

Mesosを用いた ソフトウェア定義計算機の研究

IIJイノベーションインスティテュート 技術研究所

阿部 博 (abe@ij.ad.jp)

2015/11/11

本セッションの概要



クラウド
データセンター
Mesos

【 11/11(水)1日目 】

Apache Mesosで挑む新しいクラウドアーキテクチャ

ビッグデータ処理や機械学習など、巨大なデータを扱うために、多数のコンピュータで分散処理を行うことが増えてきています。こういった用途では、Hadoopを始めとした分散処理のためのミドルウェアを使うことが一般的ですが、こうしたミドルウェアを使用したとしても、実際の処理を行うプログラムを開発する際には「分散処理である」ということを意識しなければ、高いパフォーマンスを得ることができません。

利用者の目的は分散処理ではなく、あくまで高いパフォーマンスでの計算です。分散処理を意識せずに、一般的な手法で利用できる「高パフォーマンスな計算機」ができれば、巨大なデータ処理がより容易になります。

IIJではそのような環境を実現するための新しいクラウドアーキテクチャの実現に向けて研究に取り組んでいます。今回の発表では、Apache Mesosを用いたコンセプト実装についてご紹介します。Mesosは「分散処理のためのカーネル」であり、多数のノードのリソース管理を行うための仕組みが提供され、これを使って仮想的な計算機環境を構築します。

IIJ技術研究所の立ち位置

- 最先端の研究や開発にトライすることがミッション
- 本日は話す内容がIIJでサービス化される保証はありません

アジェンダ

- 新しいクラウドアーキテクチャへの要求
- ソフトウェア定義計算機
- Mesosを用いたProof of Concept実装
- まとめ

新しい クラウドアーキテクチャ

新しいクラウドアーキテクチャ

- デザインゴール
 - 巨大なデータを扱える
 - 高いパフォーマンスを提供する
 - バックエンドシステムを隠蔽できる

巨大なデータを扱える

- 100TBデータと10GBデータの違い（2015年現在）
 - 10GB: 単一ノードで処理可能（主記憶/ローカルディスクに収まる）
 - 100TB: 複数ノードでしか処理できない
- 巨大なデータ処理が求めるシステムの定義
 - データ量に対して処理がスケールする

高いパフォーマンスを提供する

- クラウドシステムにおける処理量
 - あるタスクに対する単位時間あたりに投入される資源量(CPU/メモリ/IO帯域など)で規定される
- 高パフォーマンスなクラウドシステムの定義
 - タスクに対して要求する資源量を割り当てる速度が速い（オーバーヘッドが小さい）

