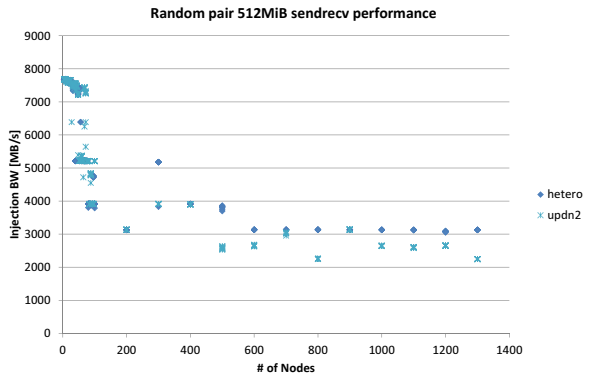


**図 8** 最低インジェクションバンド幅の分布 (不調リンク排除後)

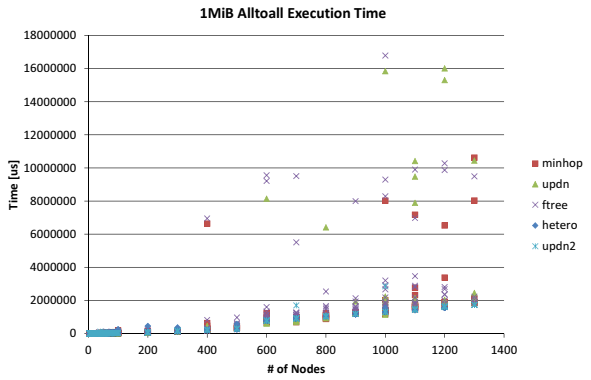


**図 9** Alltoall 通信実行時間の分布

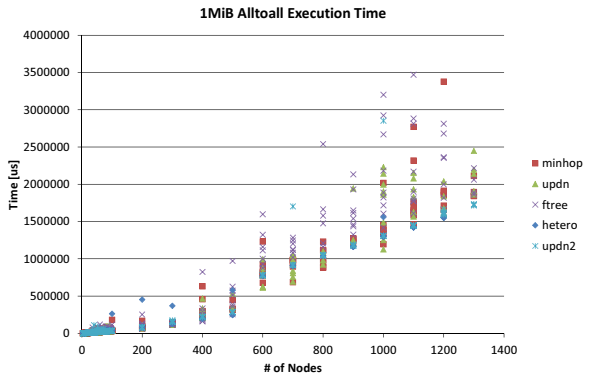


**図 10** 図 9 から異常値を除いたもの

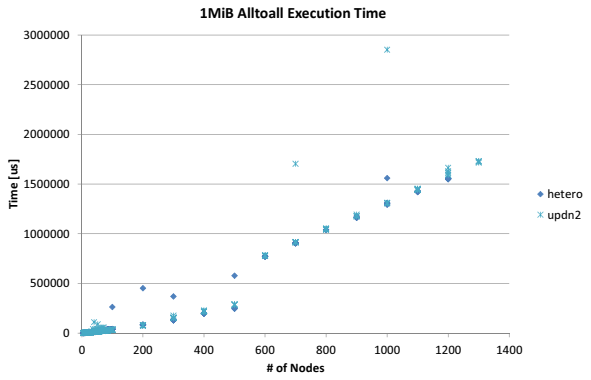


**図 11** Alltoall 通信実行時間の分布 (不調リンク排除後)

な性能劣化についてはネットワーク上の不調リンクを遮断することで観測されなくなるという結果を得た．

今後は今回の実験時に発生したバグのために実験の継続を断念したルーティング戦略の評価を行い，より理論性能に近い実行性能を得るとともに，シミュレーション上の性能と実際の実行時の性能のギャップおよび，マイクロベンチマークと実際の集団通信の性能のギャップについて原因を明らかにする必要がある．また，実アプリケーションでの性能の変化も比較して，これらの処置がアプリケーションの性能向上に資することを示す必要がある．

なお，東京工業大学学術国際情報センターでは今回の実験の成果をもとに，TSUBAME2.0 の運用において不調リンクの検出を強化して通信性能の劣化を未然に防ぐ運用を 2012 年 9 月より行っている．

**謝辞** TSUBAME2.0 における InfiniBand の性能調査のため，2012 年 8 月 8 日〜10 日の間の 48 時間，TSUBAME2.0 のネットワークを占有しての実験を行わせていただきました．本期間中 TSUBAME2.0 の利用を控えていただいた全てのユーザに感謝いたします．

また，本実験の実施時には，NEC，Mellanox，Torsten Hoefler 博士および，Jens Domke 氏に多数の助言およびご協力をいただきました．ここに感謝いたします．

## 参考文献

[1] 東京工業大学学術国際情報センター：TSUBAME 計算サービス，http://tsubame.gsic.titech.ac.jp/.
[2] Domke, J., Hoefler, T. and Nagel, W.: Deadlock-Free Oblivious Routing for Arbitrary Topologies, *Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, IEEE Computer Society, pp. 613–624 (2011).
[3] Schneider, T., Hoefler, T. and Lumsdaine, A.: ORCS: An Oblivious Routing Congestion Simulator, Technical Report 675, Indiana University (2009).