

# ヘテロジニアスなサーバ計算機環境における アプリケーションの特徴に基づいた 仮想マシン再配置手法

藤井淳(電気通信大学)

和田康孝(早稲田大学)

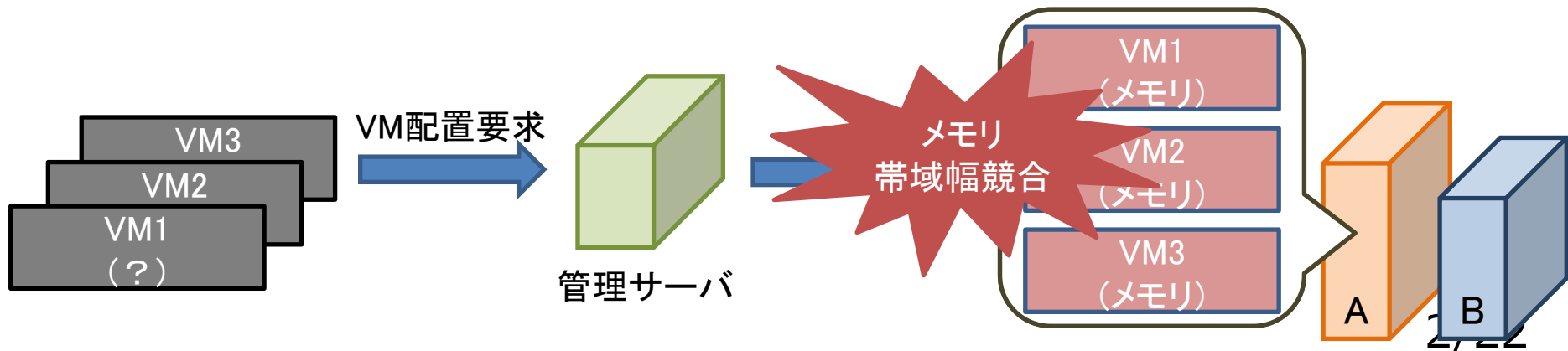
近藤正章(東京大学)

本多弘樹(電気通信大学)

# 研究背景

- データセンタの環境
  - 1台の物理サーバ上で複数の仮想マシン(VM)が動作
- VM上で実行するアプリケーション
  - 様々な特徴(メモリバウンド,CPUバウンドなど)のアプリケーション
- VM配置方法
  - 配置するVM上で実行するアプリケーションの特徴は通常不明

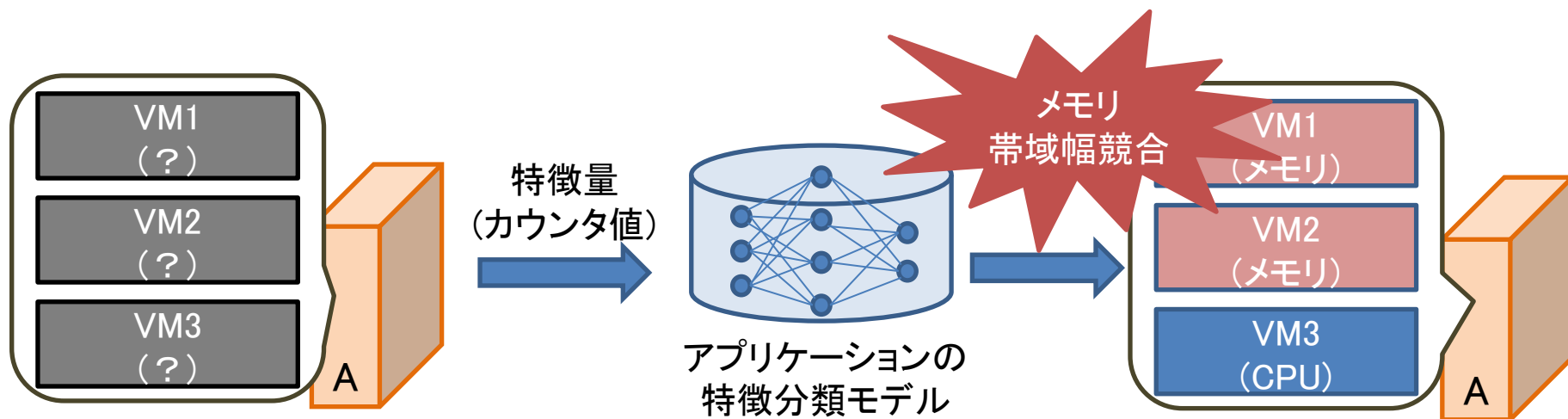
リソース競合を検知し再配置を行うことで性能改善が見込める



# VM間でのリソース競合の検知

- VM上で動作するアプリケーションの特徴をクラス分類するモデルを事前に作成
  - 共有リソースを使用する特徴のアプリケーションを実行するVMが同じサーバに配置されている
    - リソース競合が発生しているとみなす

