

MEC 向け ROS2-FPGA ノード自動生成ツールの並列処理性能評価

岡崎 英佑[†] 菅谷 みどり[‡] 大川 猛[†]東海大学情報通信学部組込みソフトウェア工学科[†]芝浦工業大学工学部情報工学科[‡]

1. はじめに

IoTや5G通信における計算基盤として注目の集まるMEC (Multi-access Edge Computing)は、エッジから近い位置にサーバを配置することで通信の最適化や超低遅延化などの役割が期待されている[1]。これらを実現するために、並列処理・負荷分散・低消費電力・再利用性が求められ、高効率なFPGA処理をROS 2 (Robot Operating System 2)ノードとして動作させることが望ましい。ROS 2は、通信方式の一つとして採用されているpublish/subscribe通信により、ROS 2ノード間で非同期が可能である。また、ROS 2は、容易に機能の追加や変更が可能で、再利用性が高い。これまで我々は、PYNQ上で動作するROS 2ノードの自動生成ツールF0rEST (FPGA-Oriented Easy Synthesizer Tool) [2]を開発してきた。F0rESTを用いることで、FPGAロジックをROS 2システムに統合したROS2-FPGAノードを自動生成し、迅速に低電力かつ低遅延なアプリケーションを構築することが可能である。また、ROS2-FPGAノードによる並列処理を可能とするため、複数ROS2-FPGAノードが生成可能なツールmeta-F0rESTを提案した[4]。しかし、複数のROS2-FPGAノードが動作する際の性能や課題が明らかとなっていない。

本研究の目的は、複数ROS2-FPGAノードの遅延時間や消費電力を評価し、複数ROS2-FPGAノードの課題を明らかとすることである。

2. 複数ROS2-FPGAノード

図1は、meta-F0rEST[4]が生成可能な複数ROS2-FPGAノードの構成である。Talker/Listenerノードは、ROS2-FPGAノードの入出力としての役割を担う。各ROS2-FPGAノードは、ARMプロセッサ上で動作するFPGAノードとPYNQドライバ、FPGA上で動作するFPGAロジックから構成される。FPGAノードは、FPGAロジックのドライバであるPYNQドライバを用いて、Talker/ListenerノードとFPGAロジック間のインターフェースとして機能する。

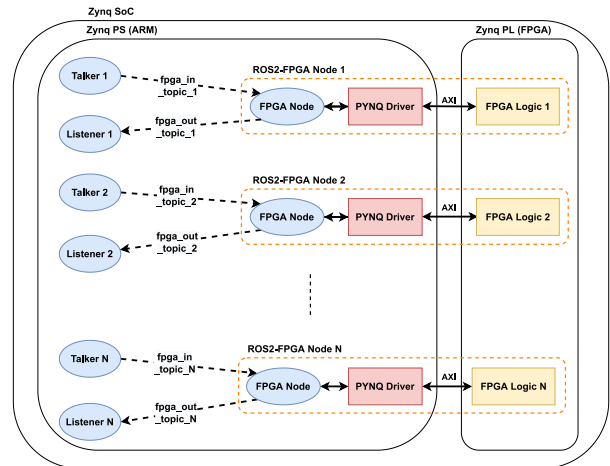


図1 複数ROS2-FPGAノードの構成

3. 評価方法

複数ROS2-FPGAノードを運用する際に、ボトルネックとなる処理を調査するため、消費電力やROS2-FPGAノード遅延時間を計測した。各ROS 2ノードは独立して動作し、計測も独立して行った。評価環境として、FPGAボードにM-KUBOS[5]を用いた。M-KUBOSはCPUにARM Cortex-A53(4コア)、FPGAにZynq Ultrascale+ MPSoC (xczu19eg-ffvc1760-2-i)が搭載されている。OSはPYNQ v2.5.1を用いた。

評価には、1024点FFT回路のIPコアを複数含むROS2-FPGAノードを用いた。FFT回路は、Xilinx FFT IPコア[6]をC++デザインから呼び出して一体として高位合成可能なライブラリ[7]を用いた。ROS2-FPGAノードの入出力を行うTalker/Listenerノードは1000BASE-Tで繋がれたデスクトップPC上で実行され、全体の遅延時間をメッセージ送信(Talker)から受信(Listener)までの時間とした。なお、Talkerノードは、2秒毎に1024個の要素を含む浮動小数点(float)配列をPublishする。

4. 評価結果

表1は、FFT回路やDMA回路などを含む全体のハードウェア使用量と、配置配線後の消費電力の推定値である。FFT回路のコア数を2のべき乗数増やしていくと、評価環境上で32個搭載できた。また、Vivadoによる消費電力の推定値は、コア数が1個の時に3.503W、32個の時に5.798Wだった。しかし、表1の結果は回路を駆動するた

表1 コア数ごとのHW量と消費電力の推定値(W)
(括弧内はxczu19におけるHW使用率)

コア数	LUT	FF	BRAM	DSP	電力推定値(W)
1	10,751 (2.06)	15,218 (1.46)	10.5 (1.07)	27 (1.37)	3.503
2	20,065 (3.84)	28,891 (2.76)	21 (2.13)	54 (2.74)	3.579
4	38,881 (7.44)	56,233 (5.38)	42 (4.27)	108 (5.49)	3.725
8	76,453 (14.63)	110,918 (10.61)	84 (8.54)	216 (10.98)	4.029
16	152,871 (29.25)	221,700 (21.21)	168 (17.07)	432 (21.95)	4.607
32	305,356 (58.42)	442,146 (42.29)	336 (34.15)	864 (43.90)	5.798

