基于容器的负载预测模型与能耗优化调度策略研究

答辩人: 王泽涛[1]　　导师: 林伟伟[2]

[1] 华南理工大学 计算机科学与工程学院

[2] 华南理工大学 计算机科学与工程学院

2019 年 5 月 26 日

目录

1. 研究动机
2. 趋势感知的容器负载预测模型
3. 基于动态伸缩的容器调度策略
4. 总结与展望

目录

1. 研究动机
   - 问题背景
   - 研究目标
     - 容器负载预测模型
     - 容器调度策略
2. 趋势感知的容器负载预测模型
3. 基于动态伸缩的容器调度策略
4. 总结与展望

问题背景

- 云计算运行模式：以容器为核心的 CaaS
  - 应用能够被简单部署并快速可用
  - CaaS 因虚拟化产生的性能损耗低：VMware[1] 和 IBM[2]
- 能耗优化与绿色计算[3]

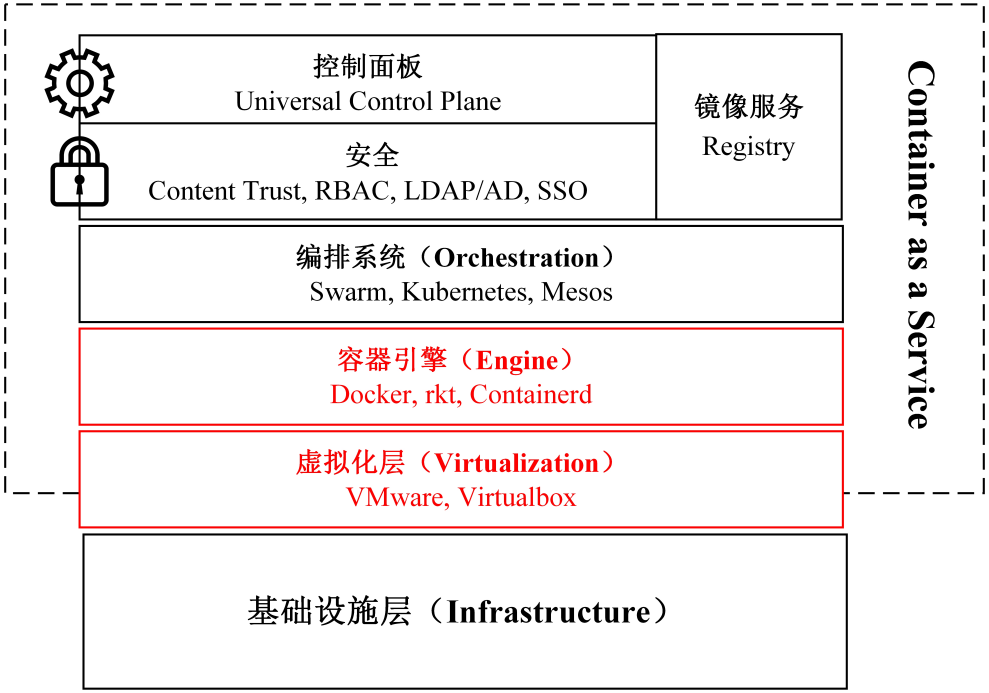

图 1: CaaS 架构示意图

研究目标

解决 CaaS 模式下云数据中心的能耗优化问题

- 负载预测模型
  - 实时性
  - 主动资源配置和容器调度
- 基于负载状态容器调度策略
  - 保证云服务 QoS（Quality of Service）
  - 降低整体能耗

研究目标

目标 1：容器负载预测模型 ⇒ **单值预测模型**

负载预测模型

- 单值预测模型：预测负载的特定值
  - **误差敏感 ⇒ 决策失效**
  - **鲁棒性差 ⇒ 决策过度**
- 区间预测模型*：预测负载的取值区间

类型	模型名称	简述
统计学模型	ARIMA	差分自回归移动平均
	Kalman 滤波	基于递推演进和观测值对预测结果进行演进
	排队论模型	基于在线作业请求和系统阻塞情况进行预测
	灰色模型	模型新陈代谢过程，进行中长期预测
机器学习模型	RNN	递归神经网络预测
	BPNN	原始信息前向传播，误差信息后向传播
	LSTM	长短期记忆网络
	混合模型	基于 c-means 聚类和模糊 (Fuzzy) 神经网络

表 1: 单值预测模型

研究目标

目标 1：容器负载预测模型 ⇒ **单值预测模型**

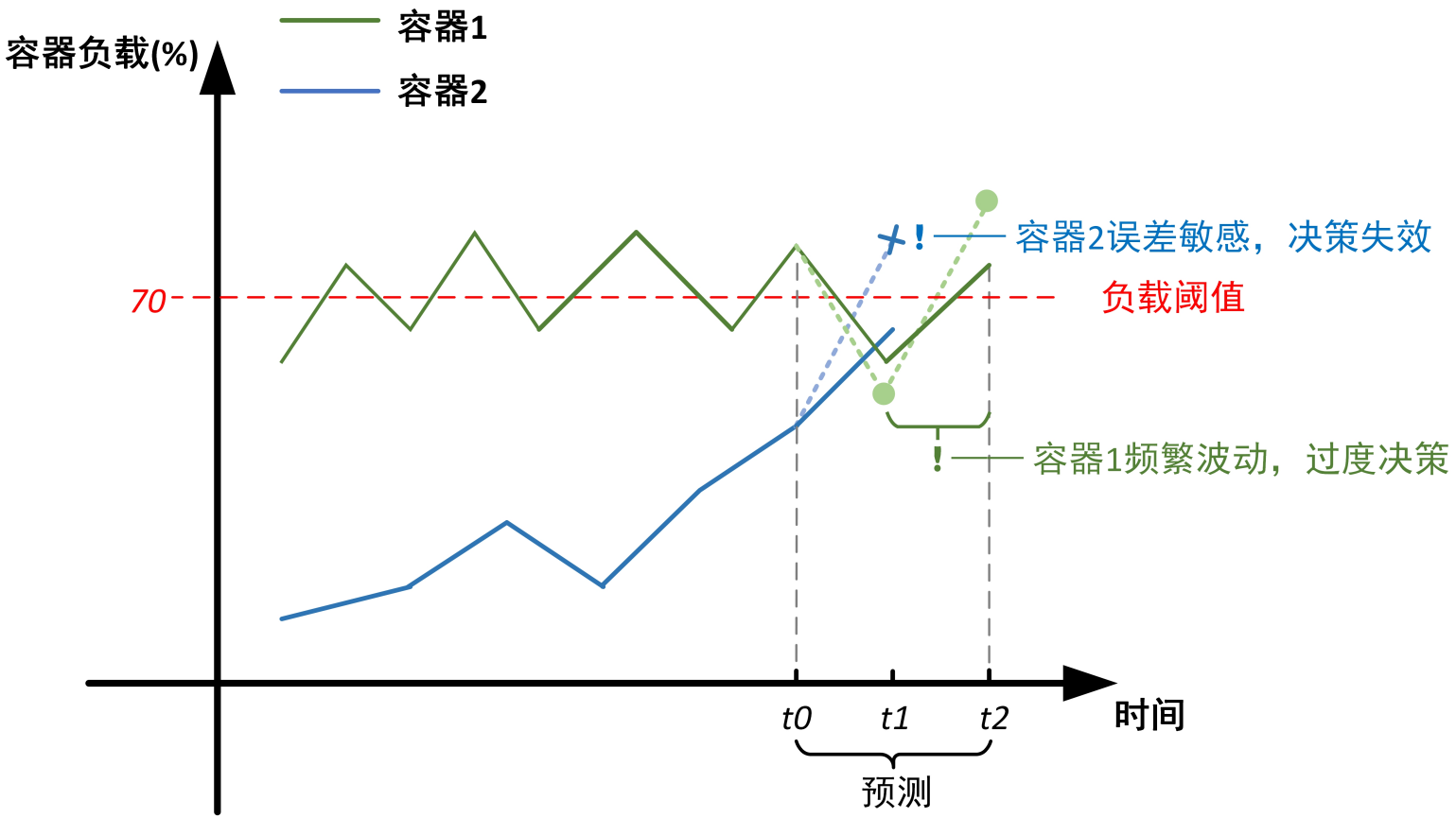

图 2: 单值预测模型

研究目标

目标 1：容器负载预测模型 $\Rightarrow$ **区间预测模型**

负载预测模型

- 单值预测模型：预测负载的特定值
  - 误差敏感 $\Rightarrow$ 决策失效
  - 鲁棒性差 $\Rightarrow$ 决策过度
- 区间预测模型*：预测负载的取值区间

类型	模型名称	简述
概率论模型	贝叶斯模型	根据样本分布假设和统计学特征扩展区间
机器学习模型	线性回归 SVM CART 随机森林	先拟合单值再根据分布假设构建区间