バックエンドシステムの隠蔽

- 処理系に対する透過性
 - バックエンドシステムが置き換わっても動作する
- バックエンドシステムの定義
 - クラウド計算リソース、ミドルウェア

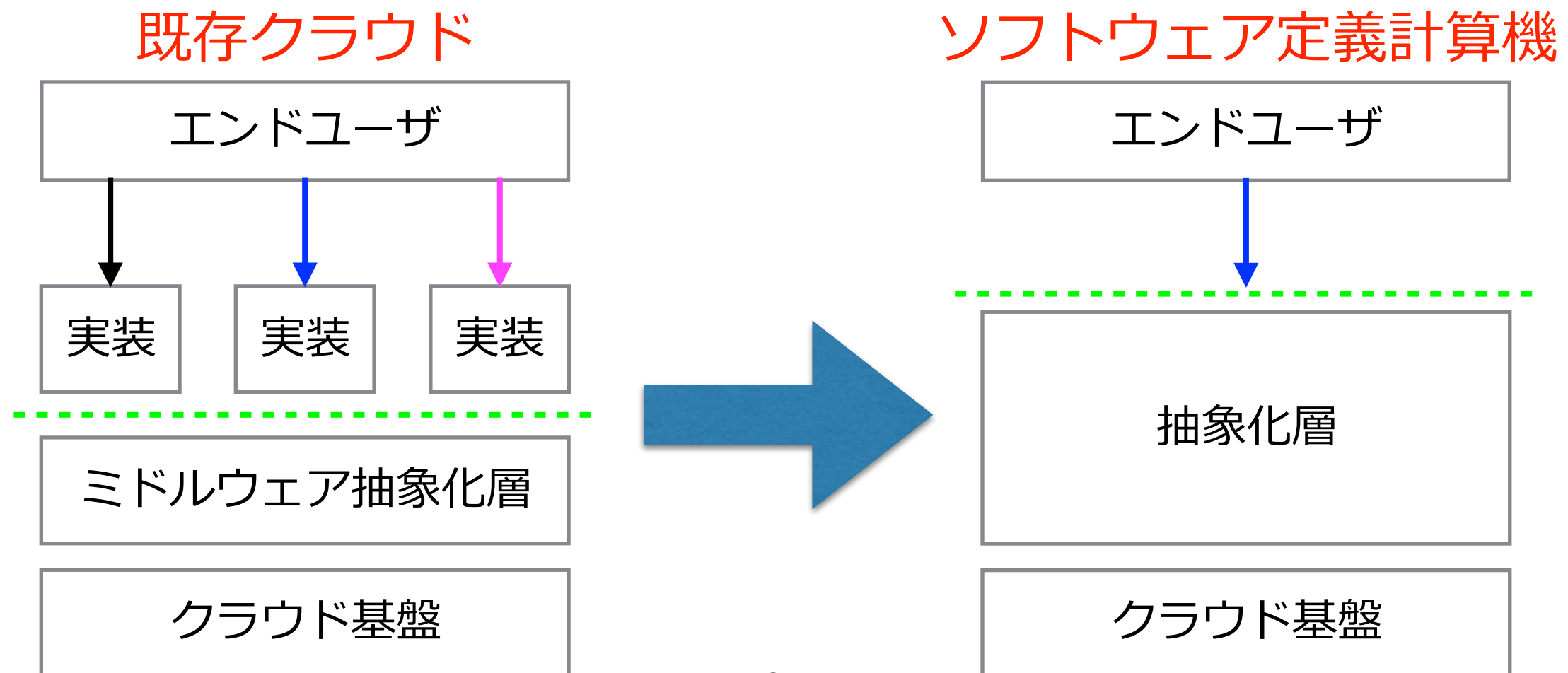
本クラウドのユースケース

- エンドユーザからの視点
 - 既存のIaaSの利用モデルでは直接計算資源を扱う必要があるが、本クラウドアーキテクチャでは計算資源自体は隠蔽される
 - エンドユーザは個別チューニングなどには行えないがシステムが提示する提案を選択することでその提案に沿った処理の最適化が行われる