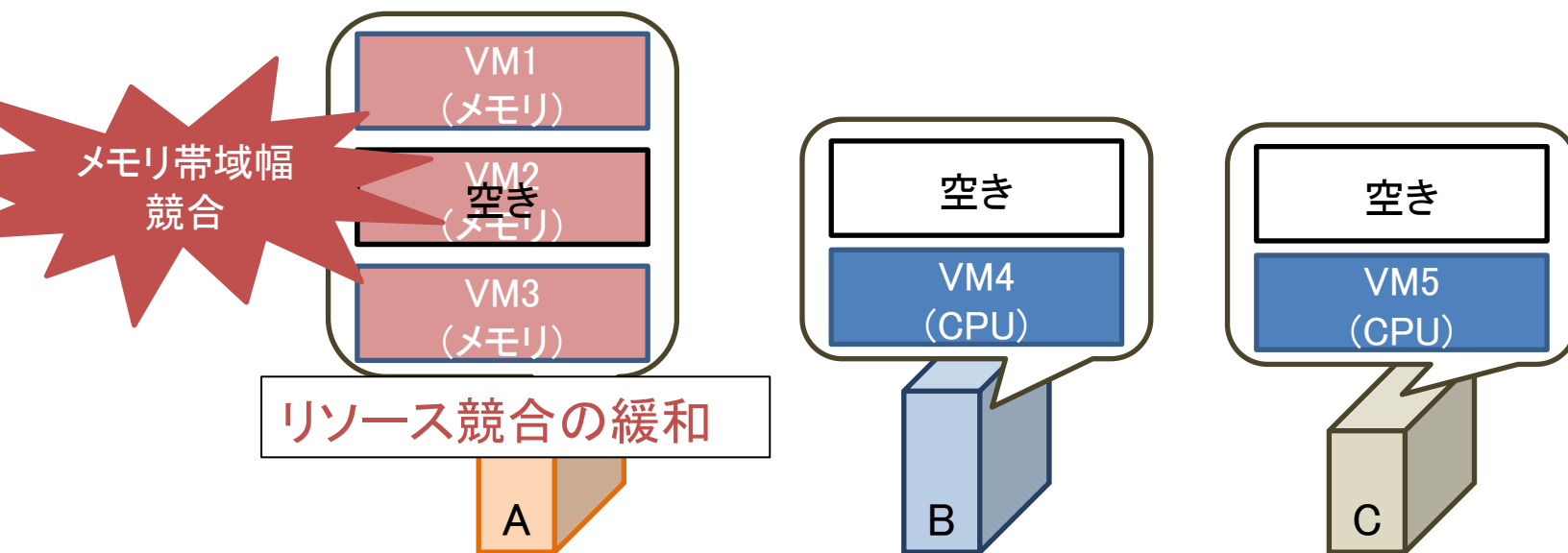
リソース競合が発生しているサーバをマイグレーション元



# 仮想マシン再配置

- リソース競合発生しているサーバをマイグレーション元
- 異なる性能のサーバが混在するヘテロジニアスな環境[1]
  - アプリケーションごとに最大の処理性能が得られるサーバが異なる  
(アプリケーションごとにサーバとの相性が存在)

## 相性を考慮してマイグレーション先を決定



# 研究概要

- 研究目標

同時に配置するVM間のリソース競合と  
アプリケーションとサーバの相性を考慮した再配置手法を提案し  
データセンタ全体のスループットの向上を目指す

- アプローチ

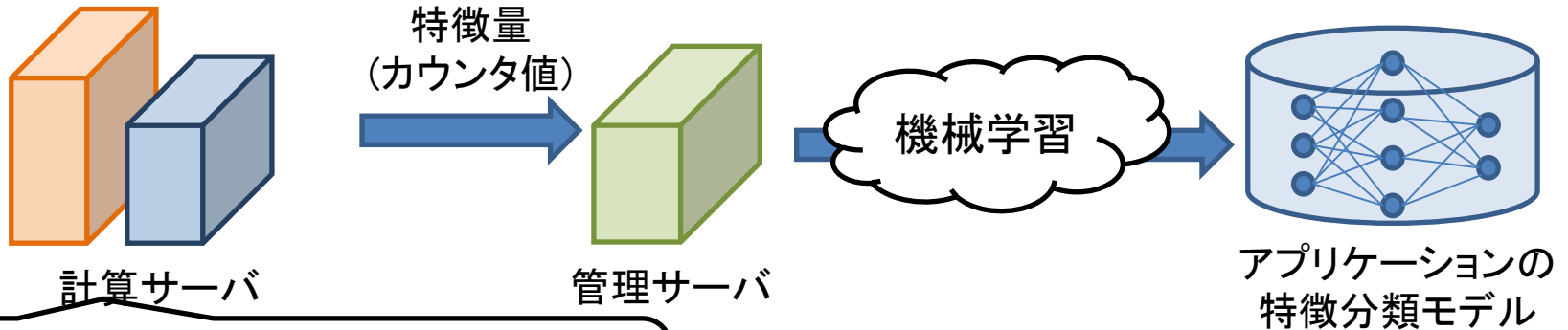
アプリケーションの特徴からリソース競合と相性の判断モデルを**事前**に作成  
リソース競合が発生するVMがサーバ間で**均一**になるように再配置  
マイグレーション先が複数ある場合には最も相性が良いサーバを選択