表 1: 区间预测模型

研究目标

目标 1：容器负载预测模型 ⇒ **区间预测模型**

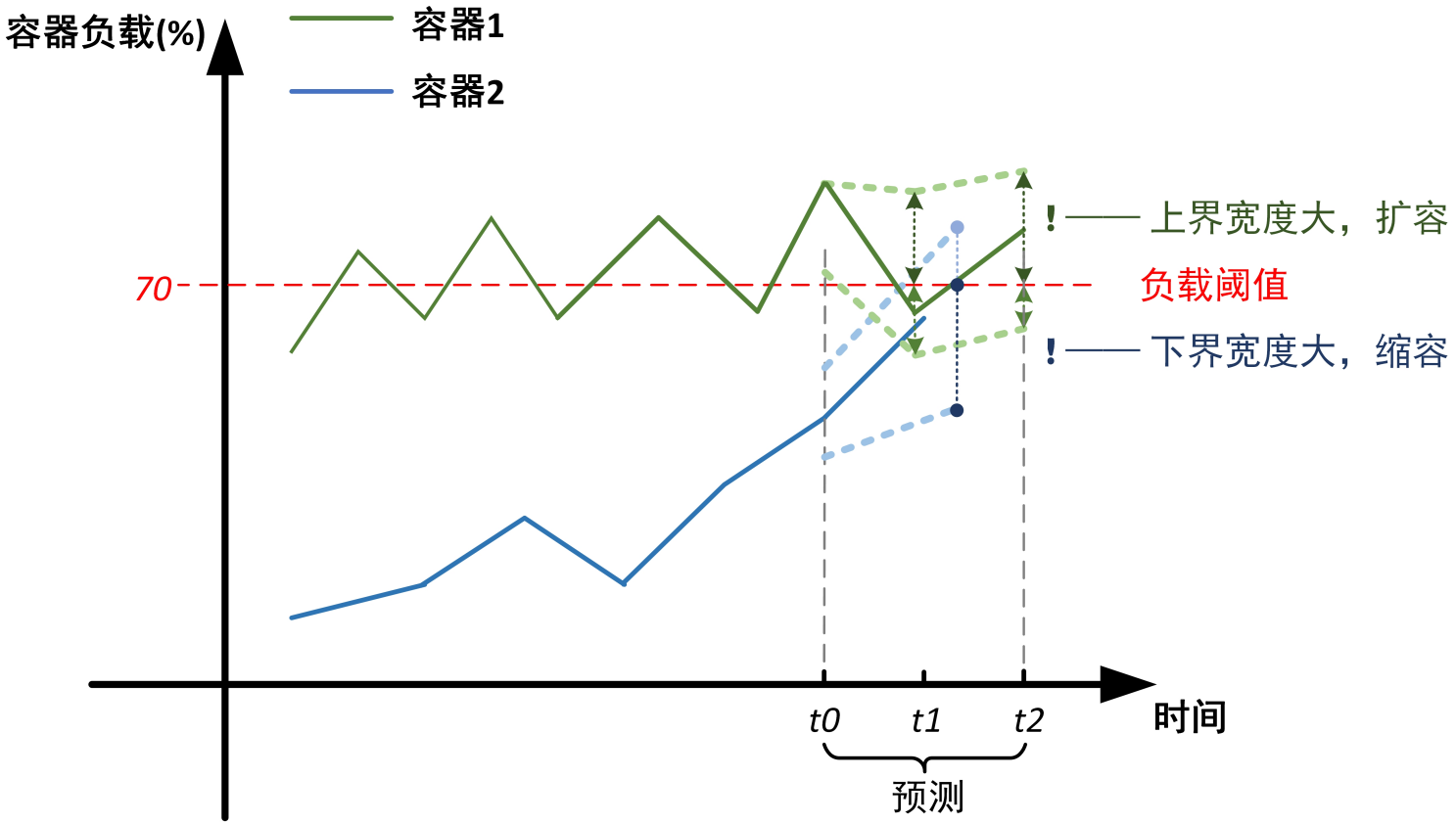

图 2: 区间预测模型

研究目标

目标 2：容器调度策略

容器调度策略：保证 QoS 同时降低能耗

- 动态调整容器数量：提高资源利用率
- 放置、迁移新增的和高负载主机上的容器：保证 QoS
- 合并低负载主机上的容器：减少虚拟机和已激活服务器数量，避免不必要的能耗

目录

1 研究动机

2 趋势感知的容器负载预测模型
- 模型结构
- 趋势感知
- 区间构造
- 区间预测
- 实验结果与分析

3 基于动态伸缩的容器调度策略

4 总结与展望

模型结构

引言

- 趋势感知的区间预测模型：不依赖分布假设，适用多种负载类型
  - **趋势感知** ⇒ 平稳型、趋势型和周期型负载
  - **区间构造** ⇒ 针对不同负载类型构造训练集上/下界
  - **区间预测** ⇒ GPSO 优化 SVM 模型超参数
  - **实时预测** ⇒ 模型建立后，根据系统反馈实时数据向前预测