ソフトウェア定義計算機

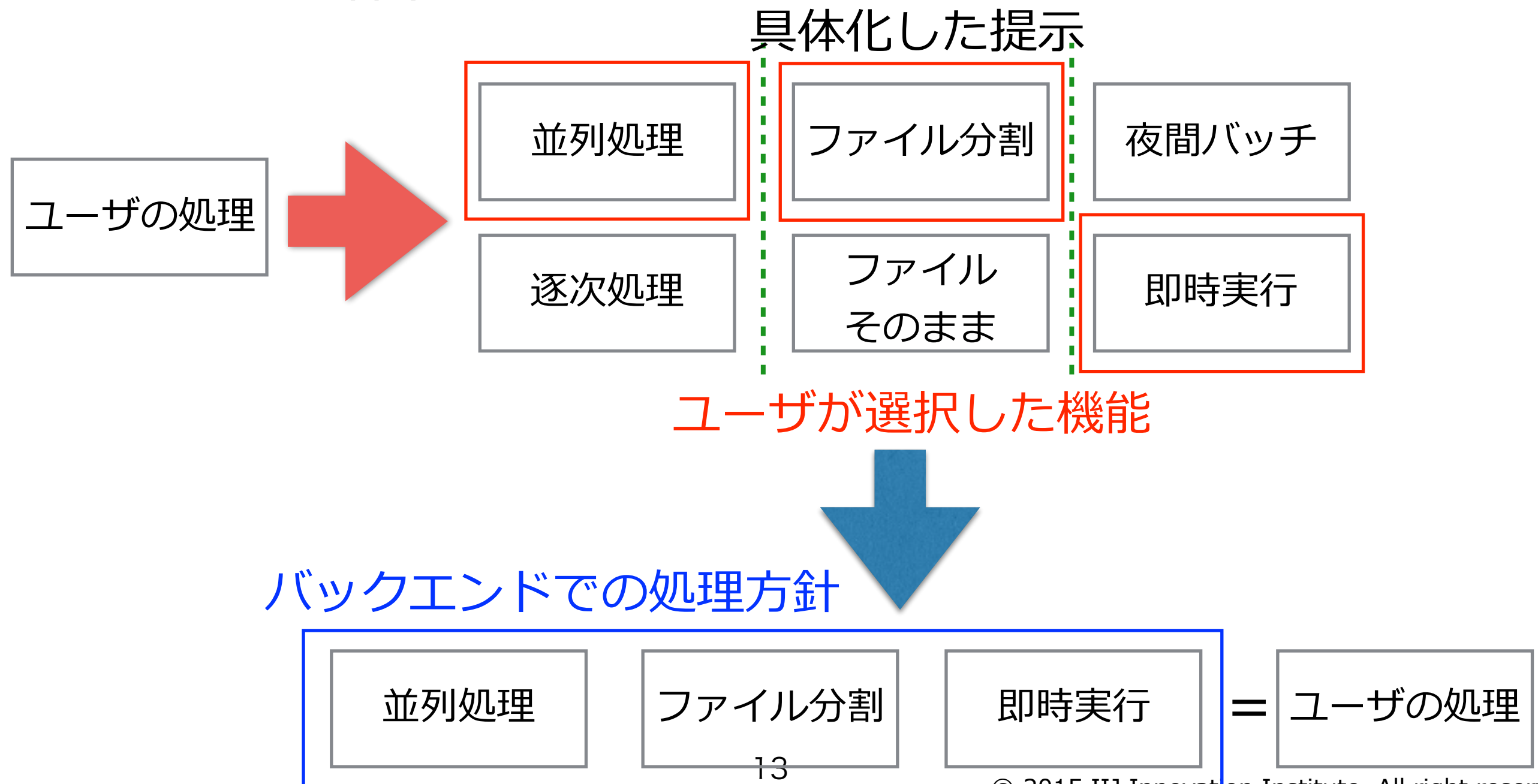
ソフトウェア定義計算機とは

- 「ソフトウェア定義計算機」とは、エンドユーザからの要求（タスク）とクラウドバックエンドでの処理を橋渡しする抽象化層である



抽象化層の機能：処理方針の決定

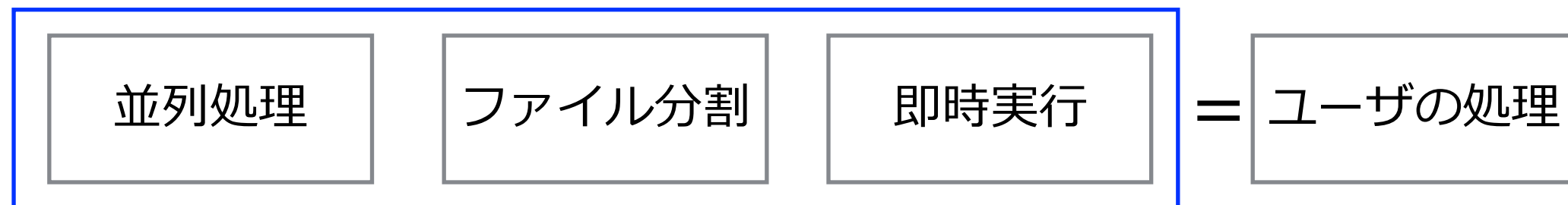
