

OpenKasugai Controller (Extensions)

v1.0.0

用語定義①

用語	説明
ファンクションチェーン	データ処理フローを構成する機能と機能間の接続関係の定義。テンプレートとして用意されている場合もある
ファンクション	ファンクションチェーンに定義される1つの機能のことであり、データ処理モジュールのこと
データフロー	ファンクションチェーンを物理リソースにデプロイするものであり、デプロイしたもの
ユーザ	本システムの利用者。3種類のユーザがいる ① DF実行管理者（DFの配備を要求する人 + DFへの入力とDFからの出力を受け取るアプリを管理する人） ② 本システムの運用者（オペレータ。DF実行管理者にDFを提供する人） ③ 開発者（DFに必要な機能(ファンクションなど)を開発する人）
GW	DF実行管理者に見える入口・出口（KafkaやRabbitMQ、NetScalerのようなイメージ。クラウドのGWサービスになる場合もあるだろう）
StartPoint(SP)、 EndPoint(EP)	データフローの送信元・宛先のこと。現在はカメラ・配信サーバのこと。 将来的には入口GW・出口GWのことなどに拡張し、セッションの切り張りなどの機能を持つ予定
切り替え	データフローの付け替え処理
スケジューリング条件	データフローを配備する際の配備先に関する指定（例：配備先ノード/デバイスの直接指定(や候補からの削除)やフィルタ実行戦略） データフロー配備要求時に定義済のスケジューリング条件を選択することで、当該条件に沿って配備先が決定される
フィルタ実行戦略	スケジューリング時に適用するフィルタに関する指定（適用するフィルター一覧やその順番、フィルタを通過した配備先候補の上位何個を採用するか等）

用語定義②

用語	説明
親bs (親bitstream)	FPGAの基本となる回路。以下の子bsを書込んで初めて使用できる
子bs (親bitstream)	親bsの上に書込む実際のFPGAの回路（例：リサイズ回路）
子bs書込み	下記の3パターンの子bsの書込みの総称
自動書込み	子bs未書込み状態のFPGAが配備先に選択された場合、動的に子bsを書込んでからDFのファンクションを配備すること
手動書込み	運用中にコントローラを停止させることなく、子bs未書込み状態のFPGAに対して子bsを書込み、コントローラに新たな配備先候補として認識させること
上書き	子bs書込み済のFPGAに子bsを書込むこと
リセット	下記の2パターンのFPGAのリセットの総称
FPGAのリセット	子bs未書込み状態のFPGAに戻すこと
子bsのリセット	今書込まれている子bsを書込み直後の初期状態に戻すこと。DF配備時に設定した接続情報などを初期化する
FPGAリソースの解放	DF配備に伴い消費していたFPGAのリソースを解放して未使用状態に戻すこと。K8sリソースやコントローラとは関係ないFPGA側の操作である
分岐	下記の2パターンの分岐の総称。ただし、文脈によっては「コピー分岐」を略している場合もある
コピー分岐	1つのデータを複数の宛先にコピーして送る
条件分岐	1つのデータを特定(複数の場合もあり)の宛先に送る
統合	複数の処理結果を1つにして流す
Glue	入出力の接続種別が異なるファンクションで、接続種別の変換を行うファンクション（例：TCP→DMA） 接続種別が異なるファンクションの間に配備することで、(本来接続できない)それらのファンクションを繋いだDFを配備できる

目次

1. はじめに

- 提供価値
- 拡張機能一覧

2. 複数のファンクションに分岐するDF

- コピー分岐処理モジュール
- OpenKasugaiコントローラの機能拡張

3. 接続種別が異なるファンクション同士を繋いだDF

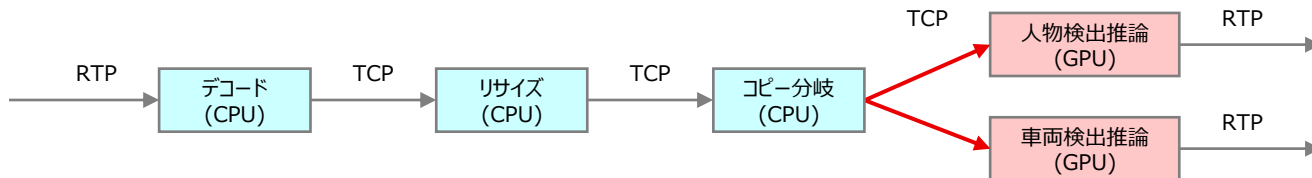
- Glue処理モジュール
- OpenKasugaiコントローラの機能拡張

1. はじめに

提供価値

- 多様なデータフローを配備できるように、配備できるDFのバリエーションを増やす
 - コピー分岐処理モジュールを用いて途中で複数のファンクションに分岐するデータフローを配備できる
 - 接続種別の変換処理モジュール(Glue処理モジュール)を用いて接続種別が異なるファンクション同士を繋いだデータフローを配備できる