アプリケーションの特徴に着目した  
仮想マシン再配置手法

# 提案手法: 運用前のモデルの作成

運用前

アプリケーションの特徴を分類し  
リソース競合を検知するための分類モデルを機械学習で作成



モデル作成用アプリケーションを実行  
・ パフォーマンスカウンタの取得

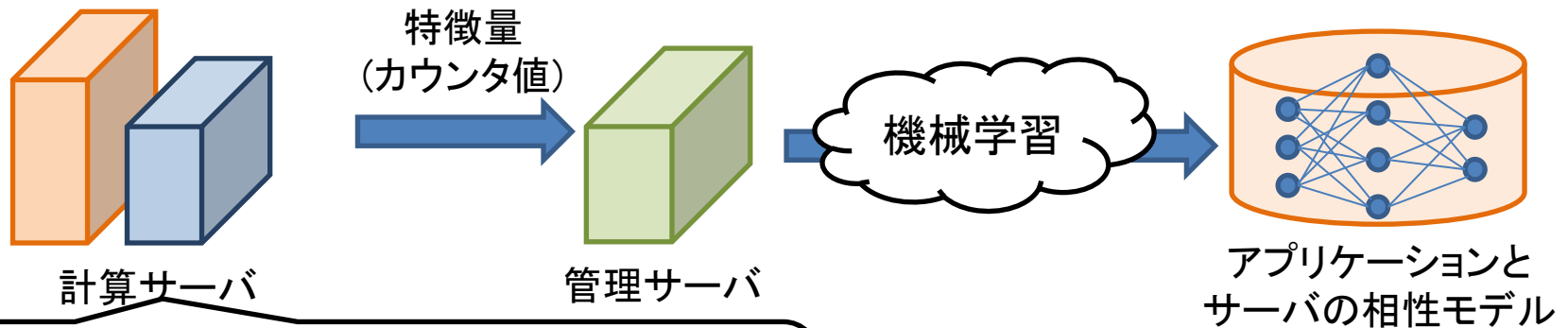
特徴量: パフォーマンスカウンタ

目的変数: アプリケーションの特徴 (メモリバウンド, CPUバウンド)

