クラウドを使いこなす技術、その到達点
“RACK” – Real Application Centric Kernel

伊藤忠テクノソリューションズ株式会社
金子　雄大

アジェンダ

- ✓ CTCのクラウドビジネス
- ✓ CTCが考える “Cloud Native”
- ✓ “Cloud Native Application” を実現する “RACK”
- ✓ “Cloud Native Application” のデモ
- ✓ コミュニティ活動

CTCのクラウドビジネス

CTCのクラウドビジネス

第２プラットフォーム

基幹システム、EDBMS、ERP、FileServer

第３プラットフォーム

BigData、Mobile、M2M、IoT、Social

クラウドインテグレーション

- TechnoCUVIC
- ElasticCUVIC
- (Virtustream)
- (Cisco)

- OpenStack
- AWS
- Softlayer
- Windows Azure
- HP Helion
- Cisco

運用サービス

CTC　DC

パートナーDC

今日のお話

第２プラットフォーム

基幹システム、EDBMS、ERP、FileServer

第３プラットフォーム

BigData、Mobile、M2M、IoT、Social

クラウドインテグレーション

- TechnoCUVIC
- ElasticCUVIC
- (Virtustream)
- (Cisco)

- OpenStack
- AWS
- Softlayer
- Windows Azure
- HP Helion
- Cisco

運用サービス

CTC　DC

パートナーDC

OpenStackへの取り組み

OpenStack基盤構築

オープンハードウェア
OpenCompute Project

クラウドOS
Mirantis

ブロックストレージ
SolidFire

オブジェクトストレージ
CleverSafe

ネットワークファブリック
Cumulus Netwoks　Penguin Computing

OpenStack/クラウド活用

PaaS実行環境
Pivotal

Web API 統合プラットフォーム
apigee

Web UI / オーケストレータ
CTC
Challenging Tomorrow's Changes
オリジナル開発

ログ統合
Splunk

バージョン管理
Github

監視
(Zabbixなど)
ビルド管理
(Jenkinsなど)

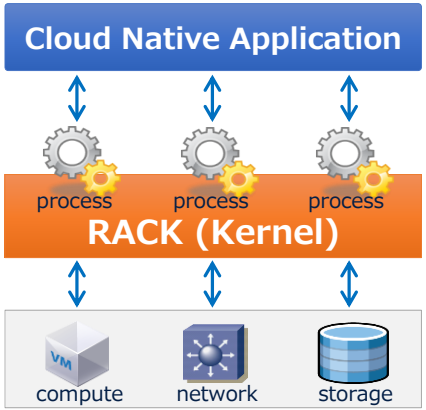
構成管理/デプロイエンジン
(Chef/Ansibleなど)
テスト自動化
(Serverspecなど)

クラウドネイティブ

クラウドネイティブ環境の実現

CTC RACK

Real Application Centric Kernel



CTCが考える “Cloud Native”

“いかに作るか“ から **”いかに使うか”** の段階へシフト

「OpenStackは使えるか？」の議論はすでに終了

「OpenStackをいかに活用するか？」が現在のテーマ

非テクノロジー企業による海外事例

- ✓ BMW　：車メーカー
- ✓ Expedia　：旅行サイト
- ✓ BBVA　：大手銀行

OpenStack Summit 2014 Paris

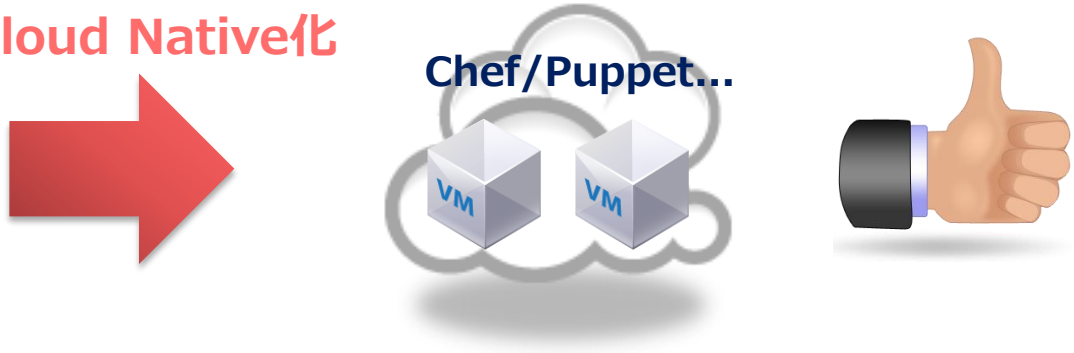
システムの Cloud Native 化

システムを Cloud Native にすることで Cloud のメリットを享受できる

- monolithic
- single-tiered
- legacy

そのまま移行

Cloud Native化



Designing for the cloud

http://docs.openstack.org/arch-design/content/designing-for-the-cloud.html

- ✓ Be a pessimist
- ✓ Put your eggs in multiple baskets
- ✓ Think efficiency
- ✓ Be paranoid
- ✓ But not too paranoid
- ✓ Manage the data
- ✓ Hands off
- ✓ Divide and conquer
- ✓ Think elasticity
- ✓ Be dynamic
- ✓ Stay close
- ✓ Keep it loose
- ✓ Be cost aware

- ➢ すべてのものは壊れると思え
- ➢ マルチプロバイダ、リージョン、AZを活用せよ
- ➢ 移植性の高いアプリケーションにせよ
- ➢ **自動化**を活用せよ　→　**Cloud API**の活用
- ➢ コンポーネントは小さくせよ
- ➢ …