めに必要な電力推定値であり、ROS 2 ノードとして動作する際は、より多くの電力を要すると考えられる。

図3は、ROS2-FPGA ノードの遅延時間である。左端がROS2-FPGA ノード化前の処理時間であり、FFT 処理の結果を得られるまでに約 0.76ms 要した。FPGA ロジックは、DMA 転送により入出力が行われ、データ入出力(約 0.05ms)・FPGA ロジック処理(約 0.31ms)の時間を含む。左端以外はROS2-FPGA ノードの通信を含む遅延時間であり、複数のROS2-FPGA ノードが動作しているときに1つのノードを計測した結果である。ノード数が増えるごとにCPU-FPGA 間データ送受信時間が増加した。これは、データ転送帯域利用率の増加が要因であると考えられる。また、FPGA ロジック処理時間が増加した。これにはソフトウェアから処理終了の判断をする時間も含まれており、終了判断をポーリングにより行うため、複数のプロセスがCPU コアを占有し合った結果、処理時間が増加したと考えられる。さらに、ノード入出力通信遅延時間において、1 ノード時に約 2.45ms であったが32 ノード時は約 5.23ms と、2 倍以上の時間を要した。これは、デスクトップ PC と M-KUBOS 間の通信トラフィックの増加に起因していると考えられる。

5. 結論

本研究では、FFT 処理を行う複数 ROS2-FPGA ノードの性能を評価した。評価環境において32 個のFFT 回路を搭載することができ、消費電力は、FFT 回路が1 個の時に3.503W、32 個の時に5.798W だった。ROS2-FPGA ノードの遅延時間の評価では、ノード数が増えるごとにROS 2 の通信遅延が増加した。また、ノード数に関わらず、ROS 2 の通信遅延が全体遅延時間の半分以上を占めているこ

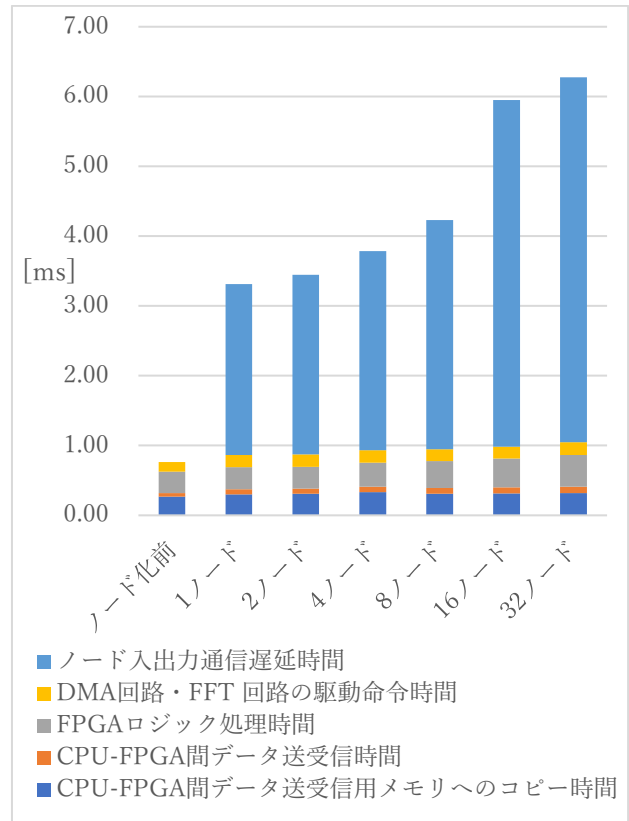


図3 ROS2-FPGA ノードの遅延時間とが分かった。今後の課題は、複数 ROS2-FPGA ノードが動作する際のROS 2 の通信遅延を削減する方法の検討である。

謝辞

本研究は、JST, CREST, JPMJCR19K1 の支援を受けたものです。

文 献

- [1] P. Porambage, J. Okwuibe, M. Liyanage, M. Yliantila and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2961-2991, Fourthquarter 2018, doi: 10.1109/COMST.2018.2849509.
- [2] D. P. Leal, M. Sugaya, H. Amano and T. Ohkawa, "Automated Integration of High-Level Synthesis FPGA Modules with ROS2 Systems," 2020 International Conference on Field-Programmable Technology (ICFPT), 2020, pp. 292-293, doi: 10.1109/ICFPT51103.2020.00052.
- [3] 森隼人, 菅谷みどり, 大川猛, "FOrEST による ROS2-FPGA ノード自動生成手法の評価", 情報処理学会第84 回全国大会 6J-01.
- [4] 岡崎英佑, 菅谷みどり, 大川猛, "PYNQ 上で動作する複数 ROS2 ノード自動生成ツールの開発", 信学技報, vol. 122, no. 174, pp. 61-62, 2022 年 9 月.
- [5] FPGA コンピューティングプラットフォーム M-KUBOS (エム・キューボス) | 株式会社 PALTEK <https://www.paltek.co.jp/design/original/m-kubos/index.html>
- [6] Fast Fourier Transform (FFT), <https://www.xilinx.com/products/intellectual-property/fft.html>
- [7] FFT IP Library, <https://docs.xilinx.com/r/en-US/ug1399-vitis-hls/FFT-IP-Library>