# 提案手法: 運用前のモデルの作成

運用前

アプリケーションとサーバの相性を判断するモデルを機械学習で作成



モデル作成用アプリケーションを実行

- パフォーマンスカウンタの取得
- 実行命令数、実行時間からMIPS値の計算

アプリケーションを各サーバ構成で実行した際の予測MIPS値を「相性」とする

特徴量: パフォーマンスカウンタ

目的変数: 各サーバ構成で実行した際の予測MIPS値

# 提案手法: 運用時の手順

運用時

各VM上のアプリケーションを  
特徴にクラス分類

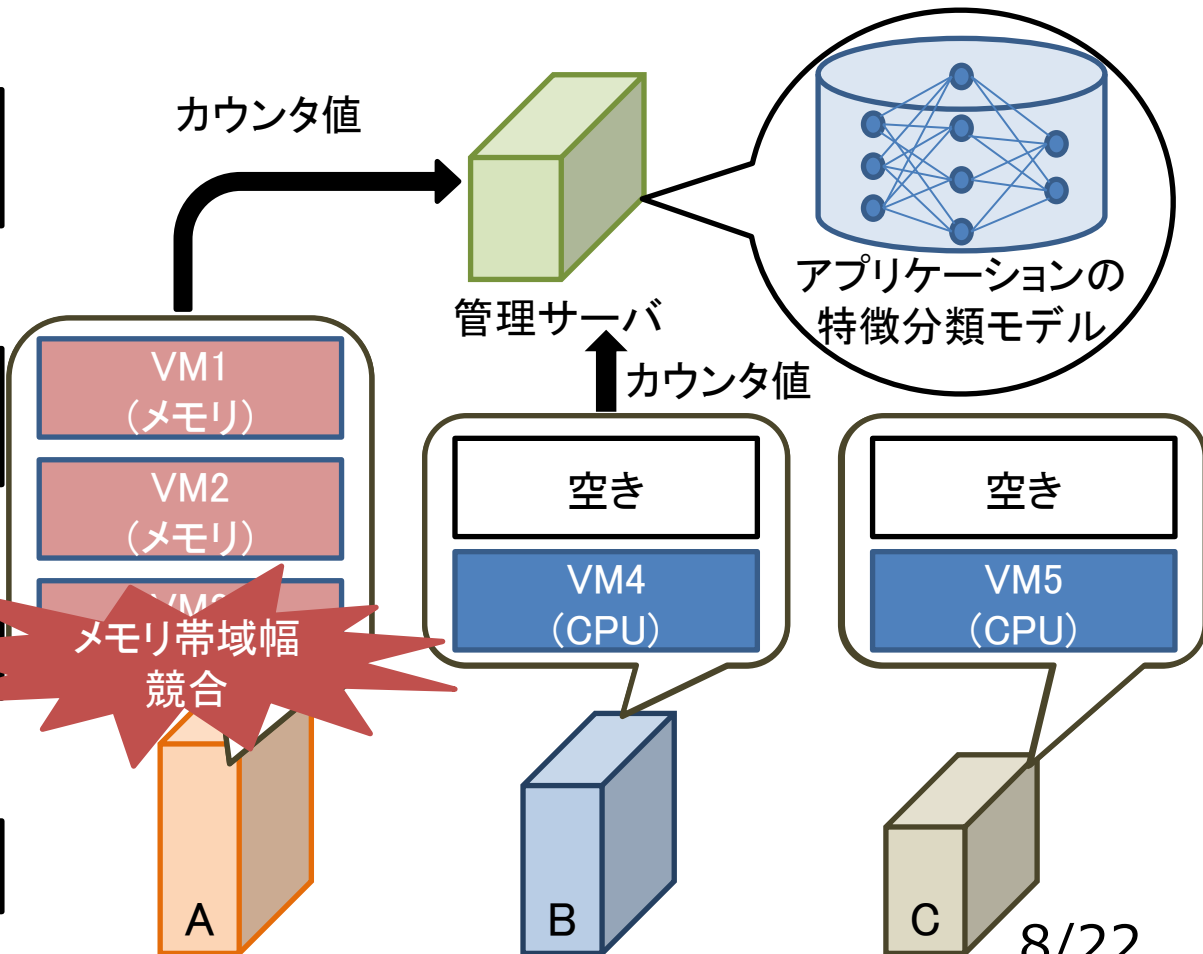
マイグレーション元と  
マイグレーション先候補の決定

VMで実行するアプリケーション  
とサーバの相性を判断

マイグレーションVMの決定

マイグレーションの実行

- 一定時間の間カウンタ値を取得
- 各サーバ上のVMで動作しているアプリケーションを特徴にクラス分類





# 提案手法: 運用時の手順

運用時

各VM上のアプリケーションを  
特徴にクラス分類

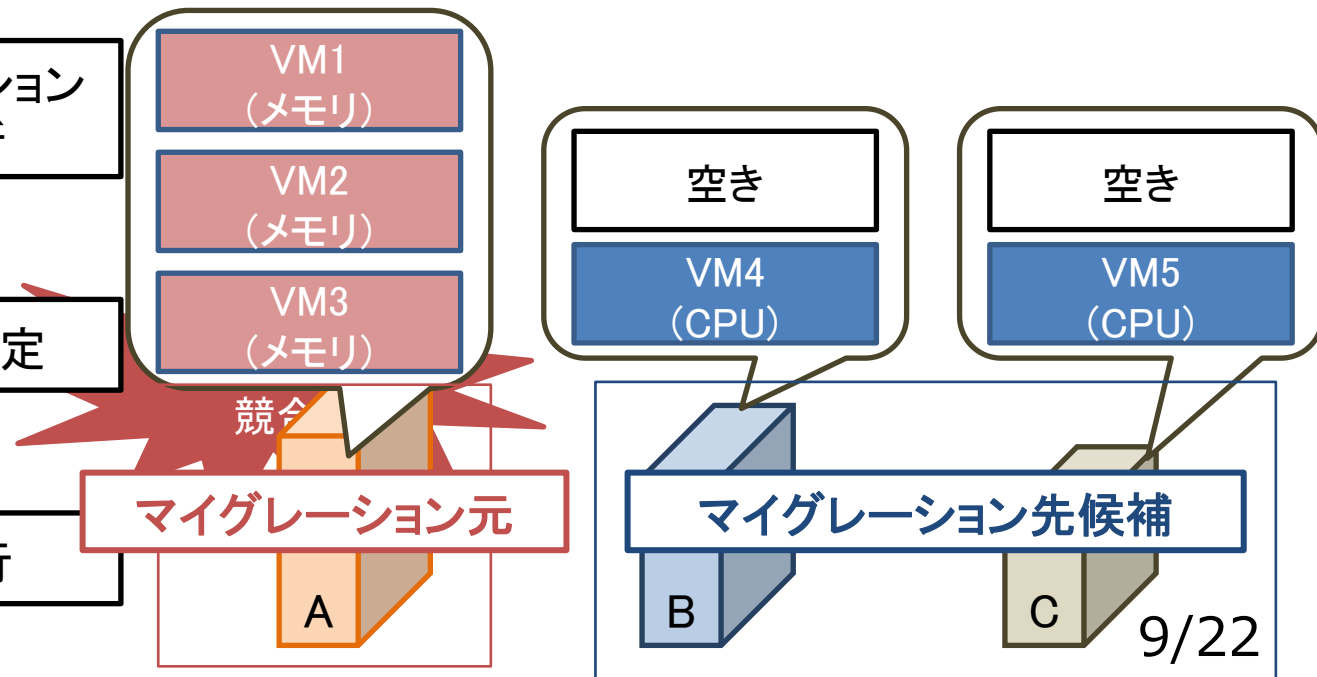
マイグレーション元と  
マイグレーション先候補の決定

VMで実行するアプリケーション  
とサーバの相性を判断

マイグレーションVMの決定

マイグレーションの実行

- リソース競合が発生するVMの数が  
( $Max - Min$ ) > 1 の際に
    - 最も多いサーバをマイグレーション元サーバ
    - 最も少ないサーバをマイグレーション先候補サーバ
- 均一にしたいので数が逆転しないようにする



# 提案手法: 運用時の手順

運用時

各VM上のアプリケーションを  
特徴にクラス分類

マイグレーション元と  
マイグレーション先候補の決定

VMで実行するアプリケーション  
とサーバの相性を判断

マイグレーションVMの決定

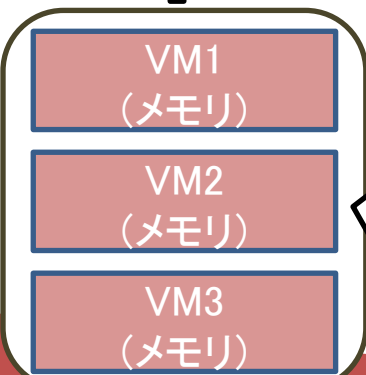
マイグレーションの実行

- マイグレーション元サーバの各VMと  
マイグレーション先サーバ間の相性を判別

カウンタ値

管理サーバ

アプリケーションと  
サーバの相性モデル



競争

マイグレーション元

	A	B	C
VM1	1.0	1.0	0.9
VM2	1.0	1.3	0.9
VM3	1.0	1.1	1.2

A

B

C

# 提案手法: 運用時の手順

運用時

各VM上のアプリケーションを  
特徴にクラス分類

マイグレーション元と  
マイグレーション先候補の決定

VMで実行するアプリケーション  
とサーバの相性を判断

マイグレーションVMの決定

マイグレーションの実行

- マイグレーション先候補毎にもっとも予測MIPS値が上昇するVMをマイグレーションVMとする

カウンタ値

管理サーバ

アプリケーションと  
サーバの相性モデル

VM1  
(メモリ)

VM2  
(メモリ)

VM3  
(メモリ)