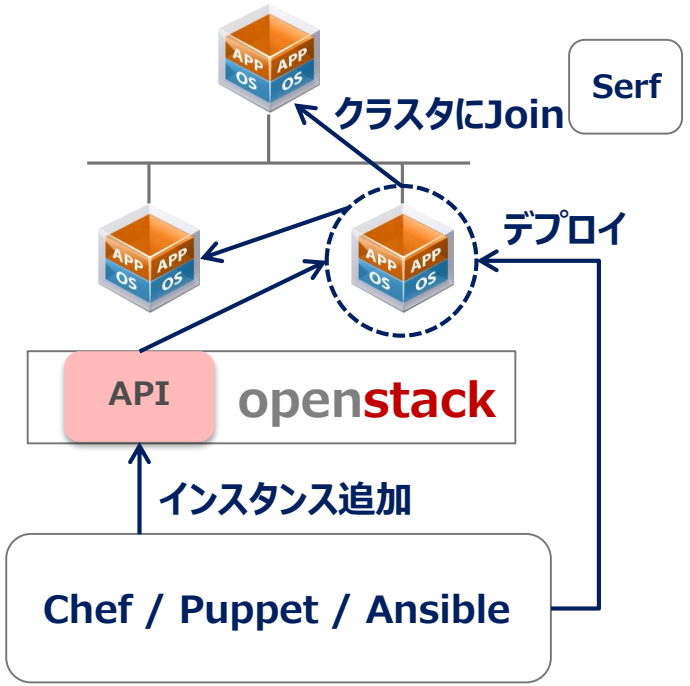
高度で複雑な Cloud Native システム

アプリケーション自身はCloud APIを使用しない

基本的な**アプリケーションのデザイン**は従来と変わらないため、スケーリング等の自動化にはオーケストレーションツールなどが必要となる

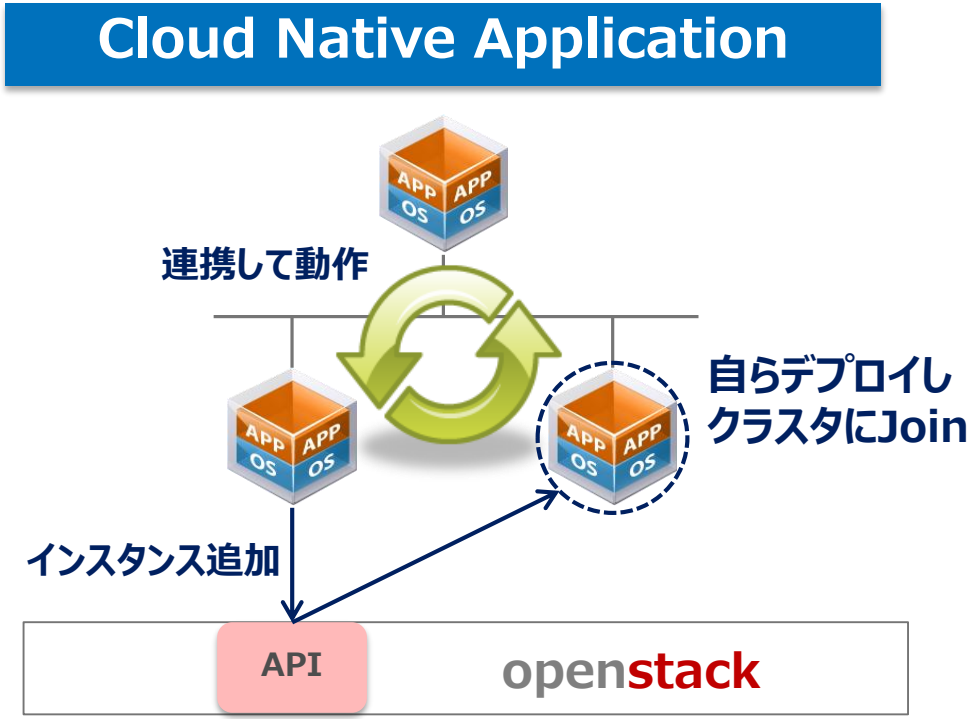
Cloud Native なシステムは運用・管理が高度で複雑なものとなる

Cloud Native システムのイメージ



CTCが考える “真の Cloud Native”

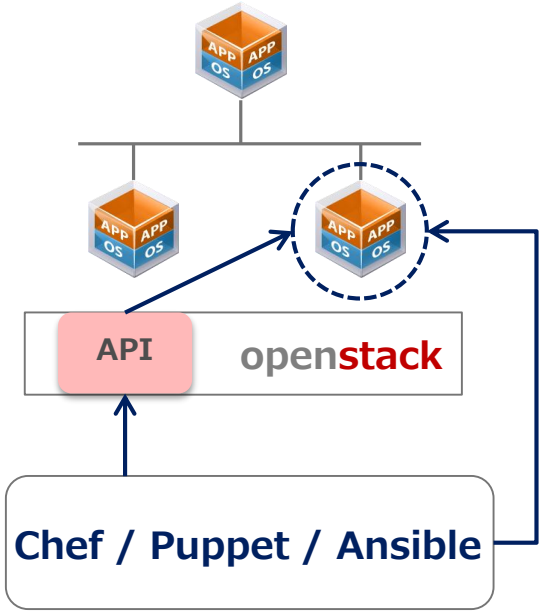
アプリケーションが自らCloud APIをコントロールし、**自律的にスケールするデザイン**にする



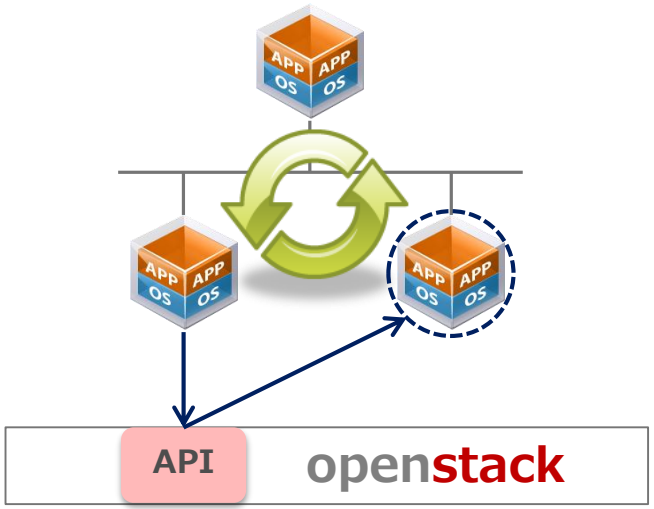
“Cloud Native Application” を実現する “RACK”