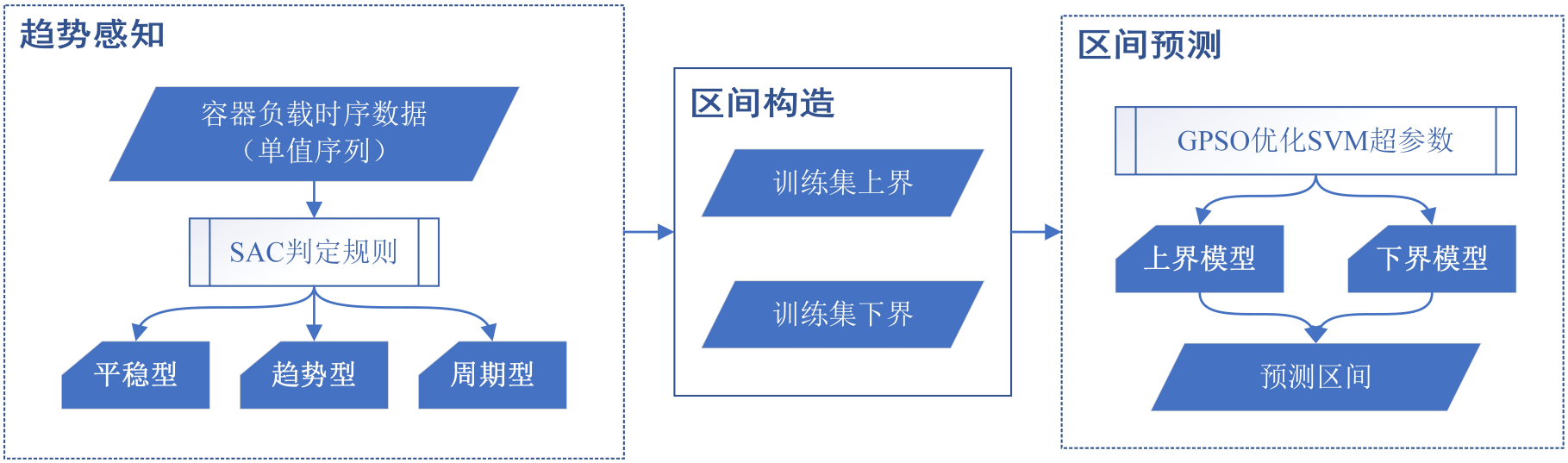

图 3: 基于趋势感知的区间预测模型

模型结构

示意图

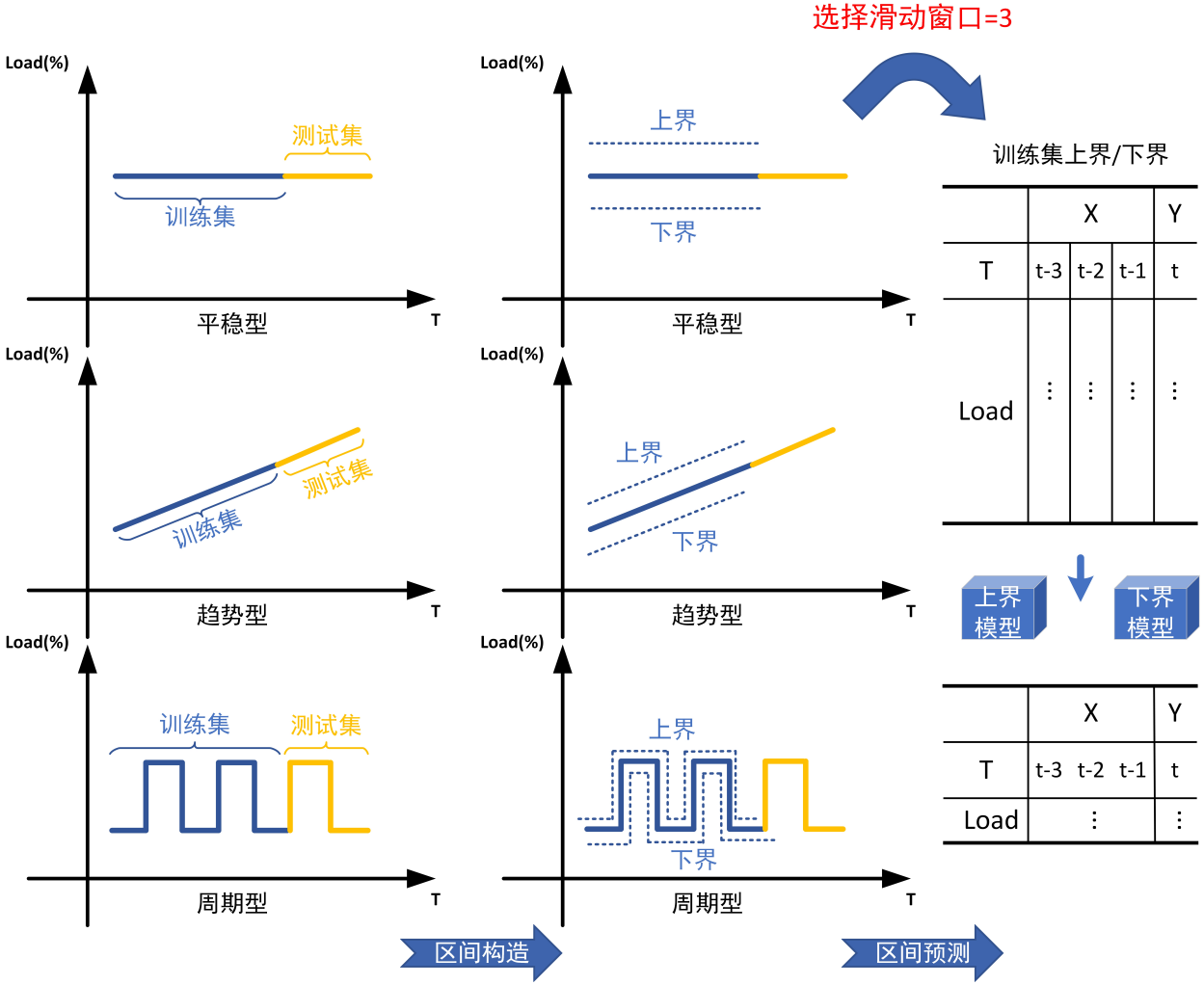

图 4: 模型预测过程

趋势感知

SAC 规则

- 频谱特征分析（**S**pectrum **A**nalysis）：判定周期性
- 自相关系数分析（**A**utocorrelation **C**oefficient）：判定平稳还是趋势

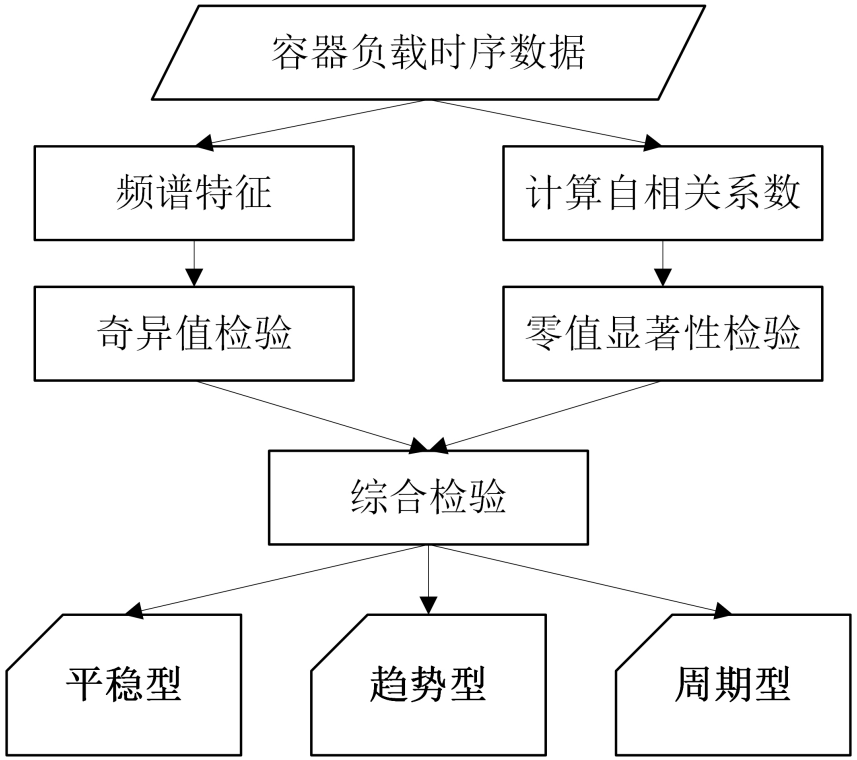

图 5: SAC 规则实施流程

趋势感知

SAC - 频谱特征分析

1. 负载数据（**时域数据**）$\Rightarrow$ 负载信号（**频域数据**）
2. 频谱特征：负载信号表现出的特征

负载信号：某一采样频率下，负载在频域上的取值

- 离散傅立叶变换（DFT）

$$\begin{cases} F_v = \sum_{j=1}^{n} e^{-i\frac{2\pi}{n}(v-1)(j-1)} load_i, v = 1, 2, \cdots, m \\ m = \lceil \frac{n}{2} \rceil \end{cases} \tag{1}$$

频谱特征：如图6所示

- 单个频谱特征

$$q_v = \frac{1}{m} |F_v|^2, v = 1, 2, \cdots, m \tag{2}$$

- 频谱特征序列

$$Q = \{q_1, \cdots, q_m\} = \{q_v\}|_{v=1}^{m} \tag{3}$$

趋势感知

SAC - 频谱特征分析

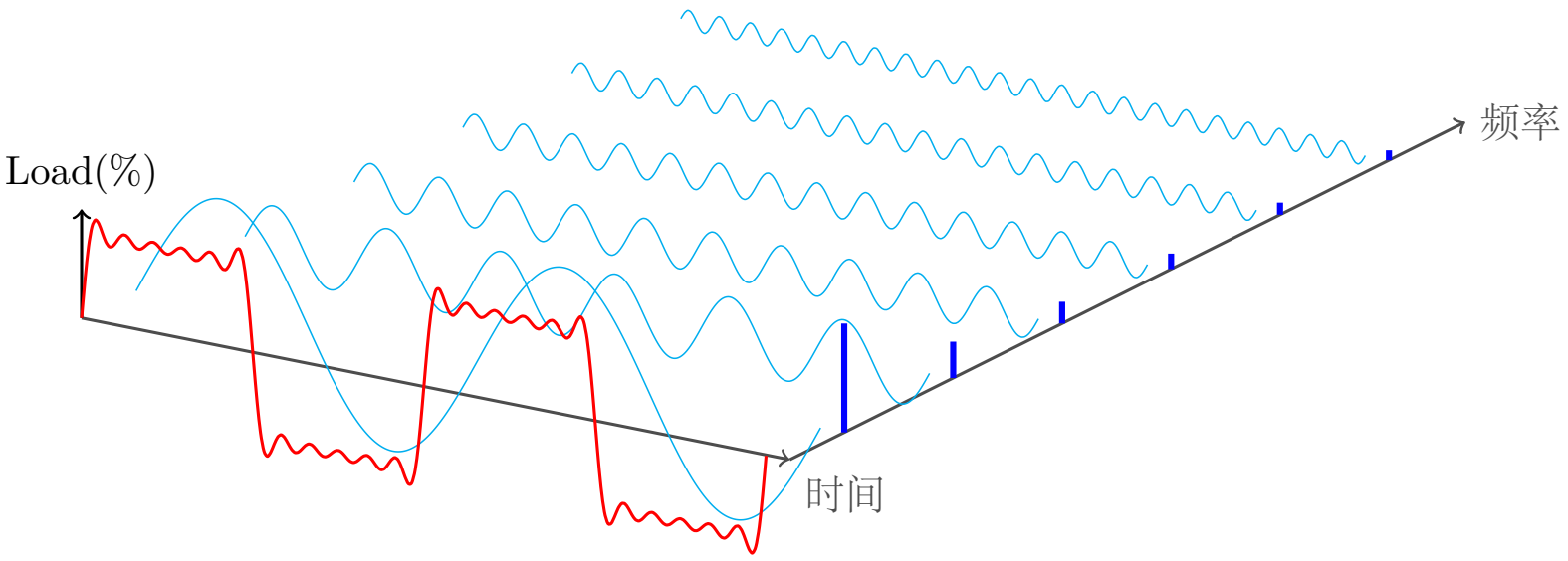

图 6: 基于傅立叶变换的频谱特征分析原理示意图

趋势感知

SAC - 自相关系数分析

1. 按滞后期 k，将原时序序列 S 拆分为两个新序列：S_{before} 和 S_{after}
   - 前置序列：$s_{before} = \{load_1, load_2, \cdots, load_{n-k}\} = \{load_i\}|_{i=1}^{n-k}$
   - 后置序列：$s_{after} = \{load_{1+k}, load_{2+k}, \cdots, load_n\} = \{load_i\}|_{i=1+k}^{n}$
2. 计算滞后期为 k 时的自相关系数：$\bar{S} = \sum_{i=1}^{n} \frac{load_i}{n}$

$$\begin{aligned} \rho_k &= \frac{\gamma(i, i+k)}{Var(S)} \\ &= \frac{\sum_{i=1}^{n-k}(load_i - \bar{S})(load_{i+k} - \bar{S})}{\sum_{i=1}^{n}(load_i - \bar{S})^2} \end{aligned} \tag{4}$$

3. 获得自相关系数序列：$k = 1, 2, \cdots, m,\ m = \lceil \frac{n}{3} \rceil$

$$P = \{\rho_1, \rho_2, \cdots, \rho_m\} = \{\rho_k\}|_{k=1}^{m} \tag{5}$$

趋势感知

SAC - 判定

- 周期性判定：奇异值检验
  1. 计算频谱特征序列 $Q=\{q_v\}|_{v=1}^{m}$ 的差分序列 $H=\{h_v\}|_{v=2}^{\lceil \frac{m}{2} \rceil}$

$$h_v = |(q_v - q_{v+1}) + (q_v - q_{v-1})|, v = 2, 3, \cdots, \lceil \frac{m}{2} \rceil \tag{6}$$