競争

	A	B	C
VM1	1.0	1.0	0.9
VM2	1.0	1.3	0.9
VM3	1.0	1.1	1.2

サーバBにはVM2  
サーバCにはVM3をマイグレーションVMとする

# 提案手法: 運用時の手順

運用時

各VM上のアプリケーションを  
特徴にクラス分類

マイグレーション元と  
マイグレーション先候補の決定

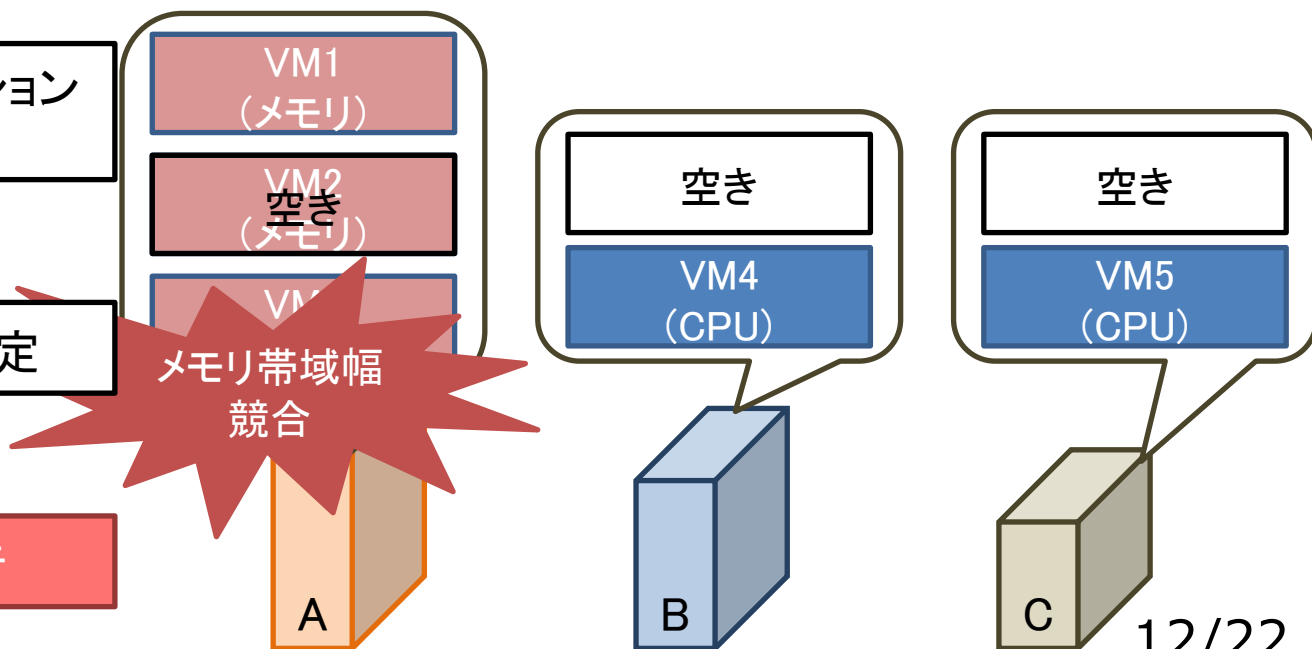
VMで実行するアプリケーション  
とサーバの相性を判断

マイグレーションVMの決定

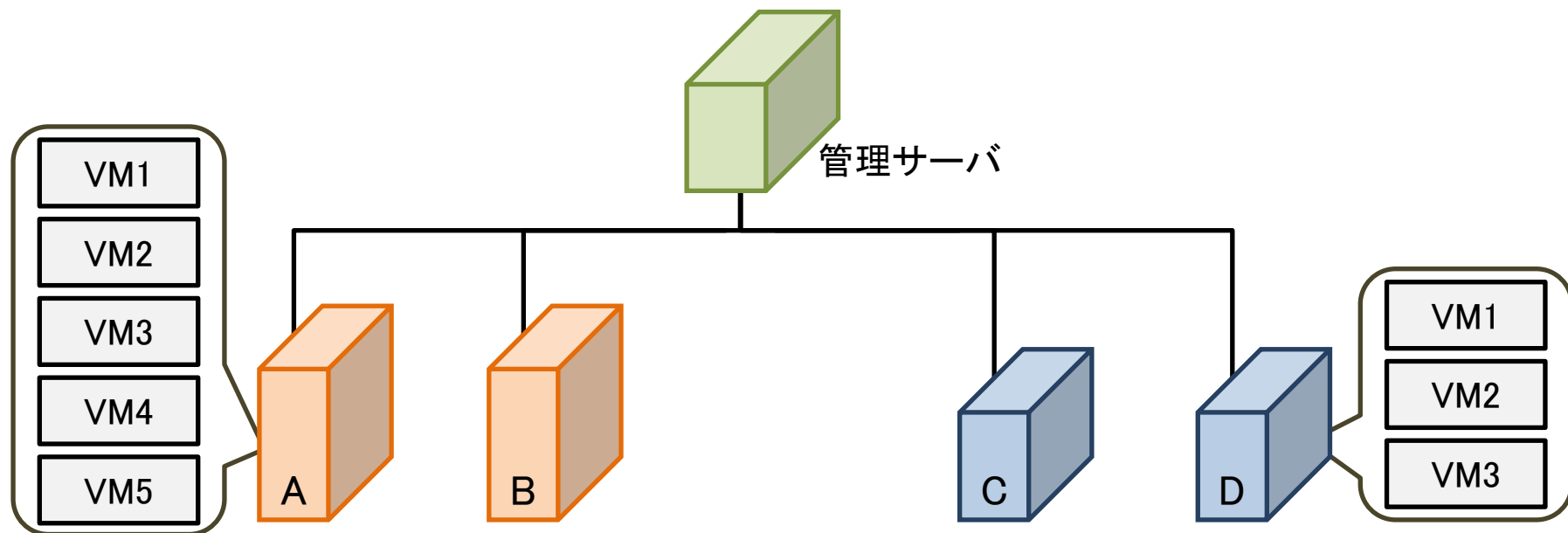
マイグレーションの実行

- VMの状況が変化していないか確認
  - 時間経過による特徴の変化の確認
  - マイグレーションVMは終了していないか
- 配置可能ならばマイグレーションを実行