バックエンドシステムが提供できる機能に応じてユーザの処理を具体化し提示する。



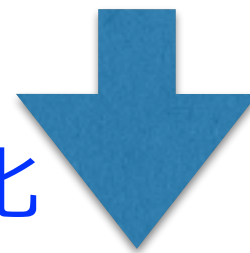
抽象化層の機能：タスクの生成

ユーザの処理をバックエンド処理に最適化した形でコンパイルし、実行タスクを生成する。

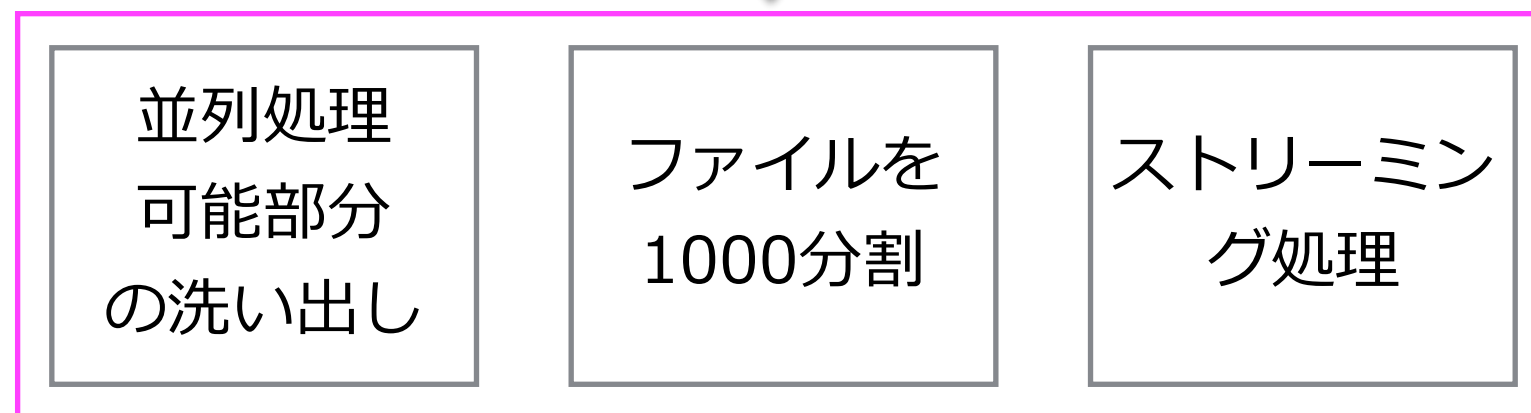
ユーザが選択した
バックエンド