  2. 计算序列 H 的均值 μ_H 和标准差 σ_H
  3. 3σ 标准：序列 Q 中存在 q_v 使对应的差分项 h_v 满足如下条件

$$h_v - \mu_H > 3\sigma_H \tag{7}$$

- 平稳性判定：显著性检验
  1. t-检验 $\Rightarrow$ H0：序列平稳，H1：序列不平稳，显著性水平：$\alpha = 0.05$
  2. 计算自相关系数序列 P 总体与零值相关概率 P_ρ
  3. 检验标准
     - $P_\rho \geq \alpha \Rightarrow$ 接受 H0 假设
     - $P_\rho < \alpha \Rightarrow$ 接受 H1 假设

趋势感知

SAC - 判定

负载类型定义

- 平稳型负载：不存在 q_v，使 h_v 满足公式7，且 $P_\rho \geq \alpha$
- 趋势型负载：不存在 q_v，使 h_v 满足公式7，且 $P_\rho < \alpha$
- 周期型负载：存在 q_v，使 h_v 满足公式7

区间构造

平稳型负载：基于序列最大最小值

1: $S_u.\text{add}(0.5 * (S_i + S_{max}))$
2: $S_l.\text{add}(0.5 * (S_i + S_{min}))$

趋势型负载：差分项绝对值的均值 ⇒ 震动幅度

1: $interval.\text{add}(\text{abs}(S_{i+1} - S_i))$
2: $interval_avg = \textbf{average}(interval)$
3: $(S_i, S_{max}, S_{min}, interval_avg) \Rightarrow (S_u, S_l)$

周期型负载：分三段计算震动幅度

1: $i \in [1, \frac{Period}{2})$
2: $i \in [\frac{Period}{2}, S_{len} - \frac{Period}{2})$
3: $i \in [S_{len} - \frac{Period}{2}, S_{len}]$
4: $(S_i, S_{max}, S_{min}, interval_avg^*) \Rightarrow (S_u, S_l)$

区间预测

超参数优化

- GPSO 来优化 SVM 超参数：C^u_{gbest}, γ^u_{gbest}, C^l_{gbest}, γ^l_{gbest}
- GPSO 的适应度函数：区间覆盖度标准（CWC）[4]
  - 区间覆盖率 (PICP)：真值落在预测区间内的百分比
  - 区间归一化宽度 (PINAW)

$$CWC = PINAW * (1 + \gamma e^{\eta(PICP-\mu)})$$
$$\text{where}\gamma = \begin{cases} 0 & PICP \geq \mu \\ 1 & PICP < \mu \end{cases} \tag{8}$$

区间预测

训练与预测

- 基于最优参数 SVM 模型：C^u_{gbest}, γ^u_{gbest}, C^l_{gbest}, γ^l_{gbest}

Require: C^u_{gbest}, γ^u_{gbest}, C^l_{gbest}, γ^l_{gbest}, S^u_{train}, S^l_{train}, S^u_{test}, S^l_{test}

Ensure: P^u, P^l

1: ▷ 基于最优超参数的**上界模型**

2: $svm_u = \text{SVM}(kernel{=}\textbf{RBF},\ C{=}C^u_{gbest},\ \gamma{=}\gamma^u_{gbest})$

3: $model_u = svm_u.\text{fit}(S^u_{train})$ ▷ 训练模型

4: $P^u = model_u.\text{predict}(S^u_{test}.\text{Time})$ ▷ 预测上界

5: ▷ 基于最优超参数的**下界模型**

6: $svm_l = \text{SVM}(kernel{=}\textbf{RBF},\ C{=}C^l_{gbest},\ \gamma{=}\gamma^l_{gbest})$

7: $model_l = svm_l.\text{fit}(S^l_{train})$ ▷ 训练模型

8: $P^l = model_l.\text{predict}(S^l_{test}.\text{Time})$ ▷ 预测下界

9: **return** P^u, P^l

实验结果与分析 (1/4)

预测效果分析

- SAC-GPSO-SVM 模型预测效果：数据来源 (PlanetLab)[1]

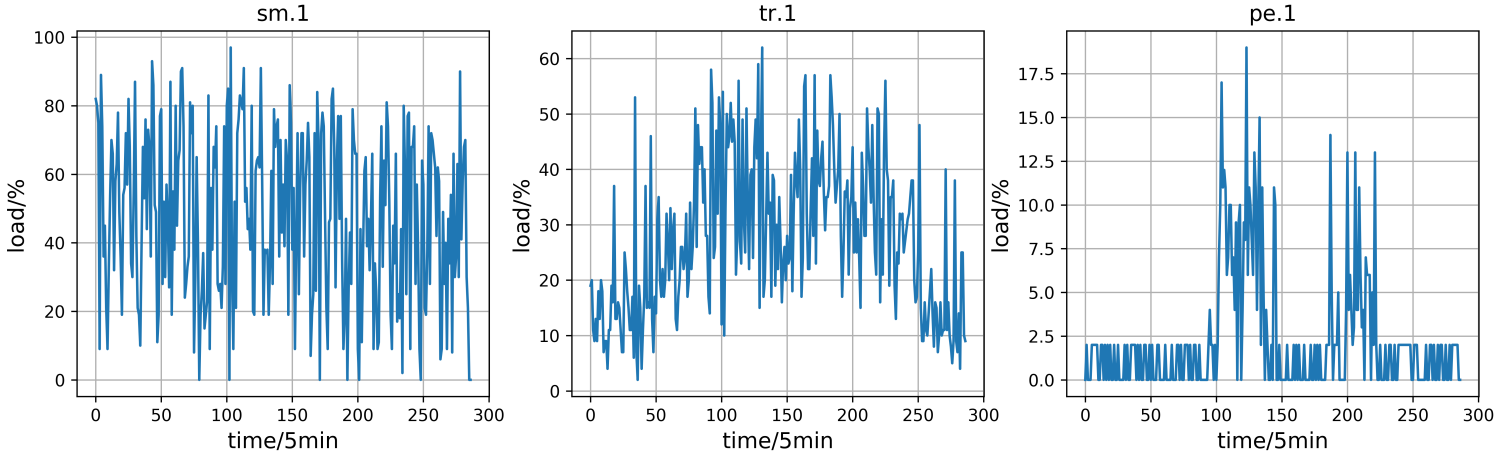

图 7: 平稳型、趋势型和周期型容器 24h/5min 的负载数据

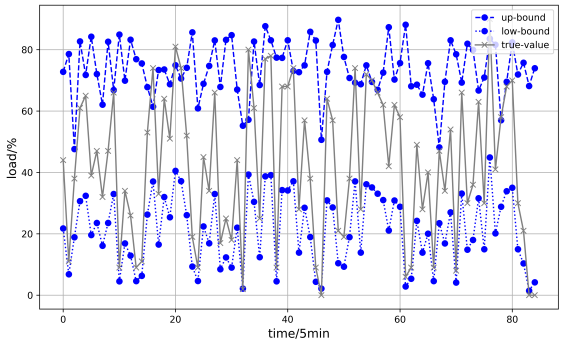

图 8: sm.1 预测结果

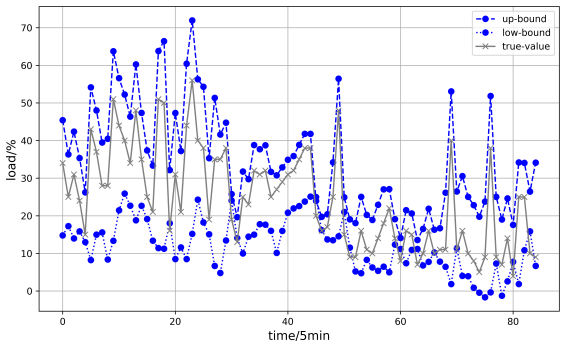

图 9: tr.1 预测结果

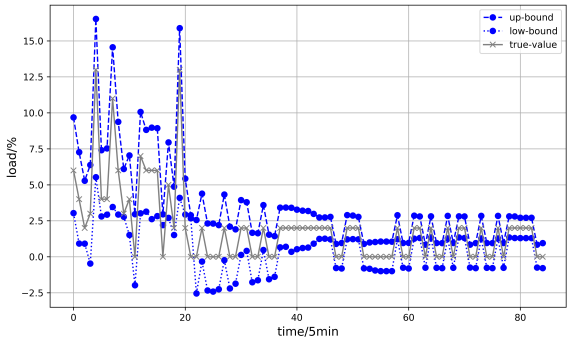

图 10: pe.1 预测结果

[1] 2011.03.03-2011.04.20 内 11746 个容器当日每 5 分钟内 CPU 利用率均值数据

实验结果与分析 (1/4)

预测效果分析

对比算法

- Native Bayes，Linear Regression，SVM，CART 和 RFR

- Native Bayes

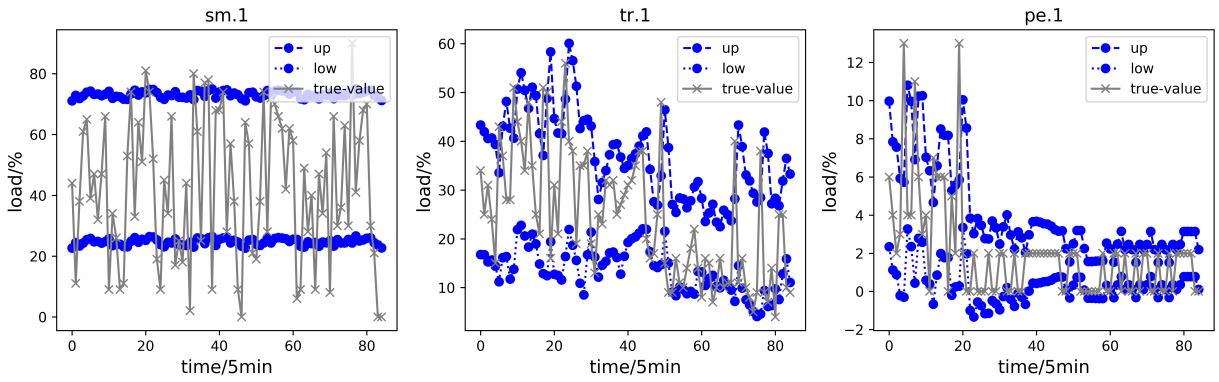

图 11: Native Bayes 预测结果

实验结果与分析 (1/4)