リソース競合の緩和



# 評価実験: 実験サーバ環境



各サーバの性能

	CPU	CPU周波数 (GHz)	CPUコア数	メモリ	メモリ容量(GB)
サーバA,B	Xeon E5640	3.46	6	DDR3 1333MHz	12
サーバC,D	Xeon X5675	2.93	4	DDR3 1066MHz	12

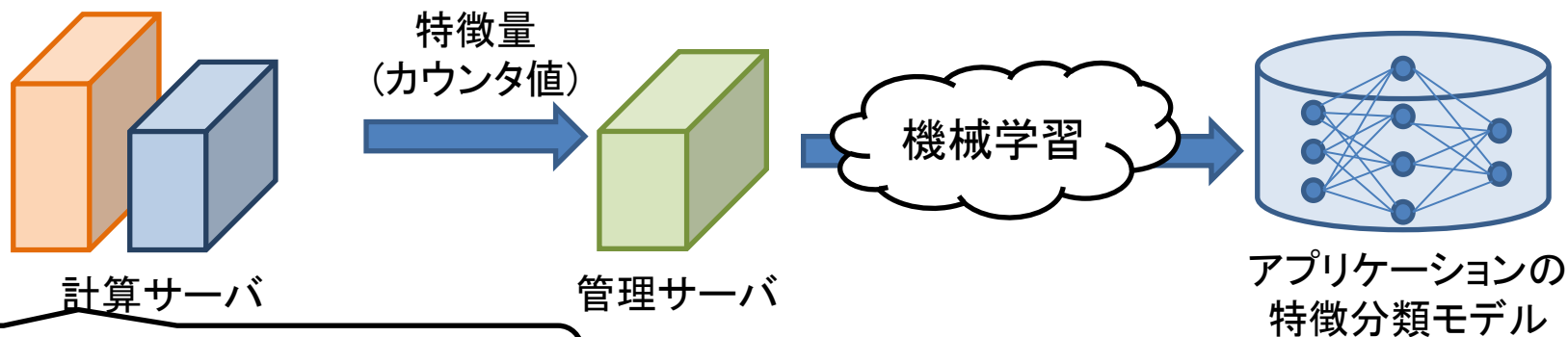
	VCPU数	メモリ(GB)
想定VMインスタンス	1	2

VMは(コア数-1)個を上限として配置

# 評価実験: 使用ベンチマーク・特徴

- モデル作成及び評価用ベンチマーク
  - SPEC CPU 2006のベンチマークアプリケーション28個
- リソース競合が発生する特徴
  - メモリバウンドのみ想定
- マイグレーションには時間がかかる
  - 同時にマイグレーションする数が多い → 再配置に時間がかかる
  - 同時にマイグレーションするVM数は1つとする

# 評価実験:アプリケーションの特徴分類モデルの作成



- モデル作成用アプリケーションを実行
- ・ パフォーマンスカウンタの取得

特徴量: パフォーマンスカウンタ

目的変数: アプリケーションの特徴(メモリバウンドかどうか)

分類器: SVM

- ・ 全カウンタ取得すると再配置に時間がかかる

→使用カウンタを選別

– カウンタ同士の相関係数を計算

- ・ カウンタAとBに強い相関関係があるならば  
取得するカウンタはどちらか1つが良い

# 評価実験:アプリケーションの特徴分類モデルの作成

- メモリに関係するカウンタを選別
  - 選別したカウンタ同士の相関係数を計算し  
相関係数の平均が低いものから4つを選択

SQ_FULL_STALL_CYCLES	Super Queue full stall cycles
L2_LINES_OUT	L2 modified lines evicted by a demand request
MEM_LOAD_RETIRED	Retired loads that miss the LLC cache
INST_RETIRED	実行命令数

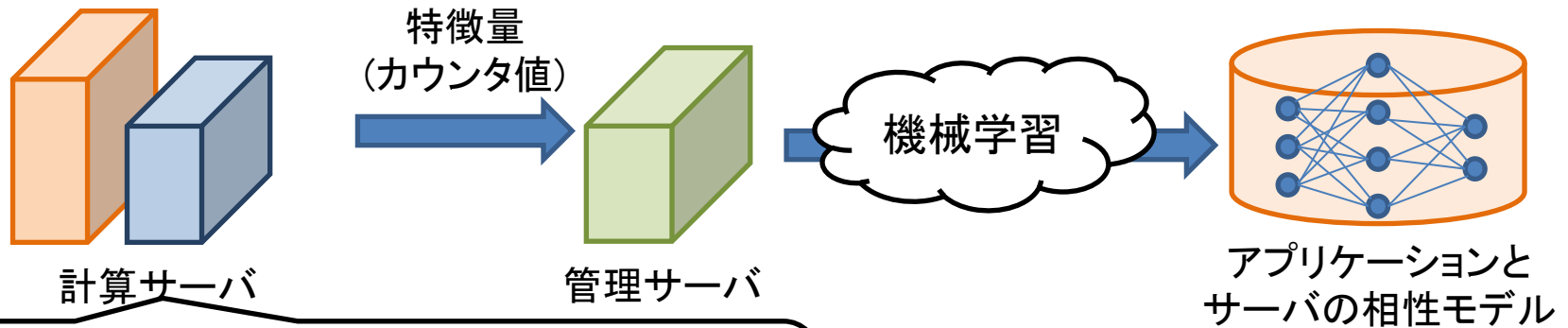
- モデルの評価
  - 28個のアプリケーションを4-fold cross validation