複数のファンクションに分岐するDF



接続種別が異なるファンクション同士を繋いだDF



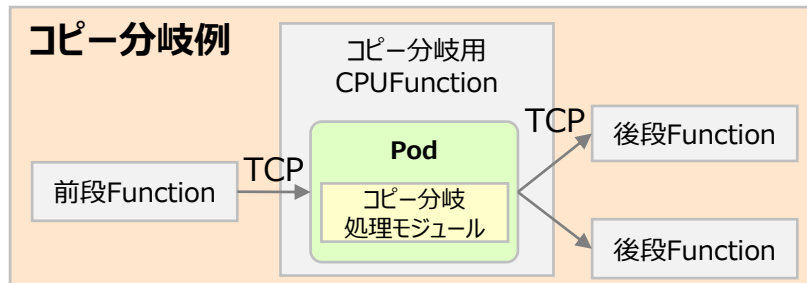
拡張機能一覧

構築できるDF	拡張機能の概要
複数のファンクションに分岐するDF	<ul style="list-style-type: none">ファンクションとしてコピー分岐処理モジュールを提供する入出力が複数あるファンクションを管理できるようにコントローラを拡張する分岐があるDFを定義できるようにコントローラを拡張する
接続種別が異なるファンクション同士を繋いだDF	<ul style="list-style-type: none">ファンクションとしてGlue処理モジュールを提供する入出力の接続種別が異なるファンクションを管理できるようにコントローラを拡張する