预测效果分析

- Linear Regression 和 SVM

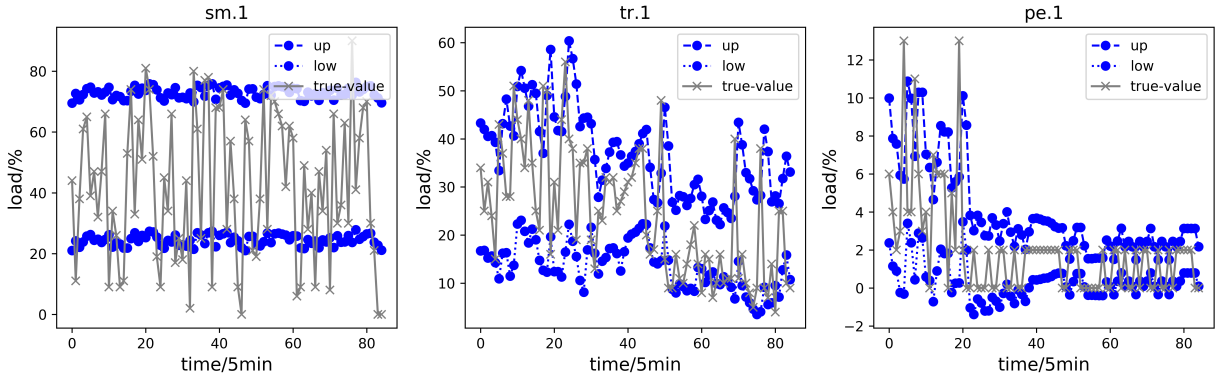

图 12: Linear Regression 预测结果

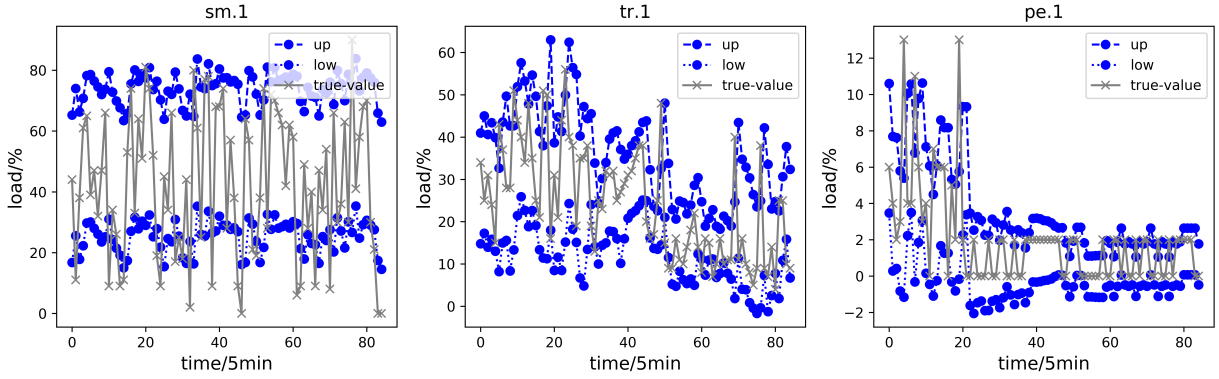

图 13: SVM 预测结果

实验结果与分析 (1/4)

预测效果分析

- CART 和 RFR

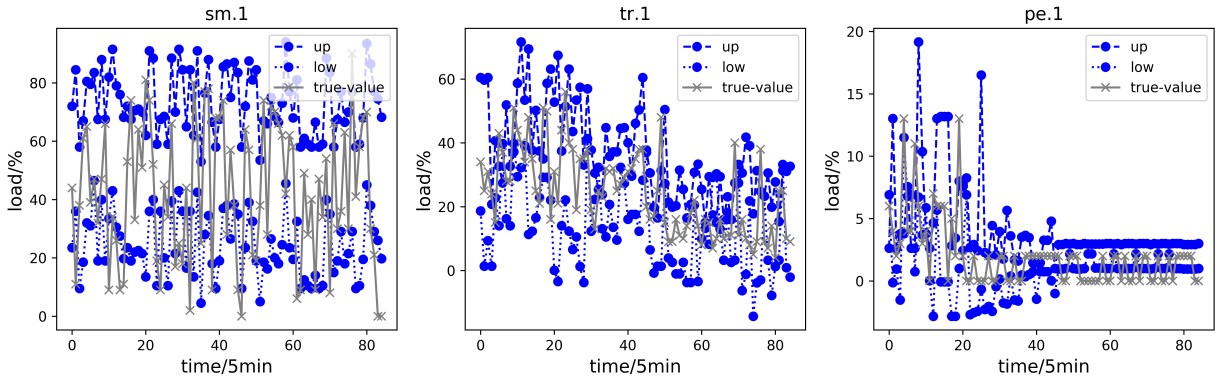

图 14: CART 预测结果

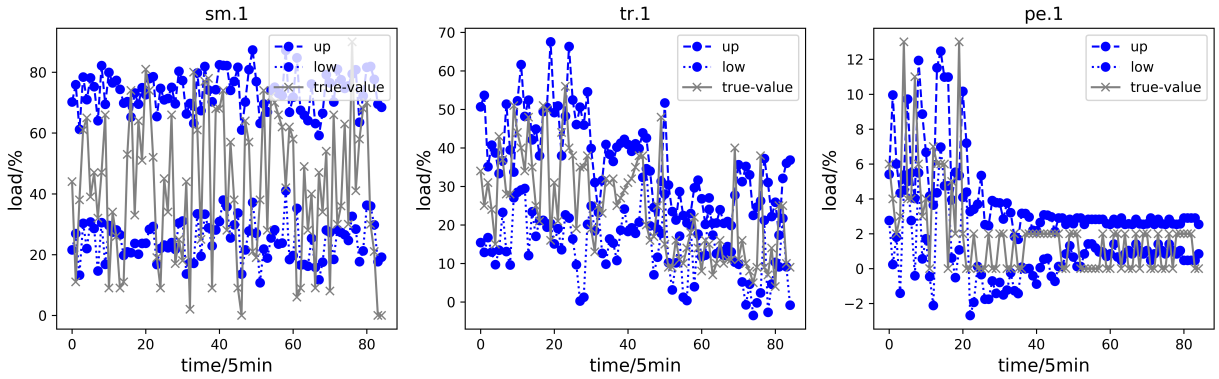

图 15: RFR 预测结果

实验结果与分析 (2/4)

预测准确性对比

- 红色：最优
- 黄色：优
- 蓝色：次优

模型	CWC(%) ⇓	PICP(%) ⇑	PINEW(%) ⇓
Native Bayes	71.95	89.41	37.41
Linear Regression	114	83.53	40.88
SVM	96.92	88.24	44.19
CART	94.16	84.31	34.39
RFR	68.56	89.42	43.55
SAC-GPSO-SVM	61.26	89.40	35.15

表 1: 预测准确性对比

实验结果与分析 (3/4)