での処理方針



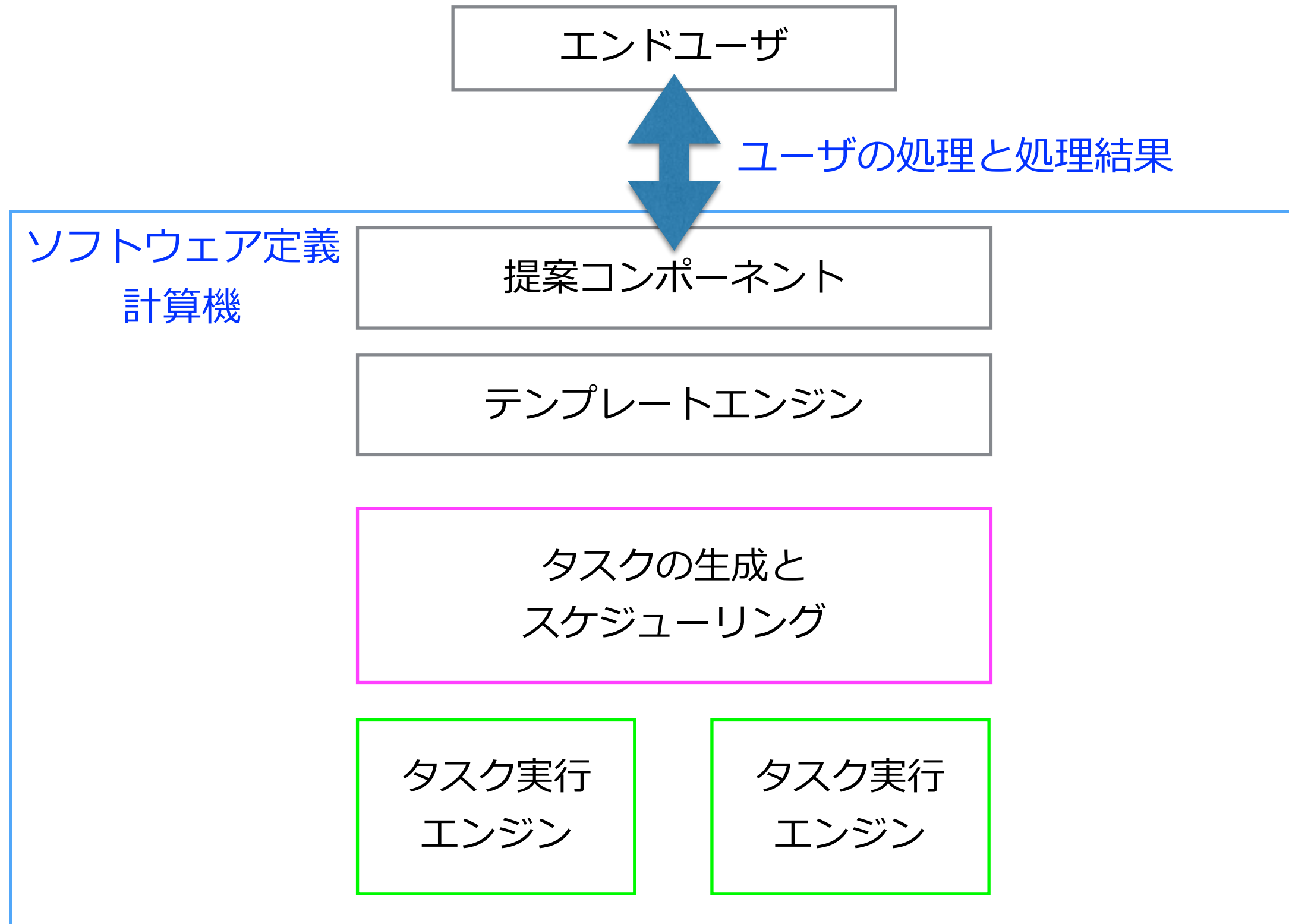
処理の
テンプレート化



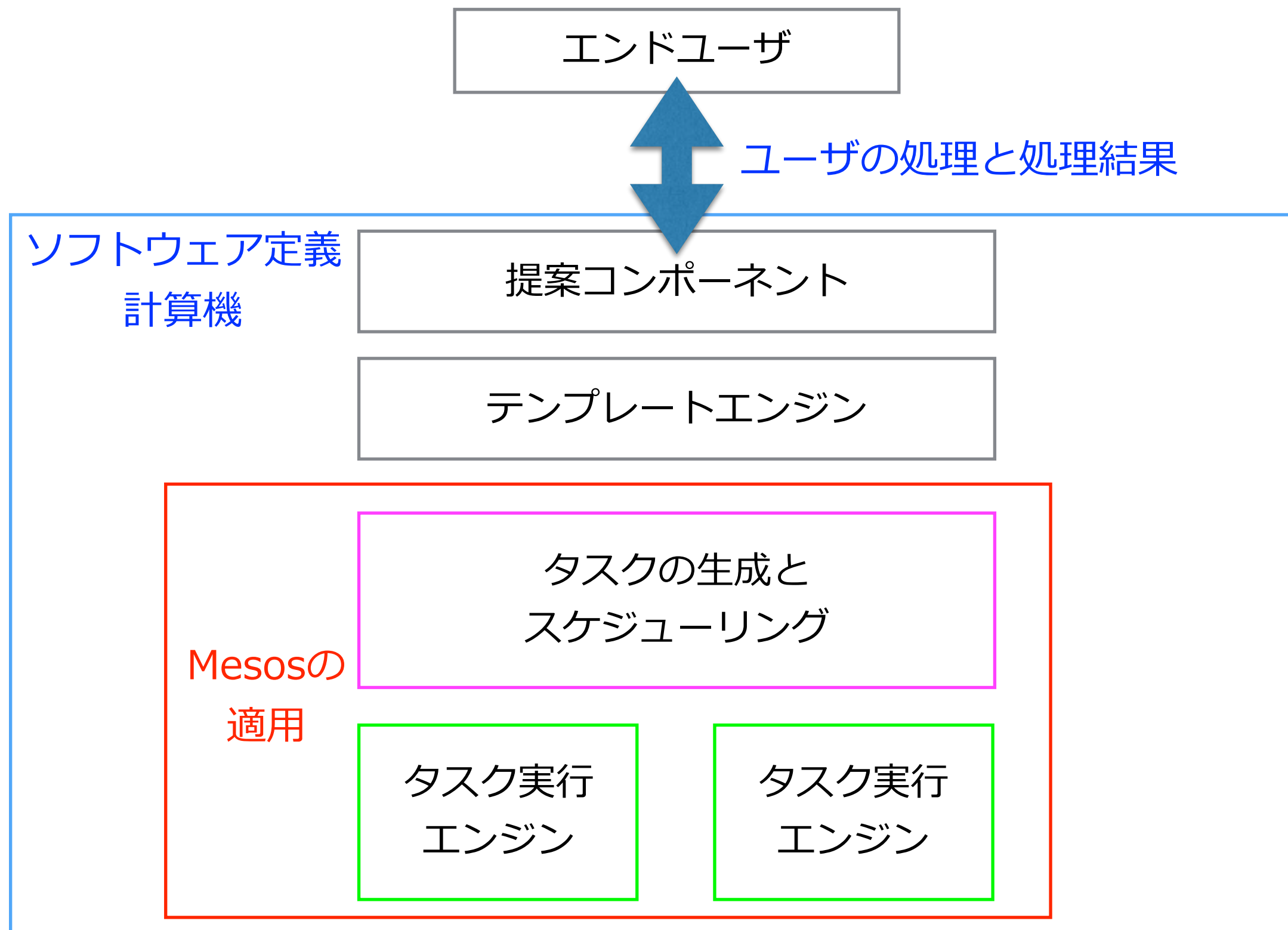
テンプレートと
処理をコンパイルして
実行タスクを生成



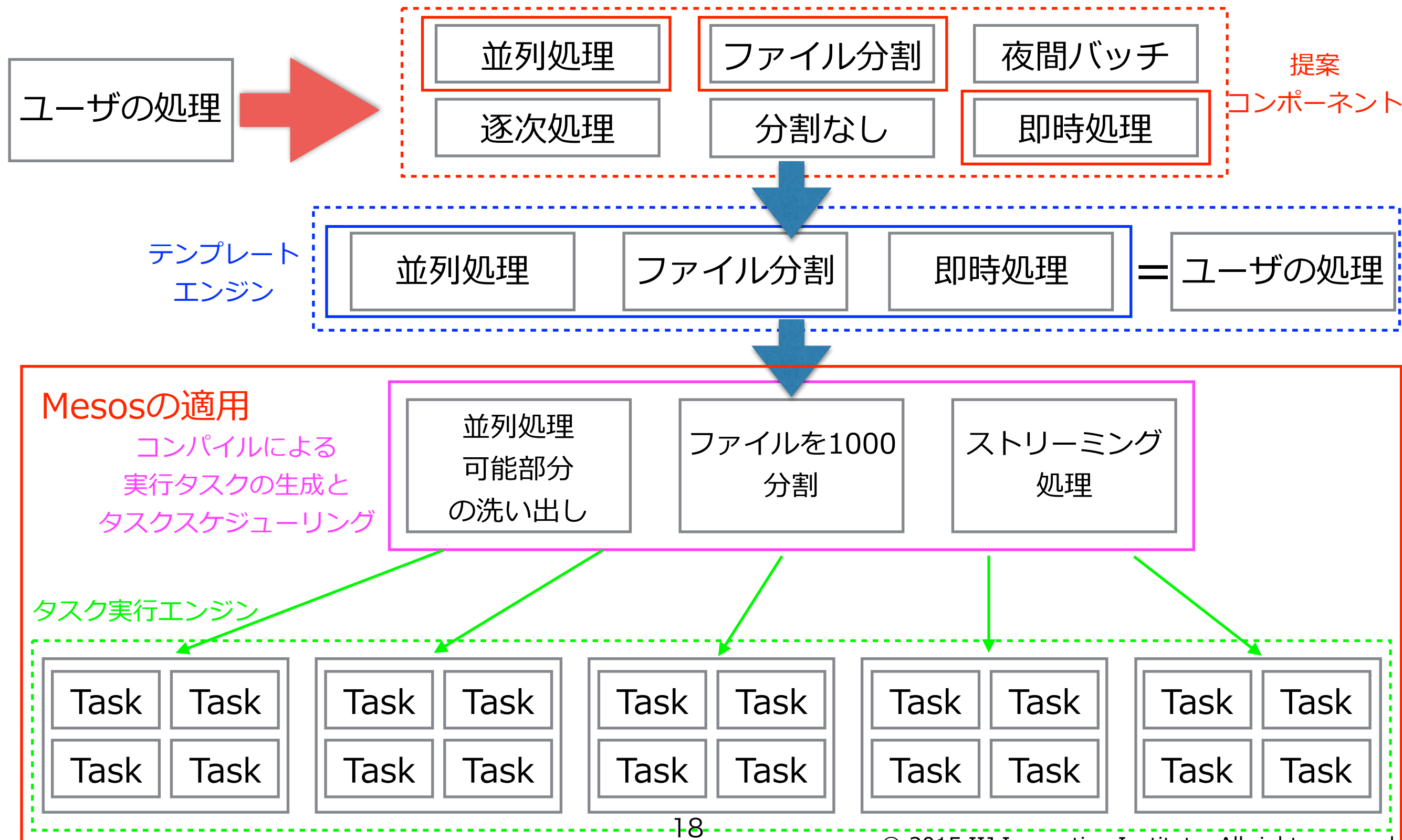
ソフトウェア定義計算機の実現