2. 複数のファンクションに分岐するDF

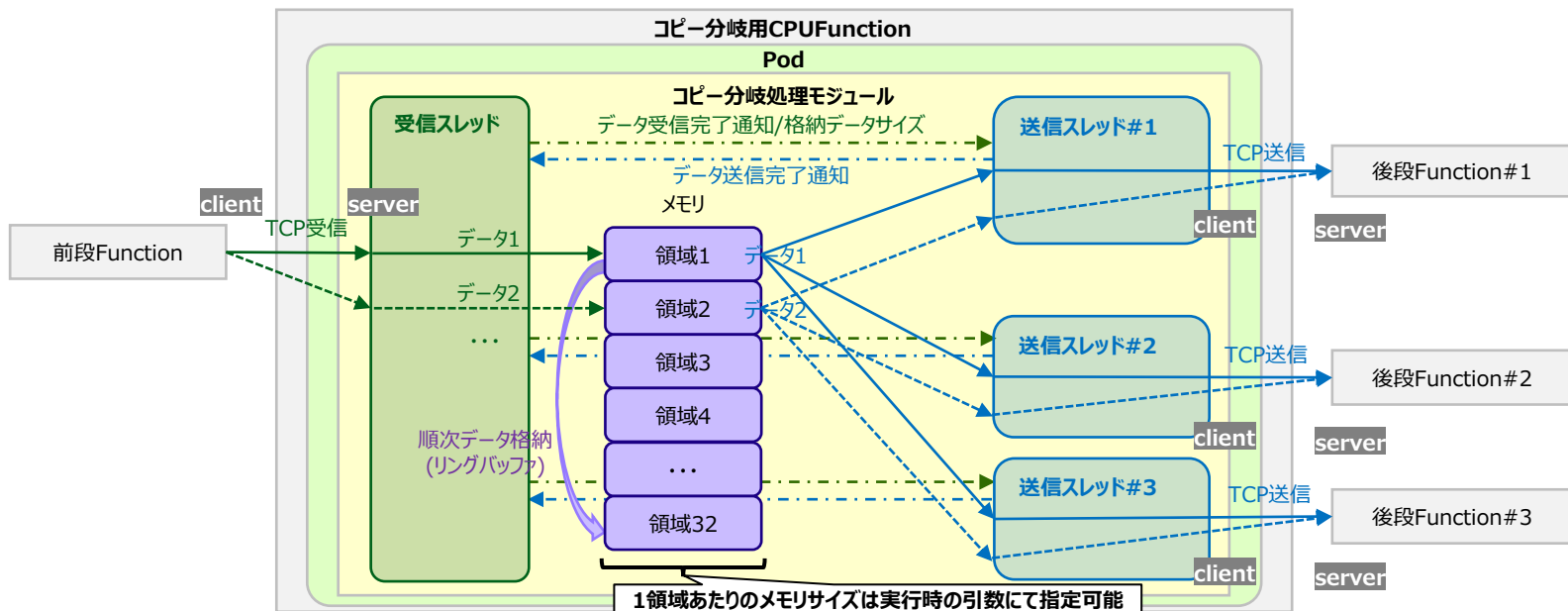
コピー分岐処理モジュール： 概要

- 前段ファンクションからの入力を複数の後段ファンクションへ分配して出力する処理
 - TCP入力/TCP出力
 - 分岐数は任意の数に設定が可能



コピー分岐処理モジュール： ブロック図

- コピー分岐処理モジュールは1つの受信スレッドと分岐数分の送信スレッドにて構成される
 - 受信/送信スレッドは各送信スレッド毎かつ各メモリ領域毎に用意した受信(送信)完了通知でやり取りを行う
 - 受信スレッドは前段Functionからのデータ受信及びメモリへの格納が完了したら、格納データサイズと受信完了を通知する
 - 送信スレッドは、データ受信が完了していれば後段Functionへ格納データサイズ分データを送信した後、送信完了を通知する



OpenKasugaiコントローラの機能拡張： 入出力が複数ある関クションの管理

- アプリカカタログのCRを拡張し、関クションの入力ポート数と出力ポート数を管理する
 - 入出力のポート数は最大数を記載し、DF定義で実際に使われるポート数が決まる
 - DF定義のチェックとして、DF定義内の各関クションのポート数が、カタログに記載された最大ポート数を超えていないかのチェックを行う
 - 注) 関クションの「ポート」はカタログ上の概念であり、Podの「NIC」などとは異なる概念である