どうやって “Cloud Native Application” を実現するか？

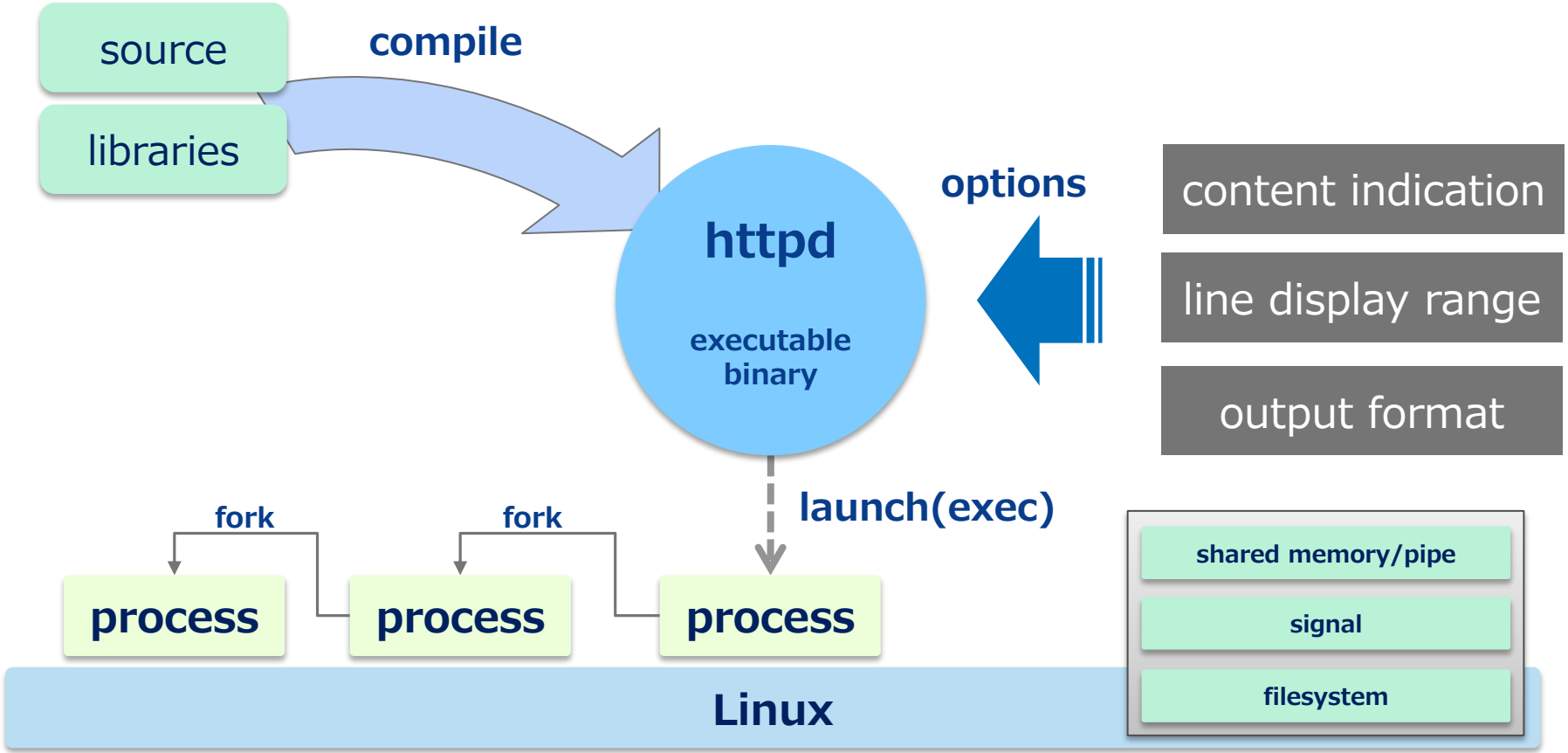
Non Cloud Native Application



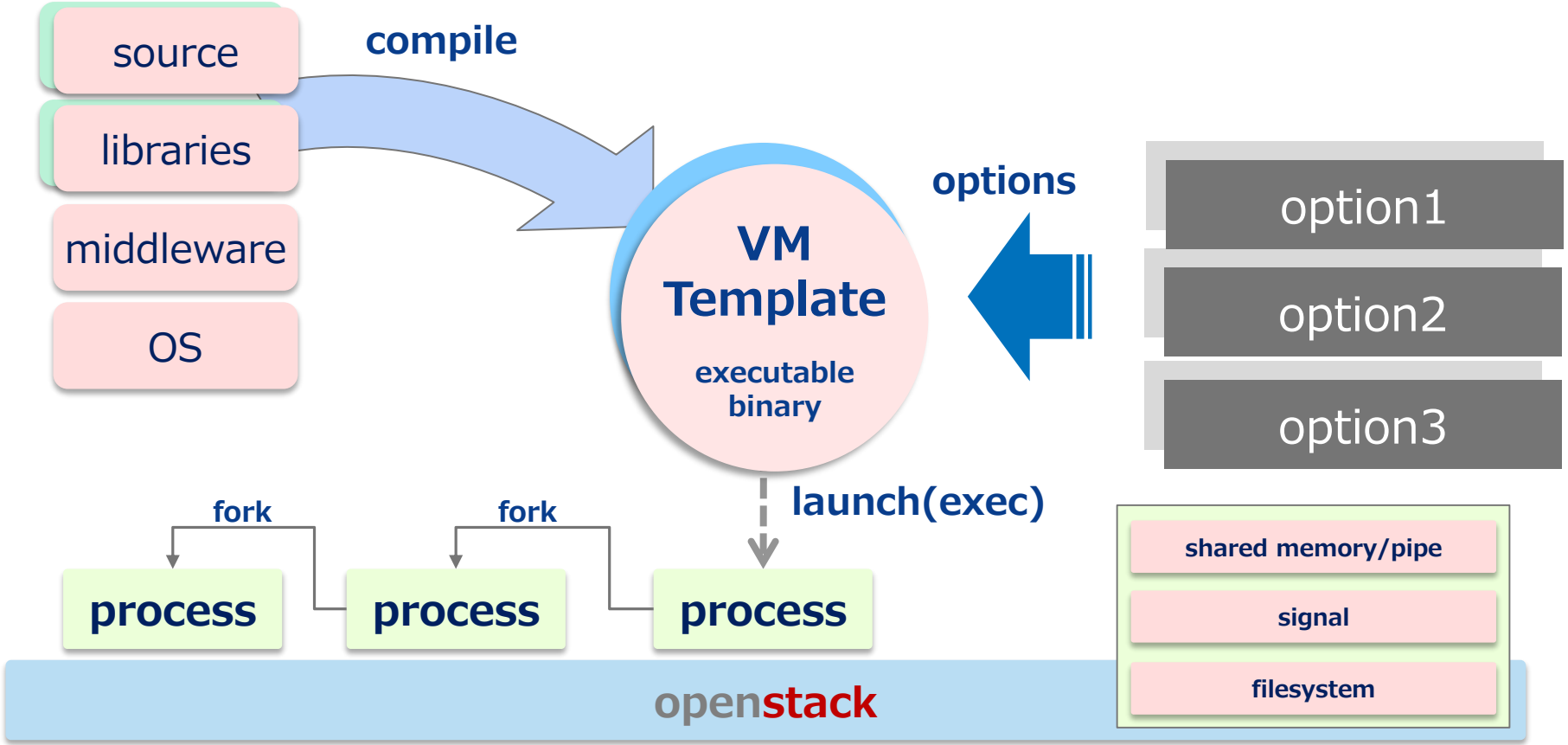
Cloud Native Application



Linux プロセスモデル



Cloud Native Applicationのモデル



RACK -Real Application Centric Kernel-

- “Cloud Native Application” の実行環境の提供
- シンプルで簡単なプログラミング環境の提供

✓ リソースの抽象化

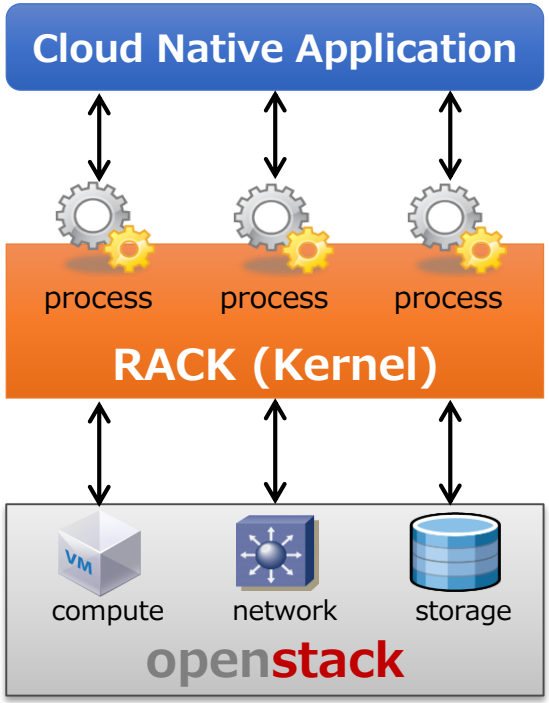
VM、ネットワーク、ストレージといった個別のリソースを抽象化し、Unixライクな “process” としてプログラムから操作可能にする