预测效率分析

- 剔除超参数优化时间，只考虑模型平均训练和预测时间

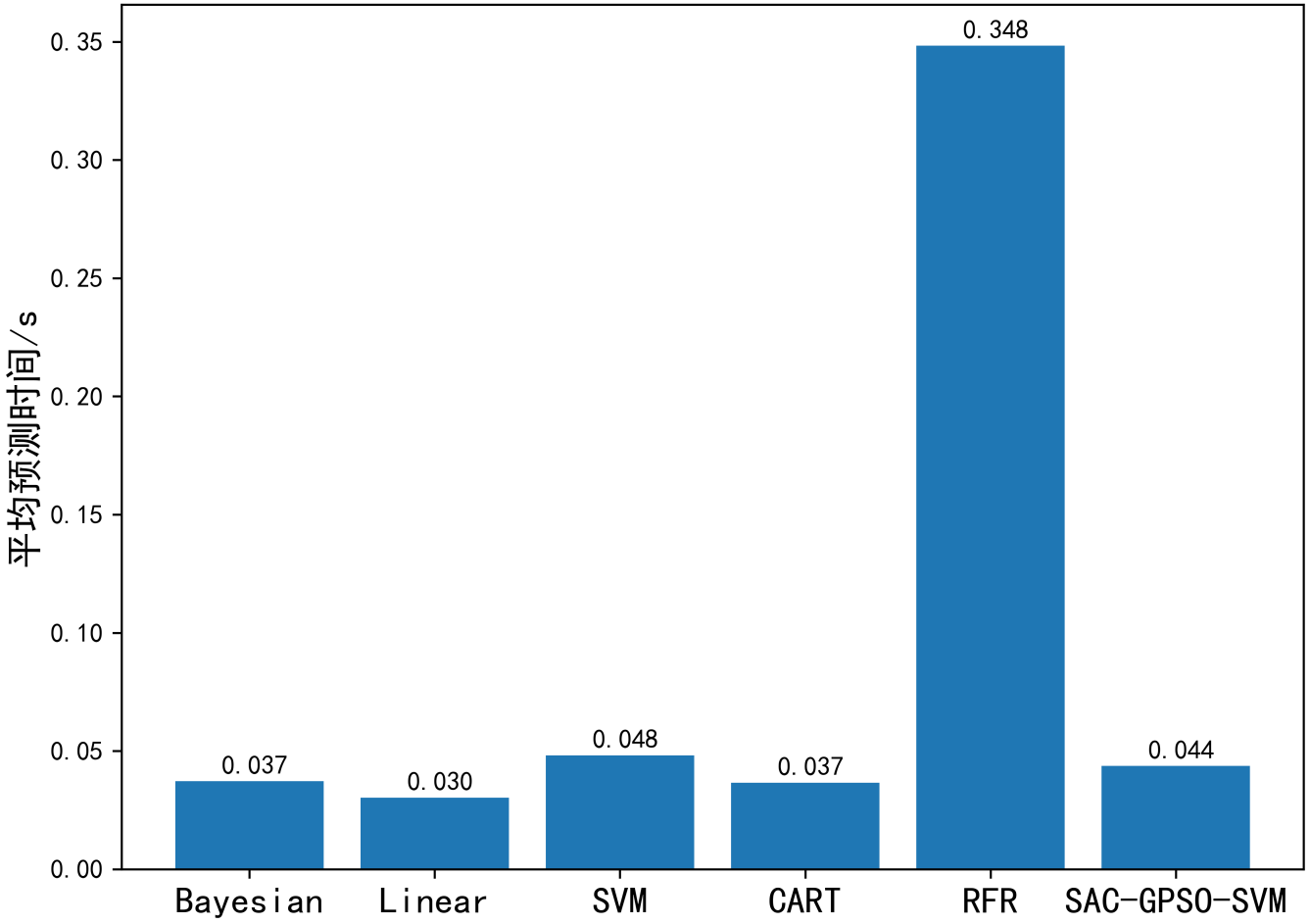

图 16: 各模型平均预测时间对比

实验结果与分析 (4/4)

参数优化效率与模型实时性检验

- 红色：最优
- 黄色：优
- 蓝色：次优

优化方法	平均最优 CWC(%)	平均迭代次数	平均预测时间 (s)
GridSearchCV	68.24	1358	465.21
RandomizedSearchCV	68.56	103	122.47
PSO	68.64	87	78.53
GPSO	68.45	30	23.34

表 2: 预测准确性对比

目录

1 研究动机

2 趋势感知的容器负载预测模型

3 基于动态伸缩的容器调度策略
- 问题定义
- 系统结构
- 调度算法
- 实验结果与分析

4 总结与展望

问题定义
调度目标

- 最小化 CaaS 数据中心总能耗

$$
\begin{aligned}
\min \quad & P_{dc}(t) = \sum_{i=1}^{N_s} P_i(t) \\
s.t. \quad & \begin{cases} \sum_{j=1}^{N_{vm}} vm_{(j,i,r)}(t) < S_(i,r), \\ \forall i \in [1, N_s],\ \forall r \in \{CPU, Memory, Disk, BW\} \\ \\ \sum_{k=1}^{N_c} c_{(k,j,i,r)}(t) < vm_(j,i,r), \\ \forall i \in [1, N_{vm}],\ \forall r \in \{CPU, Memory, Disk, BW\} \end{cases}
\end{aligned}
\tag{9}
$$

调度系统模型设计思路

- 主机状态模块：标记高负载/低负载主机
  - 高负载主机列表
  - 低负载主机列表
- 容器状态模块
  - 负载预测结果与调整阈值 ⇒ 计算容器扩容、缩容数量
  - 从高负载主机列表中选择需要迁移容器
- 容器调度模块
  - 为待迁移容器找到目的地
  - 基于资源约束条件合并低负载主机上的容器
  - 销毁并关闭空载的虚拟机和服务器 ⇒ 避免不必要的能耗

系统结构

调度系统示意图

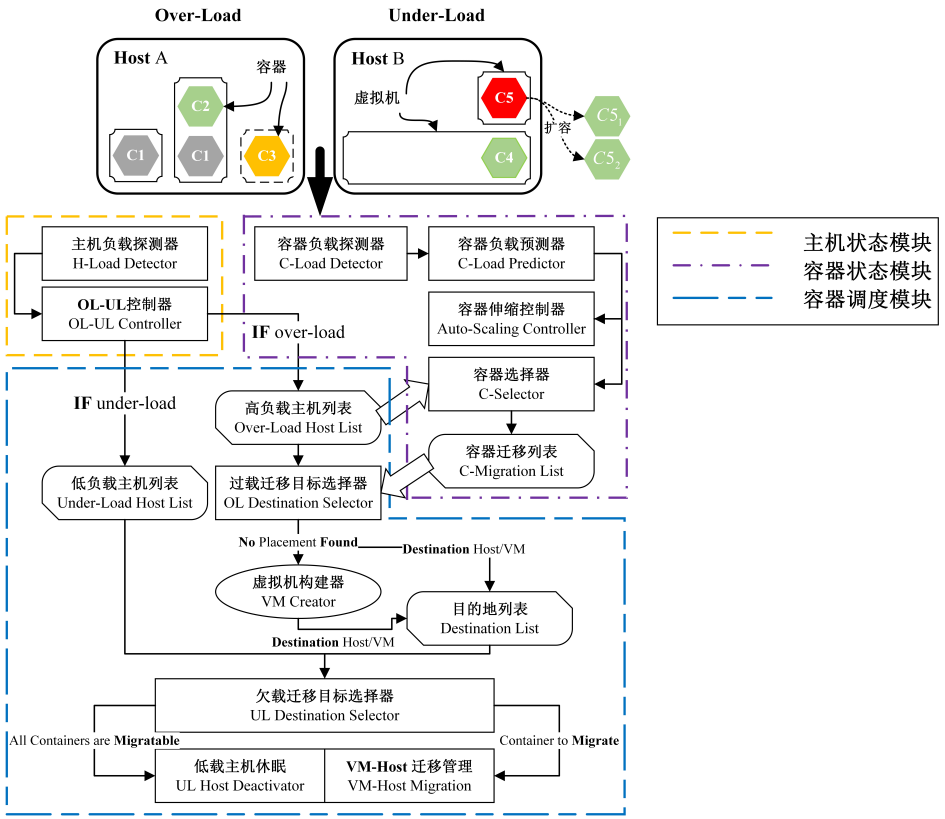

图 17: 基于动态伸缩的容器调度系统结构

调度算法

主机状态模块 (1/1)：主机状态标记算法 - 标记主机状态

- 主机状态标记：通过**静态阈值法**设置 T_{ol} 和 T_{ul}

$$Host\ status = \begin{cases} \textbf{Over-Load} & if\ U_i(t) > T_{ol} \\ \textbf{Under-Load} & if\ U_i(t) < T_{ul} \end{cases} \tag{10}$$

调度算法

容器状态模块 (1/2)：容器伸缩算法 - 计算容器扩缩容数量

- 令应用 V 的当前容器副本总数 $n = n_{now}$
  - 应用 V 负载上界：$L_v^u = \sum_{i=1}^{n_{now}} l_i^u$
  - 应用 V 负载下界：$L_v^l = \sum_{i=1}^{n_{now}} l_i^l$
- 设置用于应用 V 扩缩容的调整阈值为 L_v
  - 满足公式11(负载落在 L_v 之上) $\Rightarrow$ 扩容 $\lceil \frac{(L_v^u + L_v^l)}{2L_v} - n_{now} \rceil$ 个
  - 满足公式12(负载落在 L_v 之下) $\Rightarrow$ 缩容 $\lceil n_{now} - \frac{(L_v^u + L_v^l)}{2L_v} - 1 \rceil$ 个

$$\frac{L_v^u - n_{now} \times L_v}{n_{now} \times L_v - L_v^l} > 1 \tag{11}$$

$$\frac{(n_{now} - 1) \times L_v - L_v^l}{L_v^u - (n_{now} - 1) \times L_v} > 1 \tag{12}$$