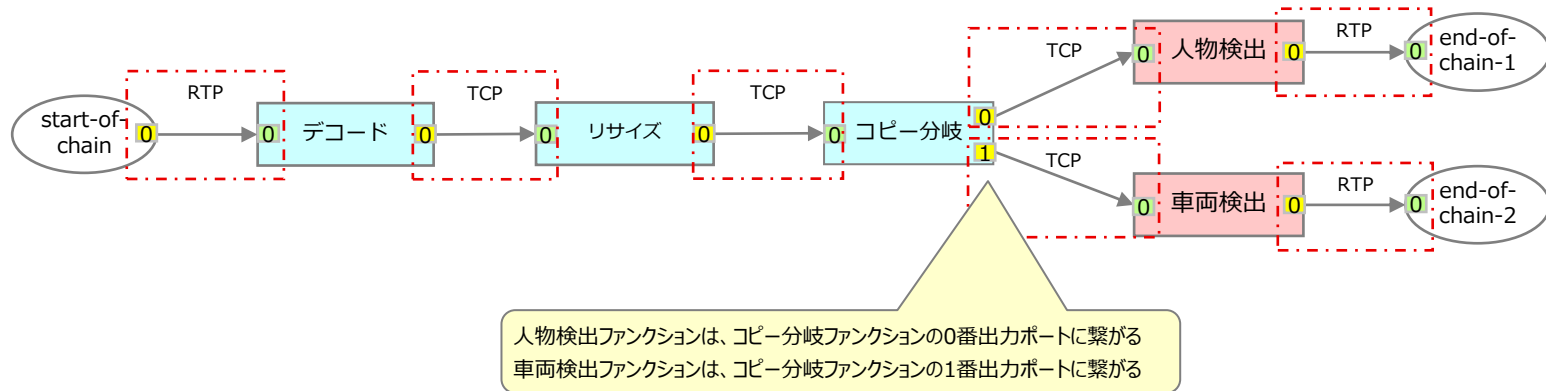
カタログの例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: funcinfo-copy-branch
  namespace: wbfunc-imgproc
data:
  deployableItems: '[
    {
      "name": "item1",
      "regionType": "cpu",
      "inputInterfaceType": "host100gether",
      "outputInterfaceType": "host100gether",
      "configName": "cpufunc-config-copy-branch",
      "specName": "spec1"
    }
  ]'
  spec: '[
    {
      "name": "spec1",
      "minCore": 1,
      "maxCore": 1,
      "maxDataFlowsBase": 1,
      "maxCapacityBase": 15,
      "maxInputNum": 1,
      "maxOutputNum": 10
    }
  ]'
```

最大接続数を記載

OpenKasugaiコントローラの機能拡張： 分岐があるDFの定義

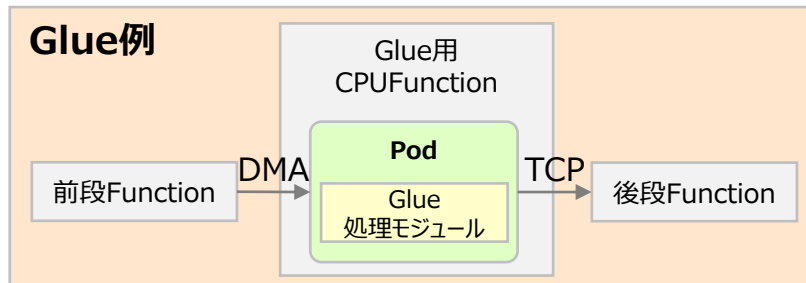
- ファンクションが一直線に並んでいない分岐があるDFを表現できるようにデータフローのCRを拡張する
 - 分岐の際、どちらのポートに繋がっているのかを番号で管理（四角内の数字）
 - DFの開始点・終了点も複数管理（start-of-chain, end-of-chain-1,2）



3. 接続種別が異なるファンクション同士を 繋いだDF

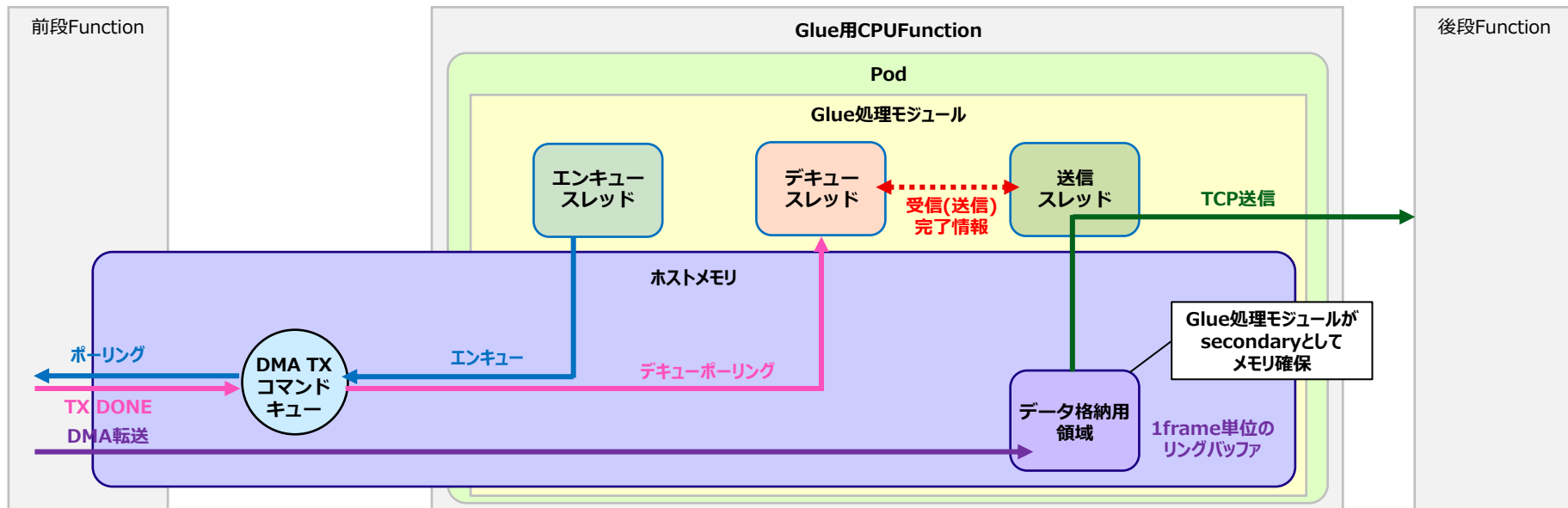
Glue処理モジュール： 概要

- 異なるコネクションタイプ間のデータ転送を接続する処理
 - DMA入力/TCP出力



Glue処理モジュール： ブロック図

- Glue処理モジュールはエンキュースレッド、デキュースレッド、送信スレッドにて構成される
 - DMA TXコマンドキューに対し、エンキュースレッドにてTX DMAリクエストを行い、デキュースレッドにてポーリングを行う
 - デキュースレッドは前段FunctionのDMA転送完了を検知したら、送信スレッドに対し受信完了を通知する
 - 送信スレッドは受信完了通知を受けたら、データを後段FunctionへTCP送信した後、デキュースレッドに対し送信完了を通知する



OpenKasugaiコントローラの機能拡張： 入出力の接続種別が異なるファクションの管理

- アプリカタログのCRを拡張し、ファクションの入力ポートと出力ポートの接続種別を管理する
 - DF定義のチェックとして、DF定義内の各コネクションの接続種別が、両端のファクションのカタログに記載された接続種別に一致するかをチェックする

カタログの例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: funcinfo-glue-fdma-to-tcp
  namespace: wbfunc-imgproc
data:
  deployableItems: '[
    {
      "name": "item1",
      "regionType": "cpu",
      "inputInterfaceType": "mem",
      "outputInterfaceType": "host100gether",
      "configName": "cpufunc-config-glue-fdma-to-tcp",
      "specName": "spec1"
    }
  ]'
  spec: '[
    {
      "name": "spec1",
      "minCore": 1,
      "maxCore": 1,
      "maxDataFlowsBase": 1,
      "maxCapacityBase": 15,
      "maxInputNum": 1,
      "maxOutputNum": 1
    }
  ]'
```

入出力の接続種別を記載