✓ process 起動確認

process(VM) 内部のアプリケーションが正常に動作しているかを確認する。プログラム上では隠蔽される

✓ process 間連携機能の提供

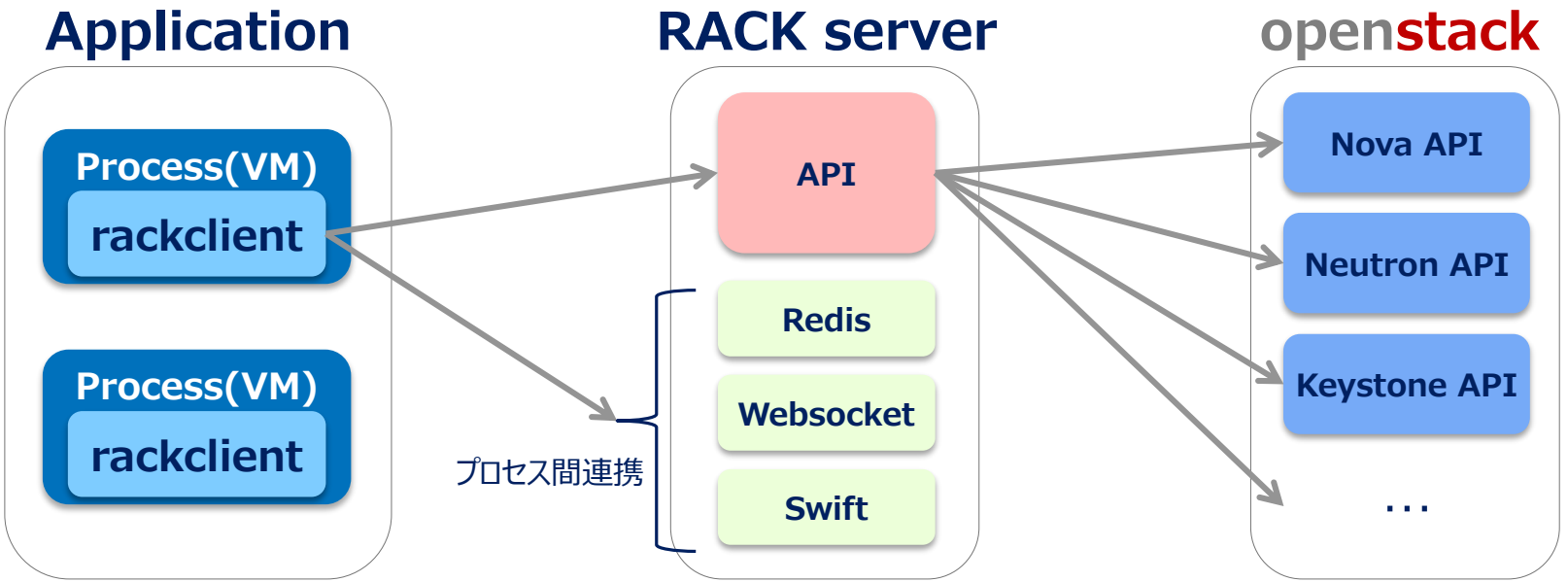
process 間でのデータ共有、シグナル通知といった機能をプログラムから操作可能にする



RACKのアーキテクチャ

RACK の構成要素

- ✓ API
- ✓ ライブラリ
- ✓ (プロセス間連携に利用するソフトウェア群)

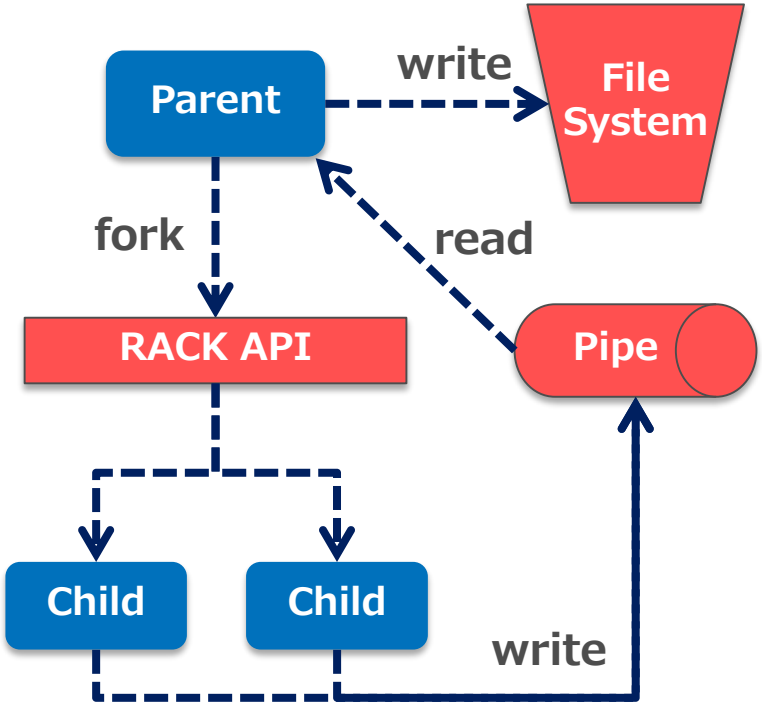


RACK を使ったプログラミング例（Python）

```
def parent(args_list):
    children = fork(args_list)
    results = pipe.read()
    file.write('result.txt', results)

def child(args):
    result = something(args)
    pipe.write(result)

if __name__ == '__main__':
    if not ppid:
        parent(args_list)
    else:
        child(args)
```