调度算法

容器状态模块 (2/2)：容器选择算法 - 选择高负载主机上需要迁移容器

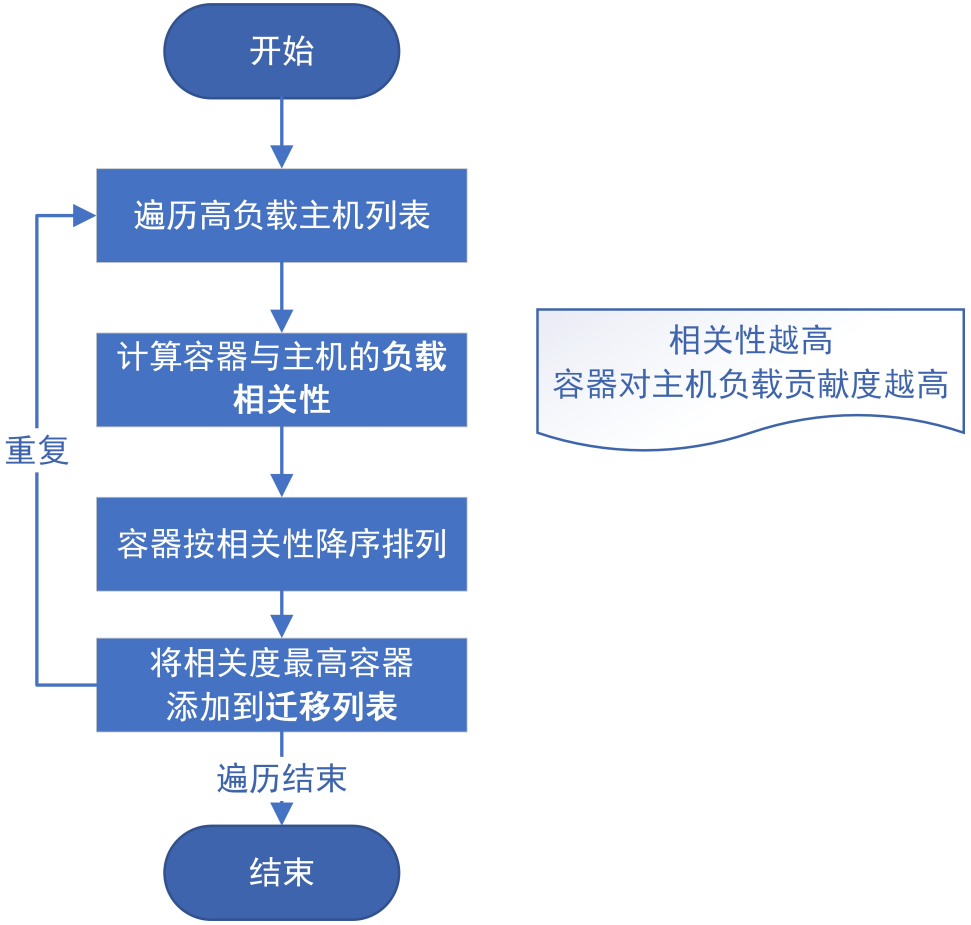

图 18: 容器选择算法流程

调度算法

容器调度模块 (1/3)：主机选择算法 - 选择容器迁移目的主机

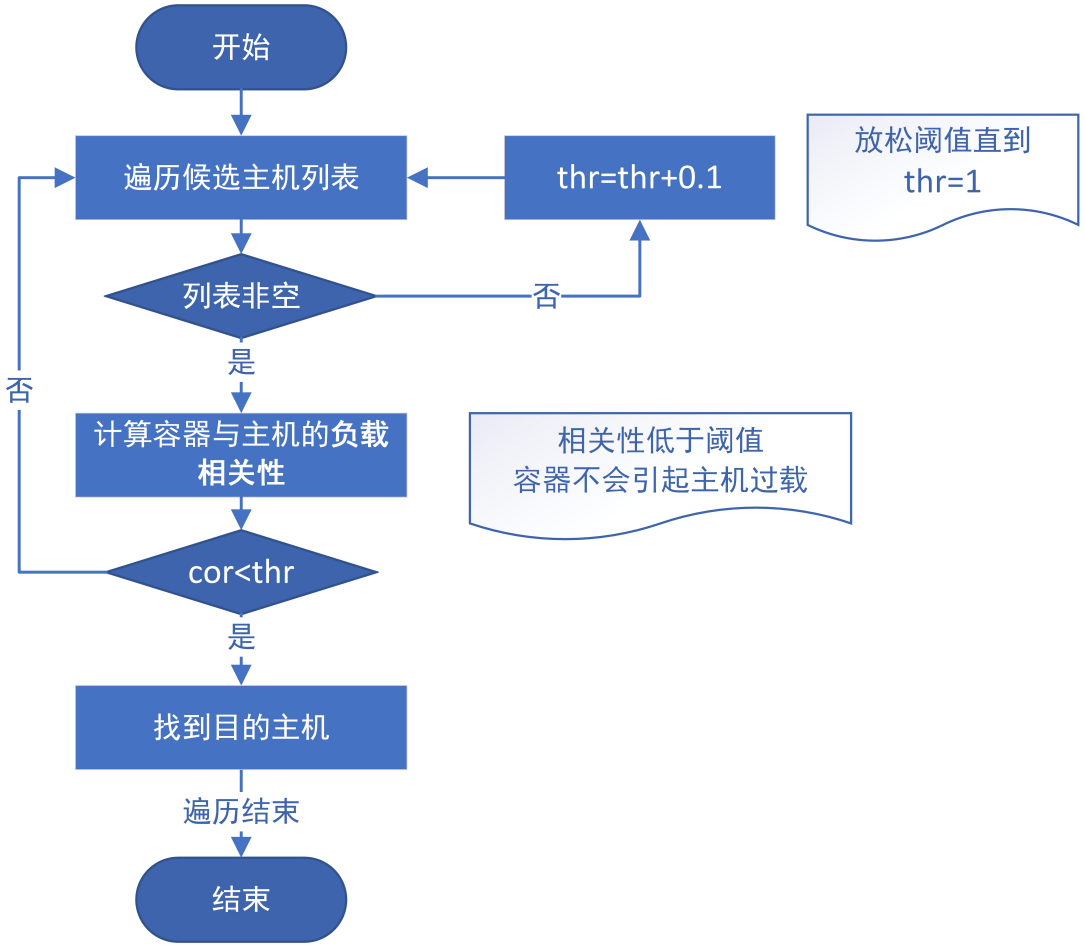

图 19: 主机选择算法流程

调度算法

容器调度模块 (2/3)：容器迁移和放置算法 - 迁移高负载主机上容器、放置新增容器

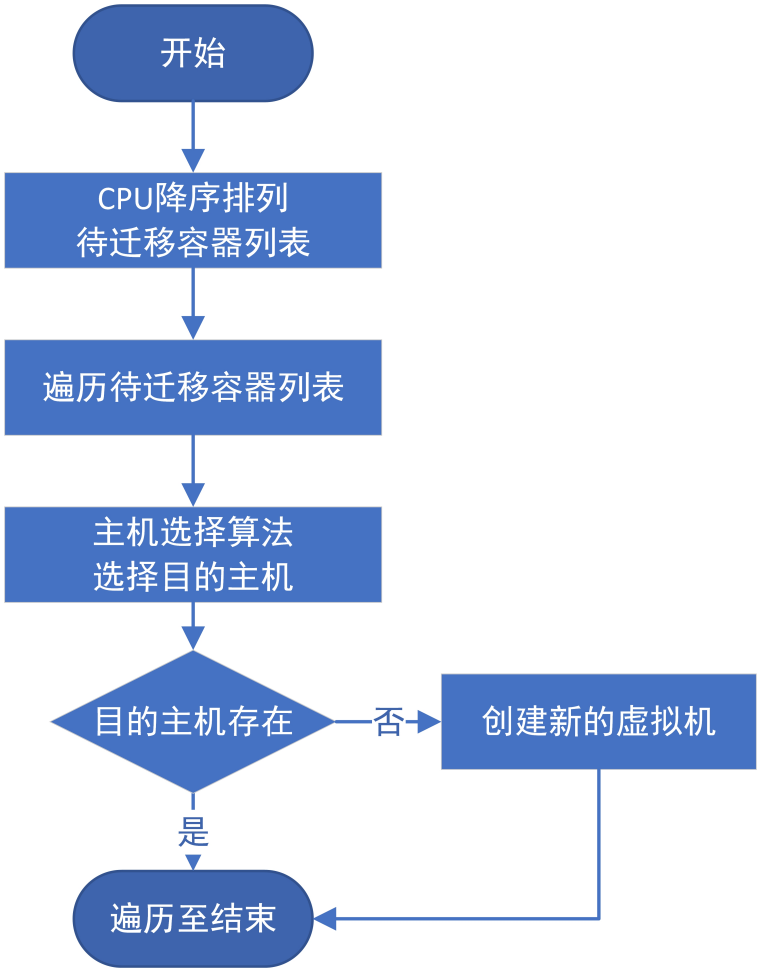

图 20: 容器迁移和放置算法流程

调度算法

容器调度模块 (3/3)：容器合并算法 - 合并低负载主机容器

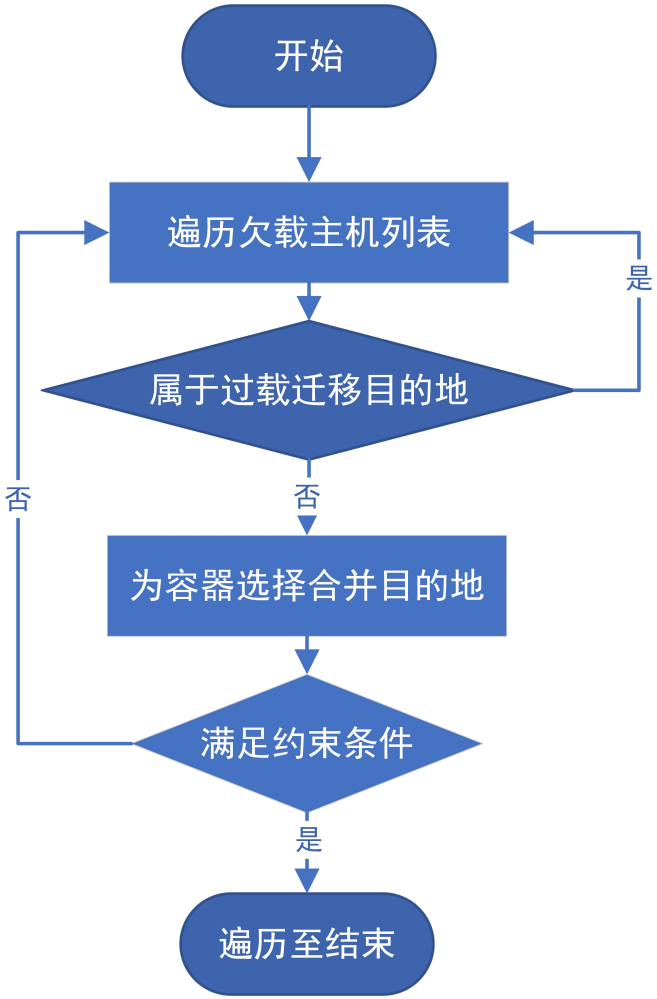

图 21: 容器合并算法流程

实验结果与分析

QoS 指标

- SLA 违反率：容器未分配的 CPU 资源量在总请求量中的占比

$$SLA = \sum_{i=1}^{N_s} \sum_{j=1}^{N_{vm}} \sum_{p=1}^{N_v} \frac{CPU_r(vm_{ji}, t_p) - CPU_a(vm_{ji}, t_p)}{CPU_r(vm_{ji}, t_p)}$$
$$\begin{cases} CPU_r(vm_{ji}, t_p) = \sum_{k=1}^{N_c^{ji}} CPU_r(c_{kji}, t_p) \\ CPU_a(vm_{ji}, t_p) = \sum_{k=1}^{N_c^{ji}} CPU_a(c_{kji}, t_p) \end{cases} \tag{13}$$

- 容器对 CPU 资源的请求量：依据**超量因子** P_n 从历史记录中获取
  1. 从容器负载历史记录中选择一个负载值 L_{per}
  2. 概率 $P(L \leq L_{per}) \geq P_n$