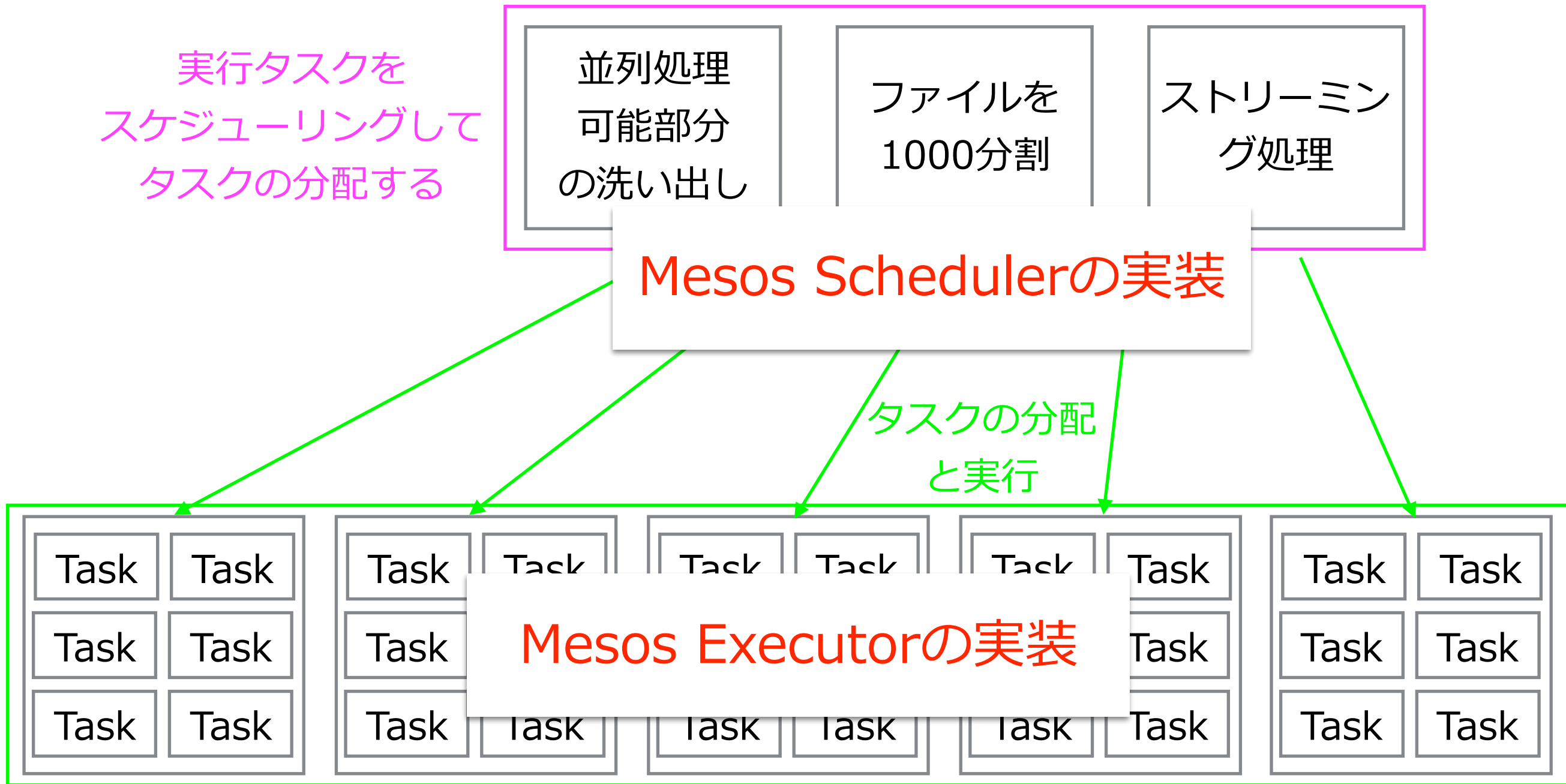
ソフトウェア定義計算機の実現



抽象化層の全体像

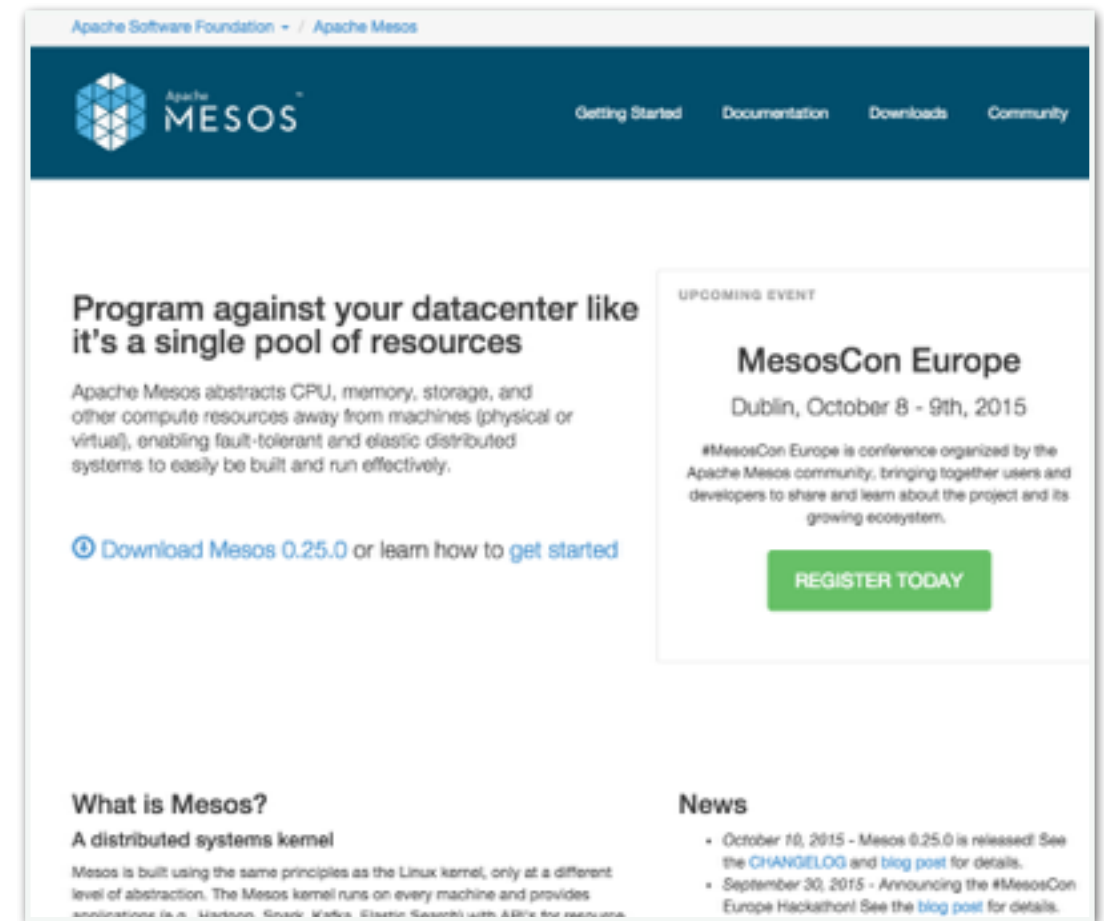


スケジューラとタスクのMesos化

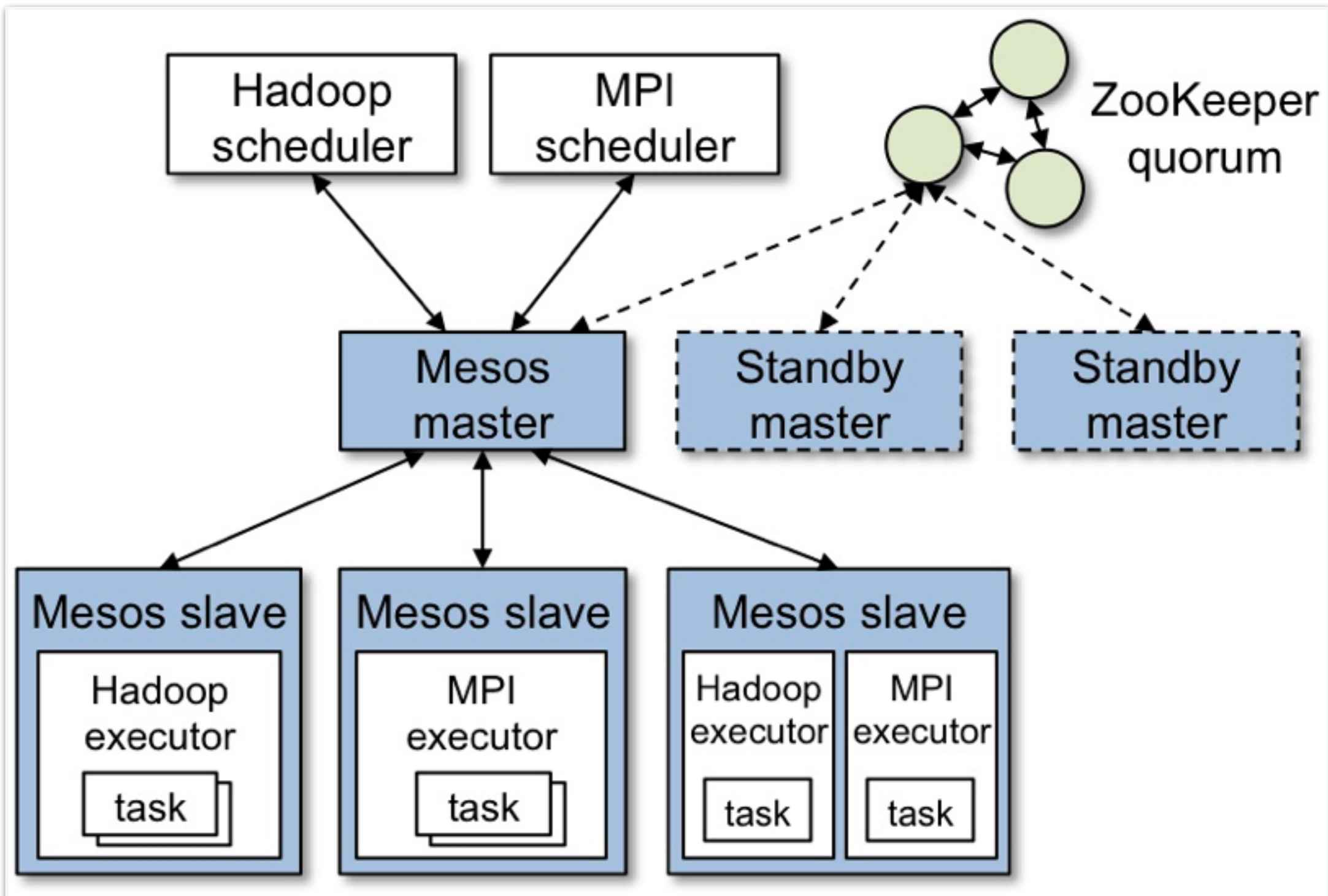


Apache Mesosとは？

- <http://mesos.apache.org/>
 - Apache Projectの一つ
- A distributed systems kernel
 - DCOS(DataCenter OS)
- Mesosでできること
 - Zookeeperを利用したMasterの高信頼性の実現
 - Slaveノードの追加によるスケールアウトの実現
 - 複数フレームワーク同時実行による利用率の向上