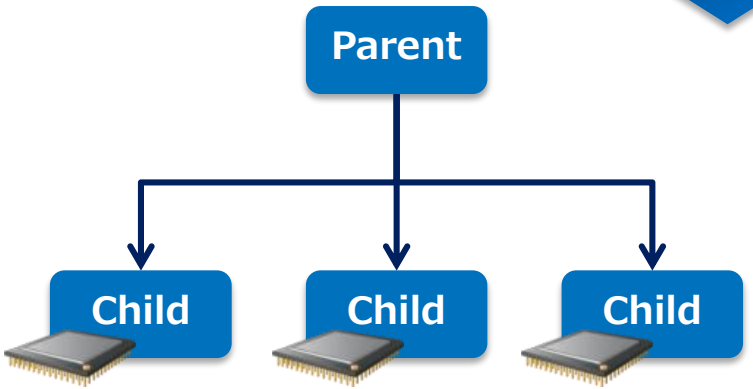


“Cloud Native Application” のデモ

円周率近似値計算アプリケーション

- ✓ モンテカルロ法というシミュレーション手法を利用する
- ✓ 実行過程で大量の乱数生成処理が発生する

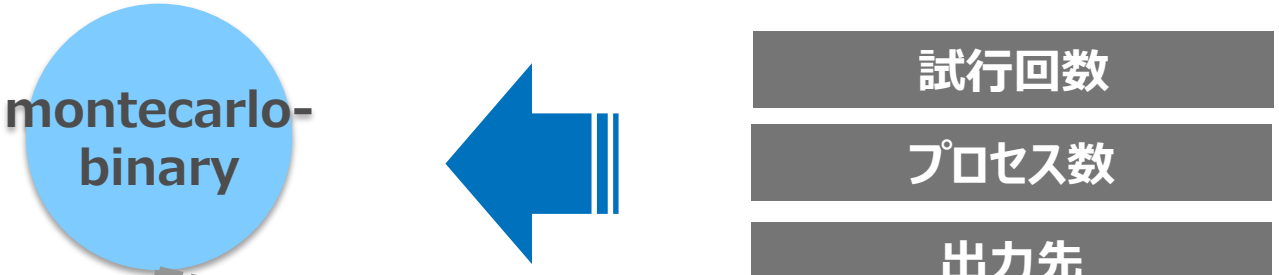
大量のCPUリソースを必要とする



CPUリソースをスケールアウトする

- ✓ 自律的にスケールアウト(fork)する
- ✓ 並列処理で高速に動作する
- ✓ 処理が終われば自らkillする

円周率近似値計算アプリケーション



親プロセス

- ✓ 子プロセスをForkする
- ✓ 子プロセスの終了を待つ
- ✓ 子プロセスの出力結果を集計してレポートを出力する

子プロセス

- ✓ シミュレーションを実行する
- ✓ 結果を親に通知する

実行と結果

実行

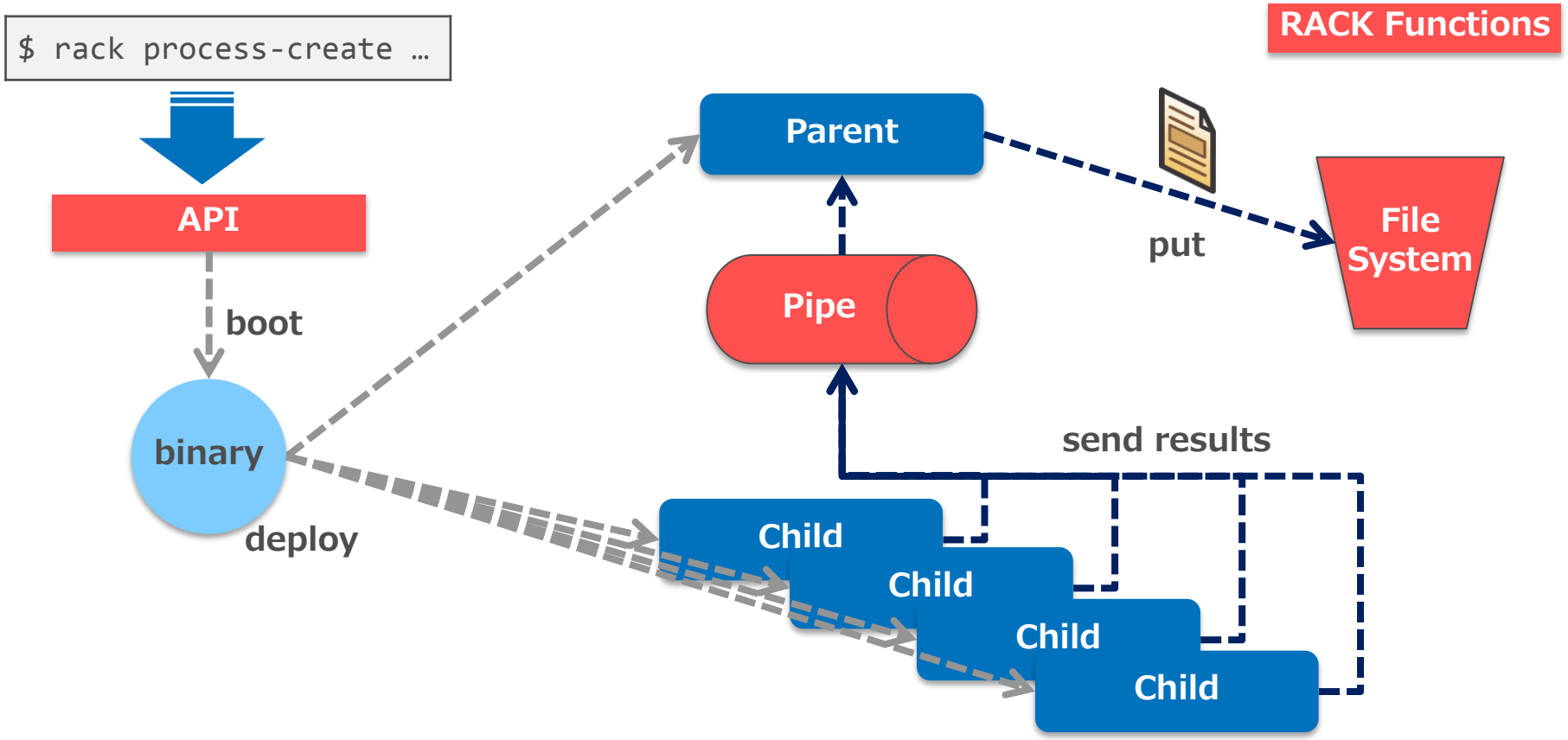
```
rack process-create ¥
  --image montecarlo ¥
  --args ¥
    trials=1000000, ¥
    workers=3, ¥
    stdout=/output/result.txt
```

結果

```
+----------+-------------------+
| Property | Value             |
+----------+-------------------+
| trials   | 1000000           |
| workers  | 3                 |
| points   | 785444            |
| pi       | 3.14159265359     |
| result   | 3.141776          |
| error    | 0.00018334641     |
| time     | 63.4065971375     |
+----------+-------------------+
```

