



学際大規模共同利用環境を想定したクラウド基盤ミドルウェアの運用性向上に関する研究

1. 研究の目的

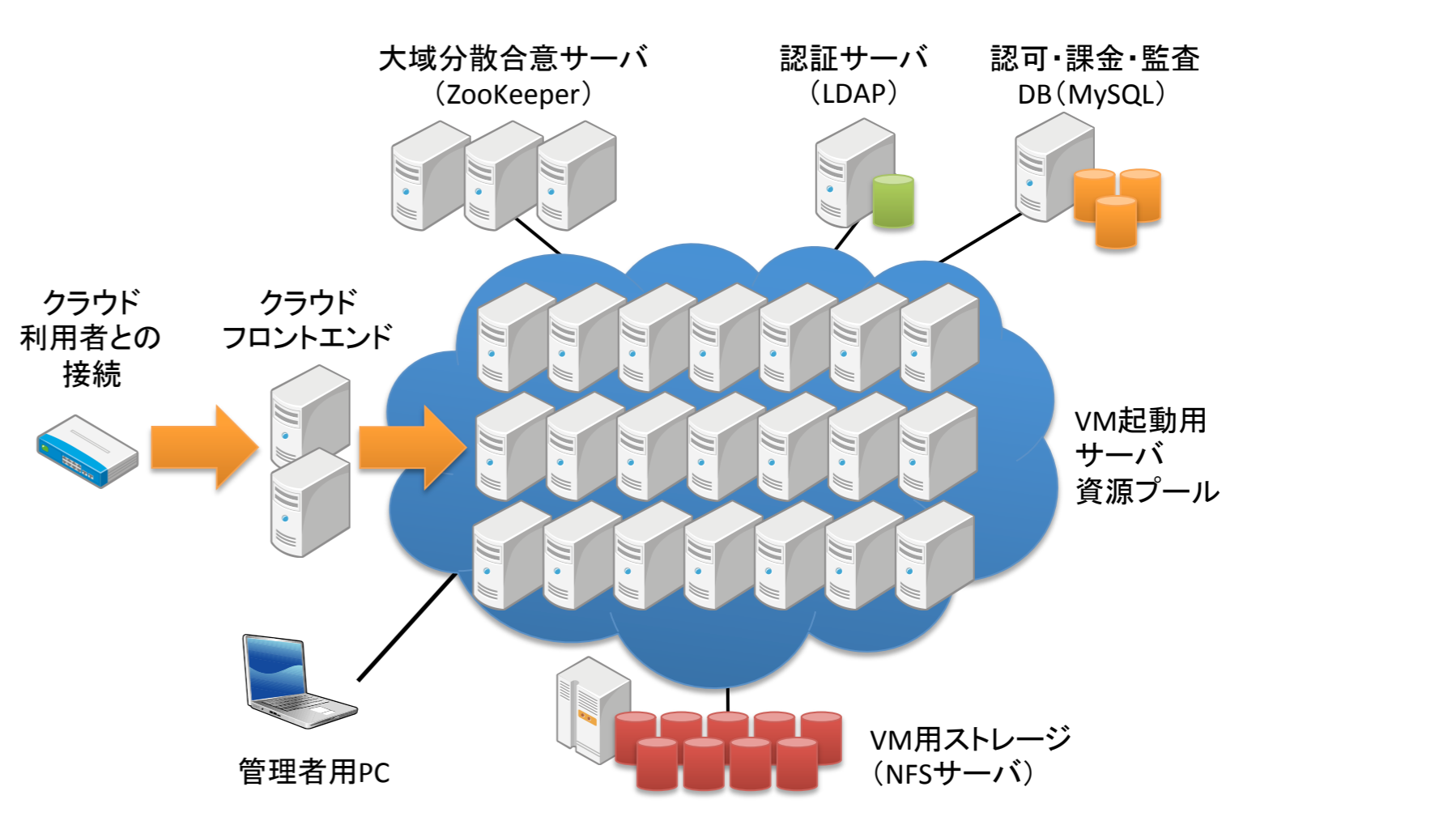
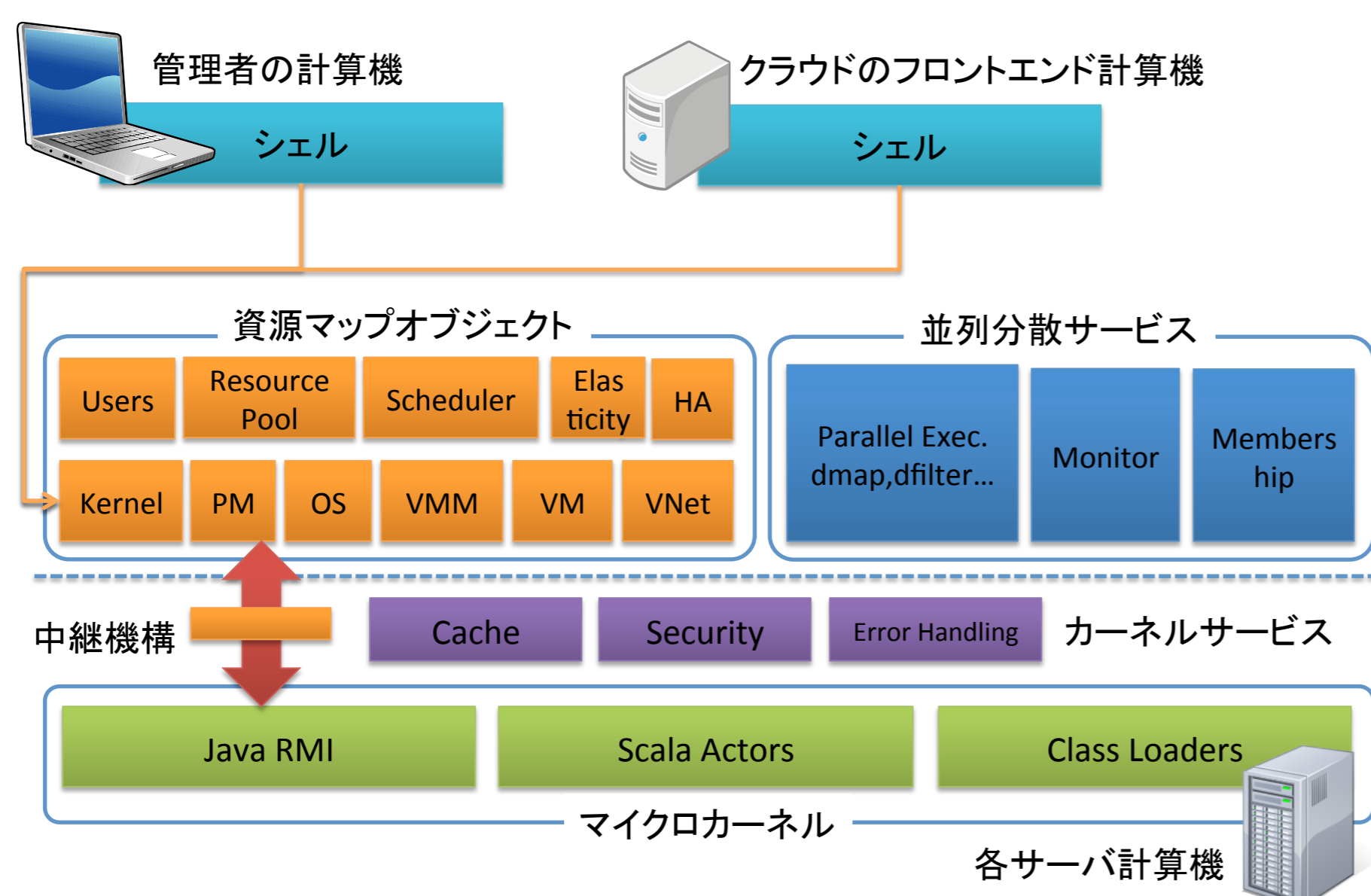
本研究では、学際大規模情報基盤共同利用環境を想定したクラウド基盤ミドルウェアの運用性向上を目指す。本研究は既存のミドルウェアの一部の改良に留まらず、独自のミドルウェアを開発し、ミドルウェアのアーキテクチャ設計からの改善を目指す。また本研究は、単なるシステム実装の工夫による運用性向上を目的とするだけでなく、数理的検証を専門とする研究者との学際的な共同研究により、信頼性の高いミドルウェアの構築を目指す。さらに、学際的な多様な要望への対応や、管理容易性の向上を含めた総合的な運用性の実現を目指す。