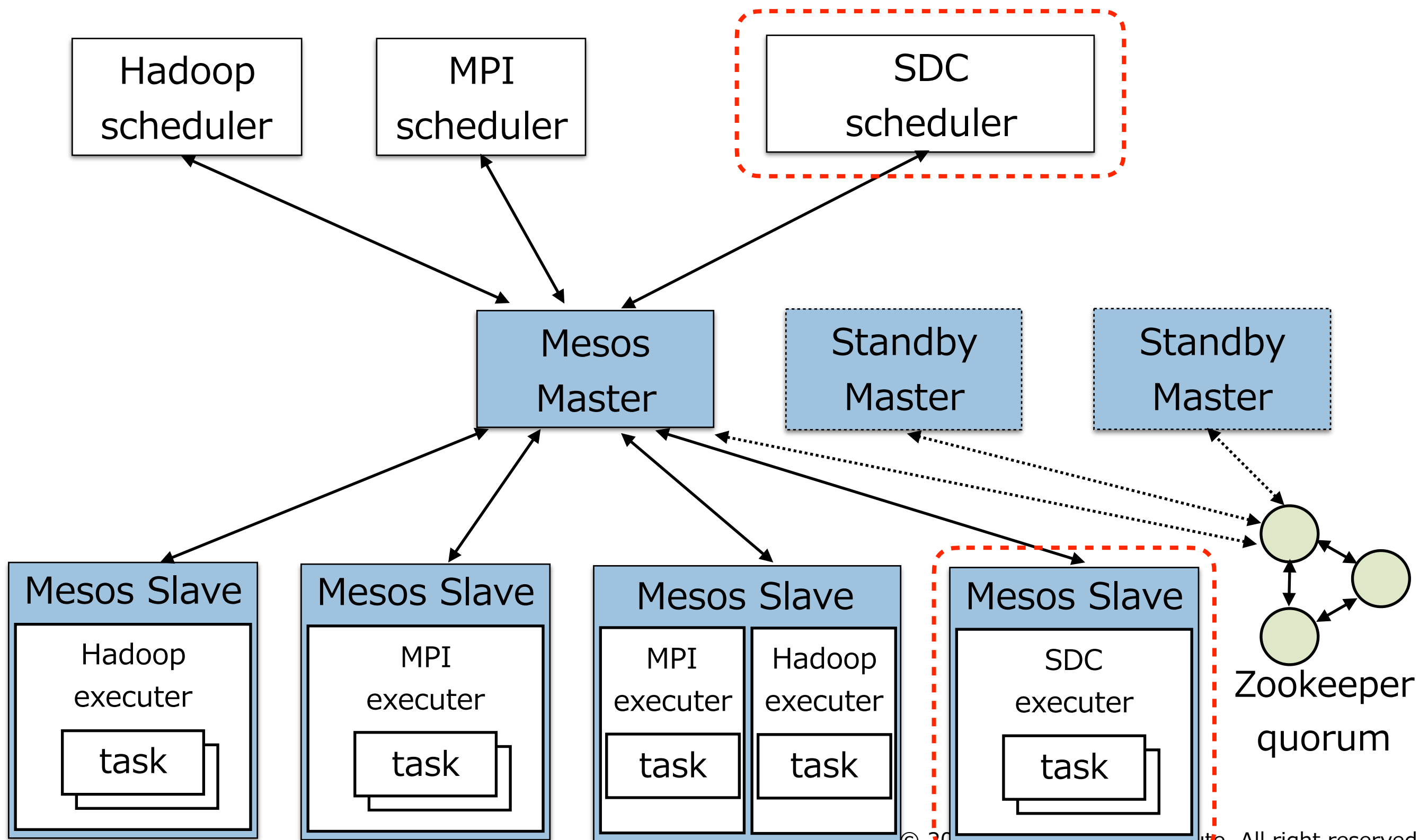


Mesosのアーキテクチャ



“Mesos Architecture”, <http://mesos.apache.org/documentation/latest/architecture/>

独自フレームワークの実装



ソフトウェア定義計算機における Mesosの役割

- スケジューラとタスク実行の実現
- クラウド基盤上でMesosノード伸縮が可能
- スケール機構を利用したデータストレージへの到達が可能

Mesosの検証

- Cisco UCS(メモリ: 48GB, CPU: 24 cores)
- Master x 3台、Slave x 10台



Mesos in StarBED

Mesos Frameworks Slaves Offers cluster00						
Master / Slaves						
Slaves Find...						
ID ▼	Host	CPUs	Mem	Disk	Registered	Re-Registered
...5050-2722-S78	192.168.0.138	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S77	192.168.0.137	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S76	192.168.0.136	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S75	192.168.0.135	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S74	192.168.0.134	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S73	192.168.0.133	24	46.1 GB	4.9 GB	11 hours ago	
...5050-2722-S69	192.168.0.131	24	46.1 GB	4.9 GB	4 weeks ago	4 weeks ago
...5050-2722-S58	192.168.0.132	24	46.1 GB	4.9 GB	4 weeks ago	
...5050-2722-S25	192.168.0.130	24	46.1 GB	4.9 GB	4 weeks ago	4 weeks ago
...5050-2722-S24	182.168.0.129	24	46.1 GB	4.9 GB	4 weeks ago	4 weeks ago

Mesos in StarBED

Mesos
Frameworks
Slaves
Offers
cluster00

Master 20151008-200438-2132218028-5050-2722

Cluster: cluster00
Server: 172.16.23.127:5050
Version: 0.24.1
Built: a month ago by root
Started: 4 weeks ago
Elected: 4 weeks ago

[LOG](#)

Slaves

Activated	10
Deactivated	0

Tasks

Staged
Started
Finished
Killed
Failed
Lost

Resources

	CPU	Mem
Total	240	460.7 GB
Used	0	0 B
Offered	0	0 B
Idle	240	460.7 GB

Resources

	CPU	Mem
Total	240	460.7 GB
Used	0	0 B
Offered	0	0 B
Idle	240	460.7 GB

	Host	
--	------	--

	Host	
--	------	--

得られた知見

- メリット
 - ノード管理を委譲できるのでロジック記述に専念できる
 - Slaveのスケールアウトが容易
 - 多様な言語バインディングが提供されている
- デメリット
 - フレームワークを作成するためのドキュメントが少ない
 - 実際に作っている人/使っている人はあまりいない
 - 開発環境を構築するのが難しい

まとめ

- 新しいクラウドアーキテクチャの提示
- ソフトウェア定義計算機の提示
- Mesosを使ったソフトウェア定義計算機の実現に向けた現状の紹介