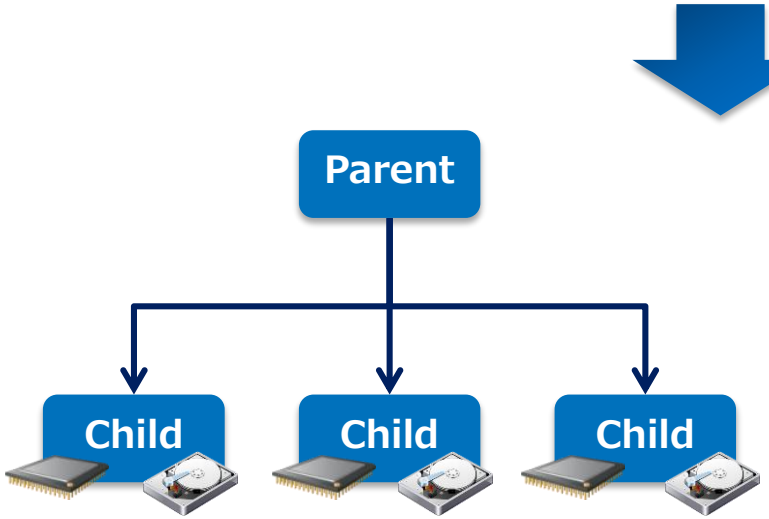
Demonstration

処理フロー


$ rack process-create …
RACK Functions
API
boot
binary
deploy
Parent
Pipe
put
File System
send results
Child
Child
Child
Child

分散ファイル解析アプリケーション

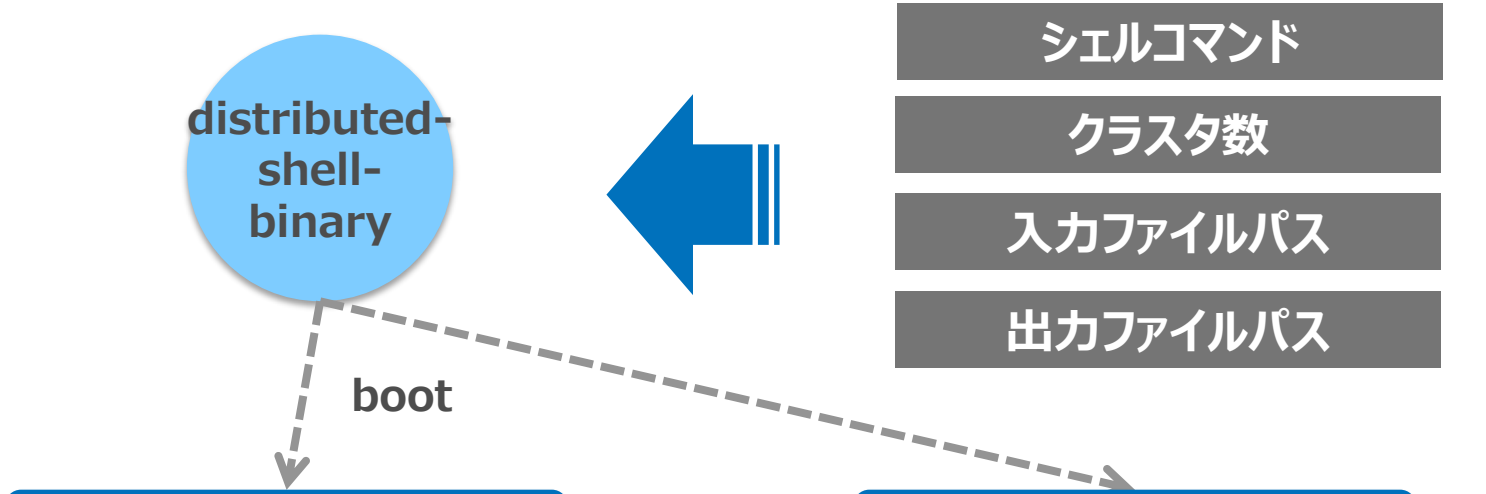
- ✓ 膨大な数のファイルを一度に処理する
- ✓ grepやsedといった簡単なコマンドをクラウドスケールで実行する



CPUリソースのスケールアウト
IOの分散

- ✓ 自律的にスケールアウト(fork)する
- ✓ 並列処理で高速に動作する
- ✓ 処理が終われば自らkillする

分散ファイル処理アプリケーション



- ✓コマンド解析
- ✓子プロセスをFork
- ✓子プロセスの処理を待つ

- ✓ファイルを取得
- ✓コマンドを実行してファイルを処理
- ✓結果を出力

実行と結果

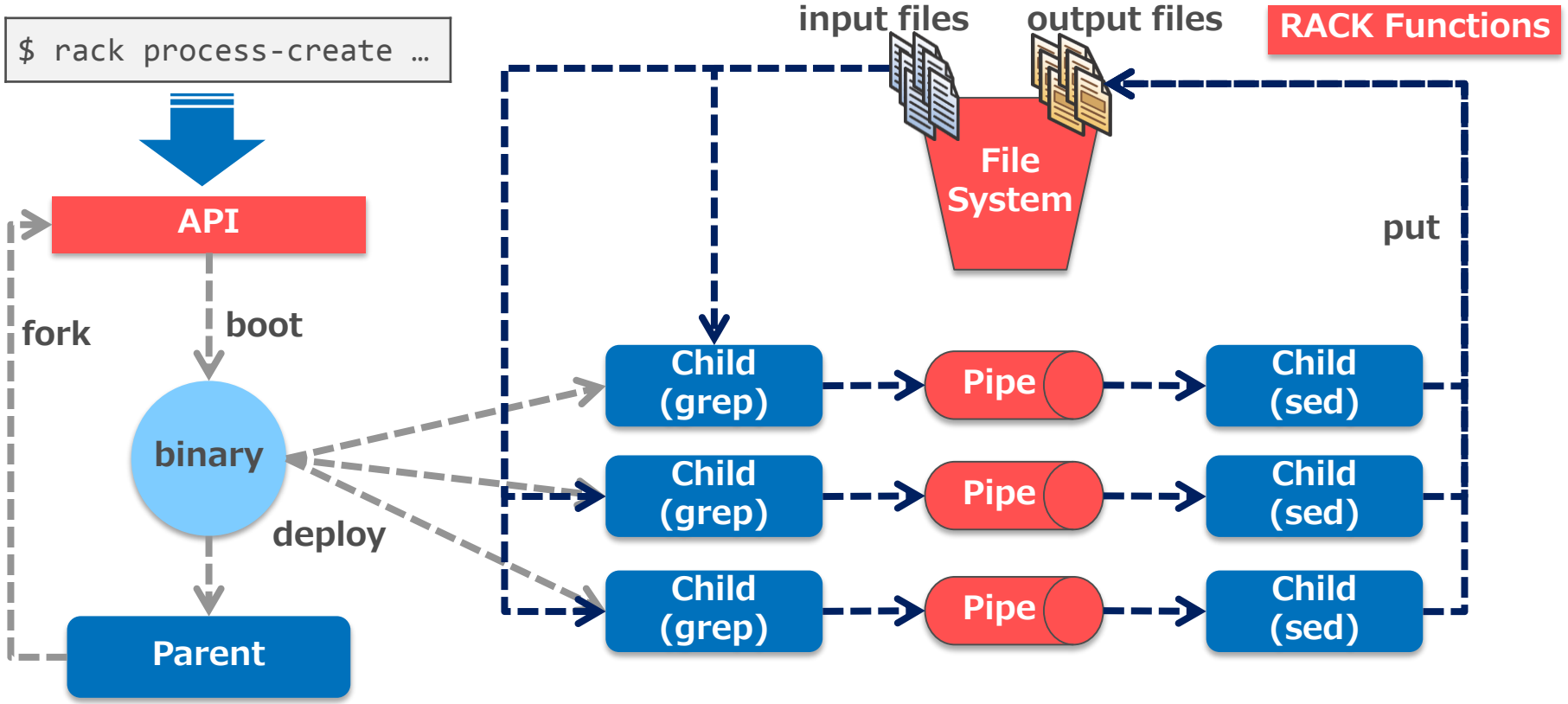
実行

```
rack process-create ¥
  --image distributed-shell ¥
  --args ¥
    command=‘grep foo | grep bar’, ¥
    stdin=/input, ¥
    stdout=/output
```