→分類精度91%

分類可能であると言える



# 評価実験:アプリケーションとサーバの相性モデルの作成



モデル作成用アプリケーションを実行

- パフォーマンスカウンタの取得
- 実行命令数、実行時間からMIPS値の計算

特徴量: パフォーマンスカウンタ

目的変数: 各サーバ構成で実行した際の予測MIPS値

教師ありニューラルネットワーク

## • 使用カウンタの選別

– 特徴分類モデルと同様に選別

全カウンタを取得し各カウンタ同士の相関係数を算出

相関係数の平均が0.4未満のものを選別

# 評価実験:アプリケーションとサーバの相性モデルの作成

PARTIAL_ADDRESS_ALIAS	False dependencies due to partial address aliasing
SSEX_UOPS_RETIRED	SIMD Packed-Double $\mu$ OPs 実行回数
L2_RQSTS	instruction fetch がL2キャッシュでヒットした回数
ILD_STALL	Instruction Length Decoderがストールしたサイクル数
UOPS_ISSUED	Fused $\mu$ OPs が発行された回数
MEM_LOAD_RETIRED	L1Dミスし、以前LFBに割り当てたものがhitしたload命令の回数
INST_RETIRED	実行命令数

## • モデルの評価

- 予測したMIPS値と実際に計測した正解のMIPS値の相関係数を評価指標とする
- 28個のアプリケーションを4-fold cross validation
  - 相関係数0.7
  - 一般的に強い相関があると言われている0.7以上

アプリケーションとサーバの相性は予測可能

# 評価実験：実験環境・評価方法

- 実験環境

- 物理サーバに予めVMが配置されている状況を想定
- VM上ではモデル作成に用いなかったアプリケーションを実行

- 評価方法

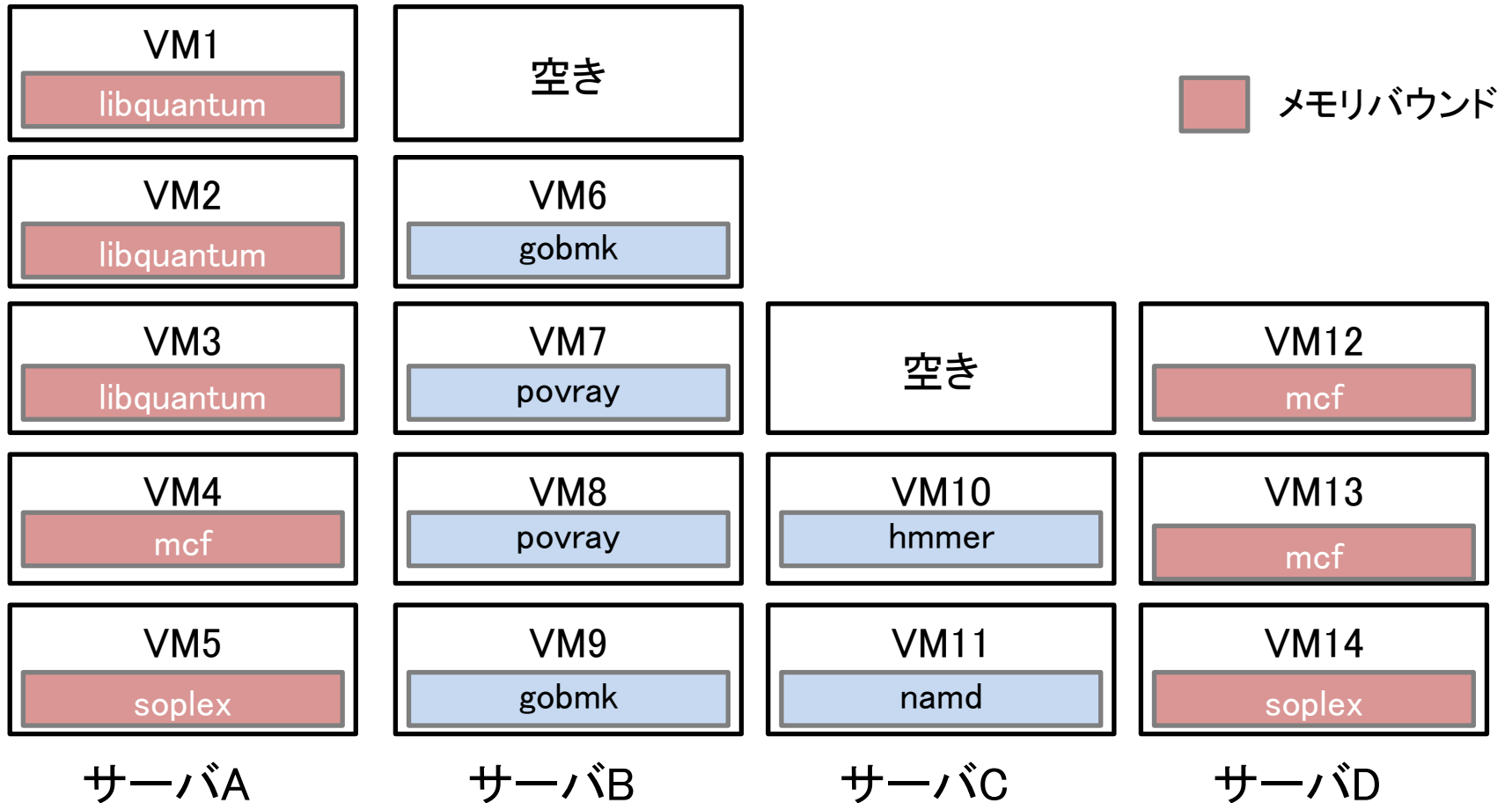
- 提案手法と再配置を行わない場合との各VM上でアプリケーションの実行が終了するまでの時間