2. 基盤技術

拡張可能なクラウド基盤ミドルウェア「Kumoi」

- ・オペレーティングシステム(OS)の設計方法を活用
- ・ミドルウェアの中核となる機能を最小化
- ・多くのクラウドとしての機能をスクリプティングで実現(メタミドルウェア)

```
def schedule(v: ColdVM,
pms: List[HotPhysicalMachine]) = {
val candidates = pms.filter(_.availFor(v))
val sorted = candidates.sortWith(rank)
val top = sorted.head
top.vmm.add(v)
}
```



3. 昨年度の成果

- ・キャッシュ機構の導入
- ・セキュリティ機構の導入
- ・透過的な障害対策
- ・仮想ネットワーク(OpenFlow)対応
- ・VM内アプリケーション(PaaS)対応

```
@remote trait HotPhysicalMachine {
@persistcache def name(): String
@persistcache def addr(): InetAddress
@persistcache def os(): HotOS
@persistcache def info(): List[PMInfo]
@cache def stats(): List[PMStat]
@cache(3000) def cpuRatio(): Double
@cache(3000) def cpuAvailable(): Double
@cache(3000) def freeMemory(): Long
@cache(3000) def freeMemory(): Long
@persistcache def maxMemory(): Long
}
@persistcache def vmm(): VMM
@cache def vms(): List[HotVM]
@invalidate @nocache def shutdown()
@invalidate @nocache def restart()
}
```

キャッシュポリシーの記述

```
kumoi> p(account)
User UID Resource Consumed
-----
sugiki c56f8105-8e98-4f12-bd16-3c74d9140017 vm.write.xvda 1008640
sugiki c56f8105-8e98-4f12-bd16-3c74d9140017 vm.tx.vf11.0 0
sugiki c56f8105-8e98-4f12-bd16-3c74d9140017 vm.rx.vf11.0 520
sugiki c56f8105-8e98-4f12-bd16-3c74d9140017 vm.read.xvda 41383936
sugiki c56f8105-8e98-4f12-bd16-3c74d9140017 vm.cputime 11774809255
```

課金ログの記録

```
kumoi> p(audit)
Date Host User UID Category Class Method Args Result Detail
-----
2012-10-09 18:10:11.0 ibm1/10.0.0.1:10100 sugiki
c2ca61bd-c7b0-4639-a5eb-f933e48ebf8c Index
kumoi.shell.pm.HotPhysicalMachine ps
List(uid=sugiki,ou=People,dc=oss,dc=cs,dc=tsukuba,dc=ac,dc=jp) true null
```

監査ログの記録

```
kumoi> pms.filter(_.cpuRatio < 0.3)
.filter(_.freeMemory > 512*1024*1024)
.take(5)
.zipWithIndex
.foreach(deployVM)
```

透過的な障害対策が可能な場合のスクリプト記述

```
kumoi> ofc.add({ case PacketIn(sw, id,
inPort, totalLen, reason, data, ctx) =>
sw.send(PacketOut(bufferId, inPort,
data, List(Output(Port.Flood, 0))),
ctx)
})
```

OpenFlowのルール記述

```
kumoi> pms.map(_.vms.map { vm =>
val apache = vm.local.apps(0)
.asInstanceOf[Apache]
apache.maxClients = 1024
apache.keepAliveTimeout = 5
apache.reload()
})
```

VM内アプリケーションの操作記述

※平成24年度JHPCN採択研究課題「クラウド基盤ミドルウェアのスケラビリティ向上に関する研究」

4. 本年度の実施予定

(1) JHPCN環境を利用した大規模システム評価

計画案: 並列高速VMデプロイチャレンジ(カスタマイズ付)
本ミドルウェアが提供するスクリプティングの柔軟性により、個別カスタマイズしたVMを高速配置

(2) 継続的なミドルウェア改良

透過的に障害回避を行うクラウド資源操作フレームワーク
本ミドルウェアに必ずしも依存しない方式で実現を目指す

(3) HPCI-JHPCNで運用されているクラウドとの連携の検討

本ミドルウェアを検証用のプロトタイプシステムとして活用
本研究で蓄積したノウハウを応用する