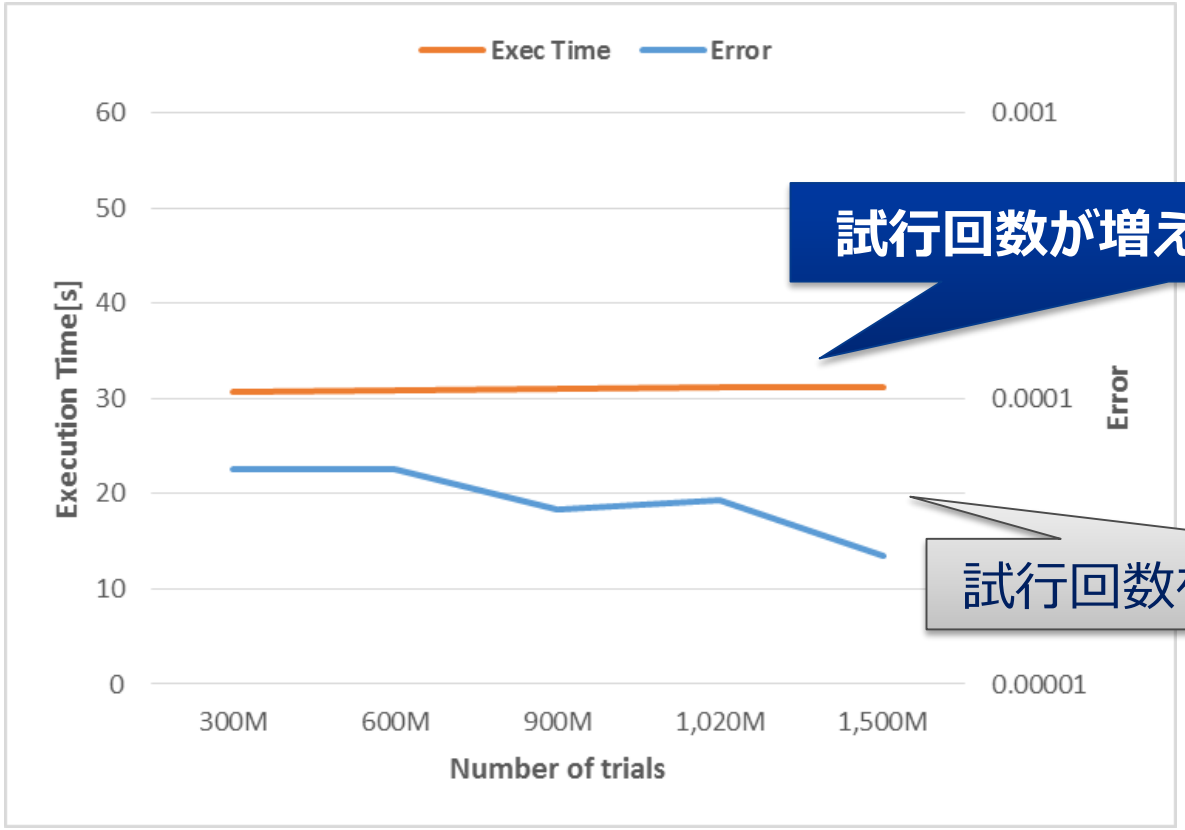
結果

```
foo bar
foo bar
foo bar
foo bar
…
```

処理フロー


$ rack process-create …
API
fork
boot
binary
deploy
Parent
input files
output files
RACK Functions
File System
put
Child (grep)
Child (grep)
Child (grep)
Pipe
Pipe
Pipe
Child (sed)
Child (sed)
Child (sed)