- VMの初期配置

- リソース競合が発生するVMが偏っている場合
- リソース競合が発生するVMの偏りが変化する場合
- リソース競合が発生するVMが存在しない場合

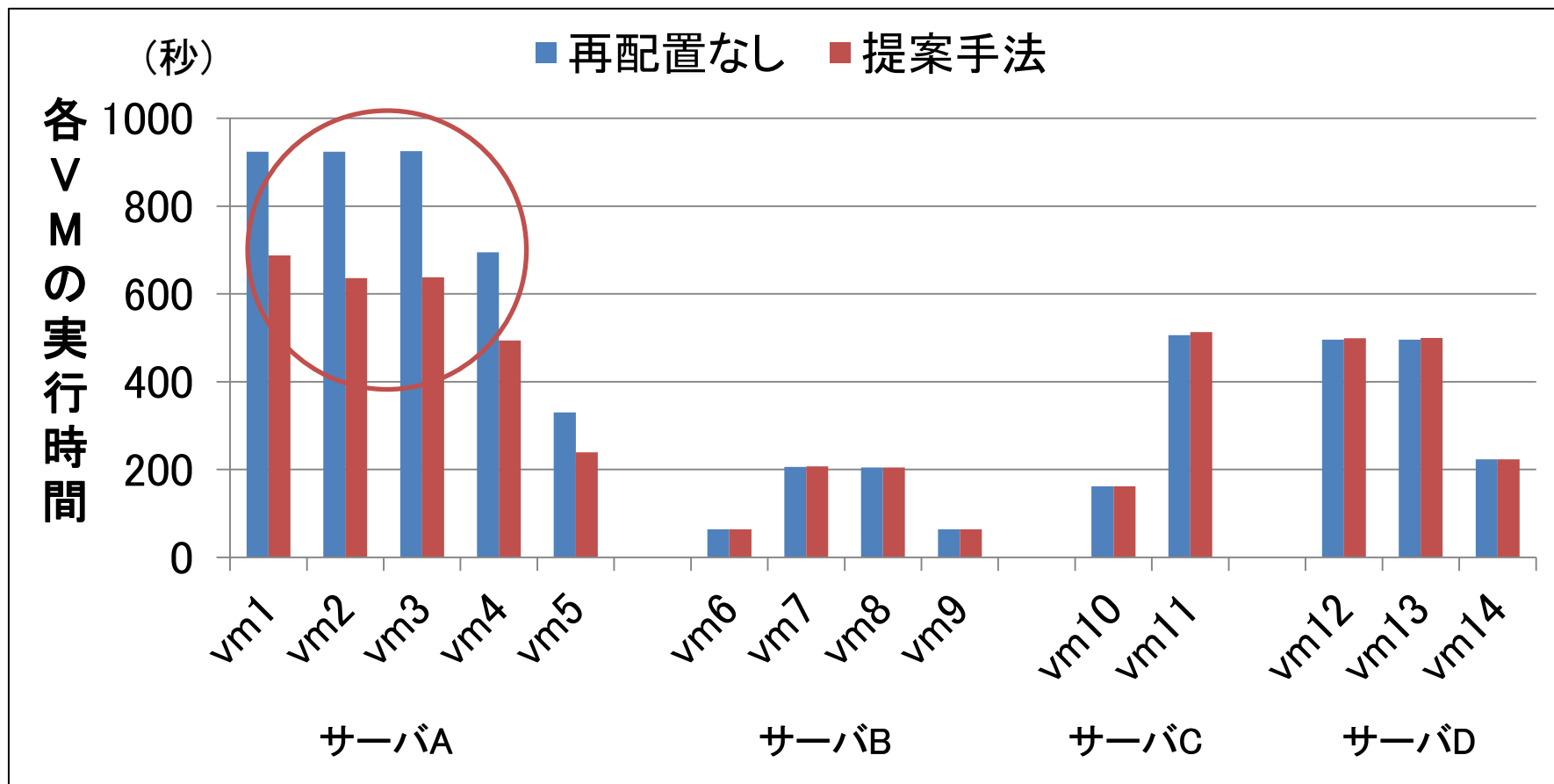
# 評価実験:

## リソース競合が発生するVMが偏っている場合



# 評価実験:

## リソース競合が発生するVMが偏っている場合の結果



アプリケーションの特徴分類は精度100%

相性の判断精度50% → 今後の課題

最大で26%実行時間を短縮

# まとめと今後の課題

- まとめ

- 同時に配置されるVM間のリソース競合とアプリケーションとサーバの相性を考慮した再配置手法を提案
- 評価実験の結果、再配置なしの場合と比べて各VMで実行するアプリケーションの実行時間の短縮ができた

- 今後の課題

- モデルの精度の改善
- より多くの共有リソースを考慮した評価
- 同時にマイグレーションを行うVMの数を増やす