パフォーマンス（円周率近似値計算アプリケーション）

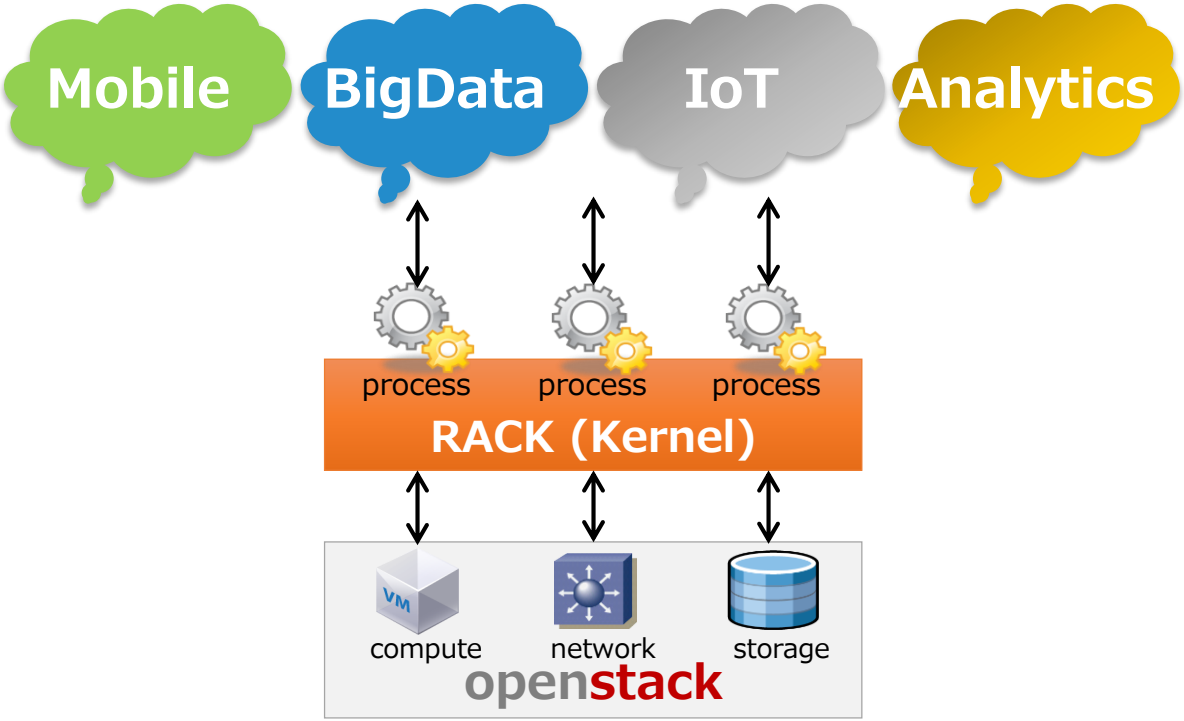


試行回数が増えても実行時間は一定

試行回数を増やすほど精度が向上

RACKの可能性

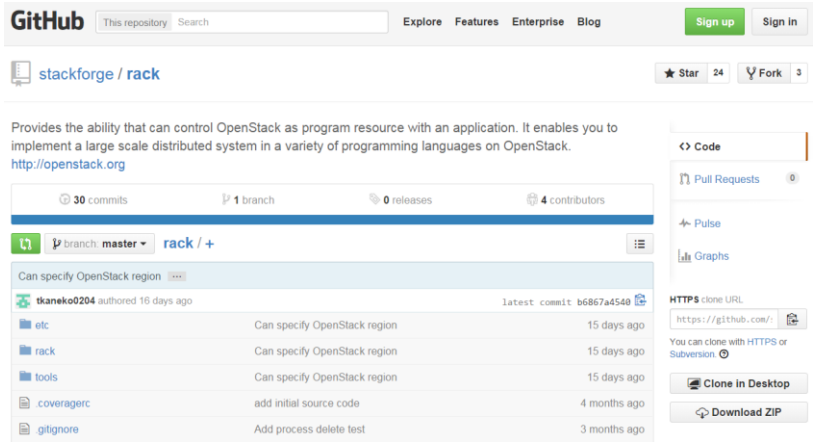
- ✓ RACKはその名の通り “Kernel” としての役割を果たす
- ✓ その上で動かすアプリケーションは自由に開発することができる



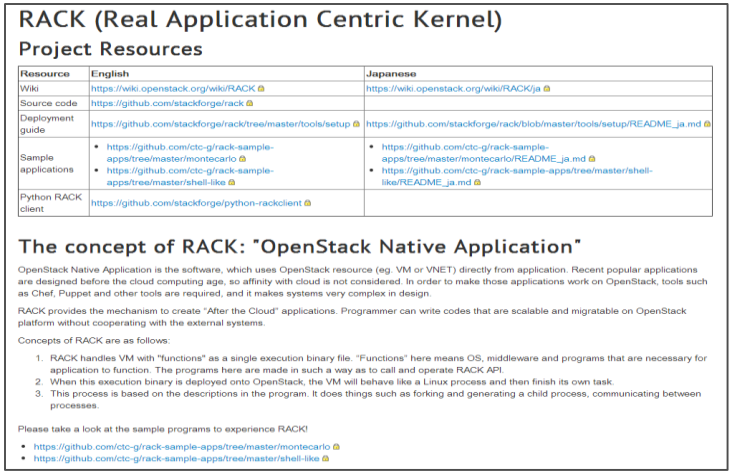
コミュニティ活動

OpenStack Community

- ✓ OpenStack プロジェクトを運営している OSS コミュニティ
- ✓ ソースコード、レビュー管理システムなどの開発関連ツールの提供や、OpenStack Summit の運営などを行っている
- ✓ RACK もこのコミュニティで開発を行っており、**誰でも開発に参加可能**である



https://github.com/stackforge/rack



RACK (Real Application Centric Kernel)

Project Resources

Resource	English	Japanese
Wiki	https://wiki.openstack.org/wiki/RACK	https://wiki.openstack.org/wiki/RACK/ja
Source code	https://github.com/stackforge/rack	
Deployment guide	https://github.com/stackforge/rack/tree/master/tools/setup	https://github.com/stackforge/rack/blob/master/tools/setup/README_ja.md
Sample applications	• https://github.com/ctc-g/rack-sample-apps/tree/master/montecarlo • https://github.com/ctc-g/rack-sample-apps/tree/master/shell-like	• https://github.com/ctc-g/rack-sample-apps/tree/master/montecarlo/README_ja.md • https://github.com/ctc-g/rack-sample-apps/tree/master/shell-like/README_ja.md
Python RACK client	https://github.com/stackforge/python-rackclient	

The concept of RACK: "OpenStack Native Application"

OpenStack Native Application is the software, which uses OpenStack resource (eg. VM or VNET) directly from application. Recent popular applications are designed before the cloud computing age, so affinity with cloud is not considered. In order to make those applications work on OpenStack, tools such as Chef, Puppet and other tools are required, and it makes systems very complex in design.

RACK provides the mechanism to create "After the Cloud" applications. Programmer can write codes that are scalable and migratable on OpenStack platform without cooperating with the external systems.

Concepts of RACK are as follows:

1. RACK handles VM with "functions" as a single execution binary file. "Functions" here means OS, middleware and programs that are necessary for application to function. The programs here are made in such a way as to call and operate RACK API.
2. When this execution binary is deployed onto OpenStack, the VM will behave like a Linux process and then finish its own task.
3. This process is based on the descriptions in the program. It does things such as forking and generating a child process, communicating between processes.

Please take a look at the sample programs to experience RACK!

- https://github.com/ctc-g/rack-sample-apps/tree/master/montecarlo
- https://github.com/ctc-g/rack-sample-apps/tree/master/shell-like

https://wiki.openstack.org/wiki/RACK

OpenStack Summit 2014 Paris

- ✓ OpenStack Summit で RACK を発表
- ✓ 海外の開発者たちからの注目、評価を得た

https://www.openstack.org/summit/openstack-paris-summit-2014/session-videos/presentation/the-road-to-a-openstack-native-application-what-if-vms-are-treated-as-linux-processes

Cloud Native Hackathon

- ✓ Cloud Native Application/RACK に関する勉強会・ハッカソン等を企画予定
- ✓ オープンな場での意見交換やアプリケーション開発を通して、Cloud Native Application の可能性を広げていきたい

Thank you!


openstack rack
ウェブ
画像
動画
ニュース
ショッピング
もっと見る
検索ツール
約 219,000 件（0.49 秒）
RACK - OpenStack
https://wiki.openstack.org/wiki/RACK このページを訳す
The concept of RACK: "OpenStack Native Application". OpenStack Native Application is the software, which uses OpenStack resource (eg. VM or VNET) directly from application. Recent popular applications are designed before the cloud ...
stackforge/rack · GitHub
https://github.com/stackforge/rack このページを訳す
2014/10/06 - rack - Provides the ability that can control OpenStack as program resource with an application. It enables you to implement a large scale distributed system in a variety of programming languages on OpenStack.
このページに複数回アクセスしています。前回のアクセス: 14/10/29