  3. $min(L_{per})$ 即为容器对 CPU 资源的请求量

实验结果与分析

实验设置

- 容器负载数据 (PlanetLab)、服务器和虚拟机配置、CloudSim
- 对比算法
  - **FFHS(Kubernetes)**：选择第一台满足资源请求的主机
  - **LFHS**：FFHS 基础上增加主机 CPU 降序排列预处理
  - **WEEC**：聚类划分容器，基于能耗特征的启发式算法
  - **本文基于动态伸缩的容器调度策略**
- 对照实验：
  - 控制容器伸缩的" **应用总体负载阈值**"$= 70\%$
  - CorHS 中的" **负载相关度阈值**"$= 0.5$

实验组	约束目标	UL	OL	Overbooking
group.1	OL 阈值	70%	[80%, 90%, 100%]	$80th$
group.2	UL 阈值	[50%, 60%, 70%]	80%	$80th$
group.3	Overbooking 阈值	70%	80%	$[20th, 40th, 80th]$

表 3: 实验对照组设置

实验结果与分析

实验结果：group.1 **UL**=70% **OL**=[80%, 90%, 100%], **Overbooking**=80*th*

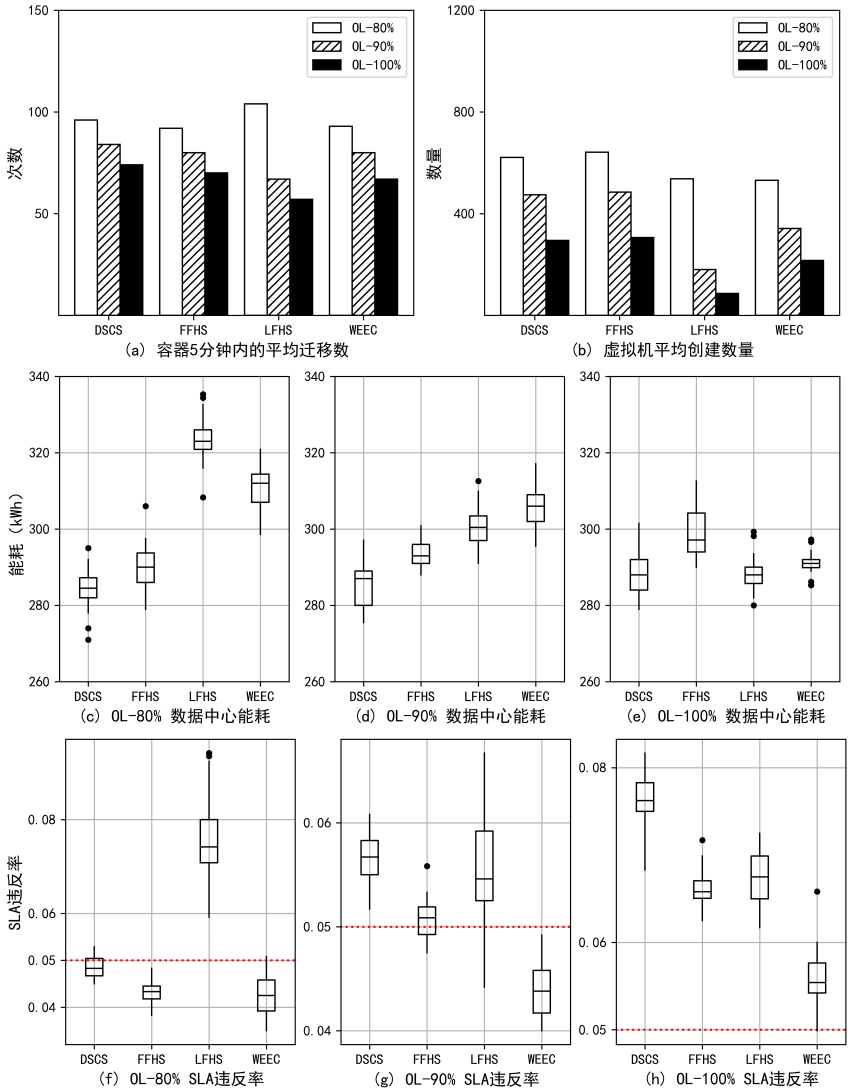

图 22: group.1 4 种调度策略容器迁移、虚拟机创建数、能耗和 SLA 对比

实验结果与分析

实验结果：group.2 **UL**=[50%, 60%, 70%] **OL**=80%, **Overbooking**=80*th*

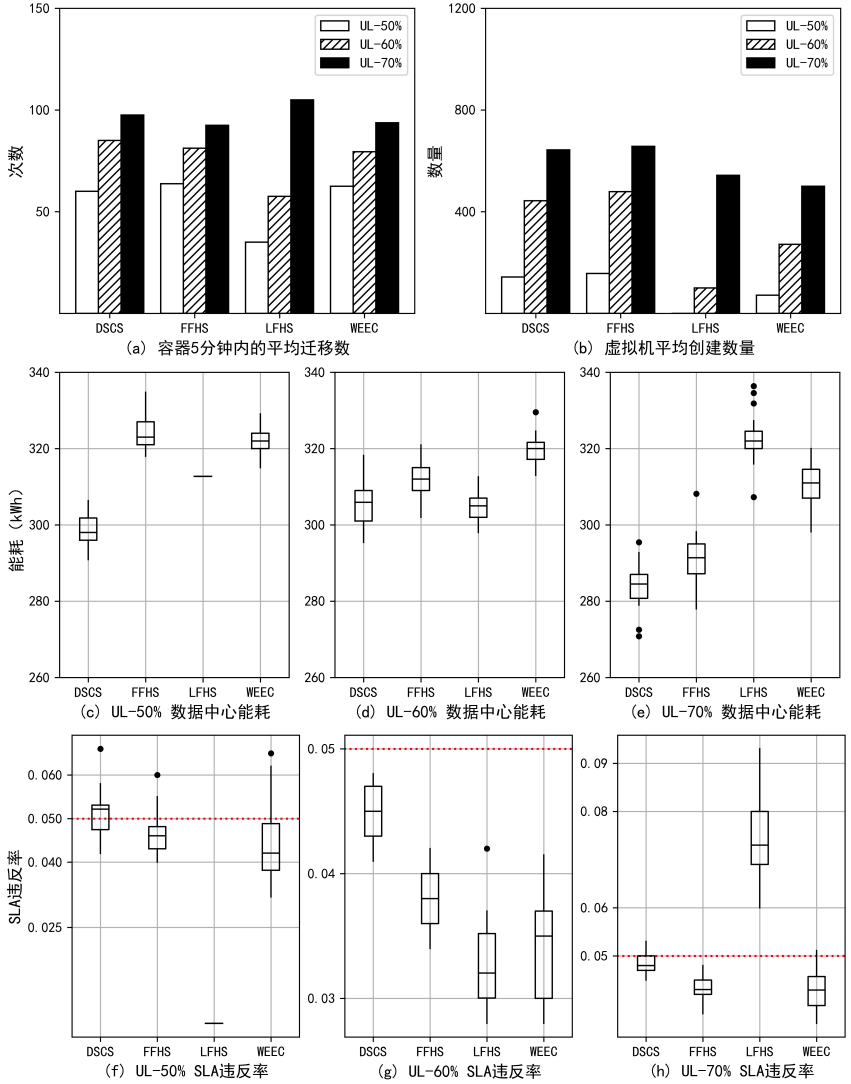

图 23: group.2 4 种调度策略容器迁移、虚拟机创建数、能耗和 SLA 对比

实验结果与分析

实验结果：group.3 **UL**=70% **OL**=80%, **Overbooking**=[20th, 40th, 80th]

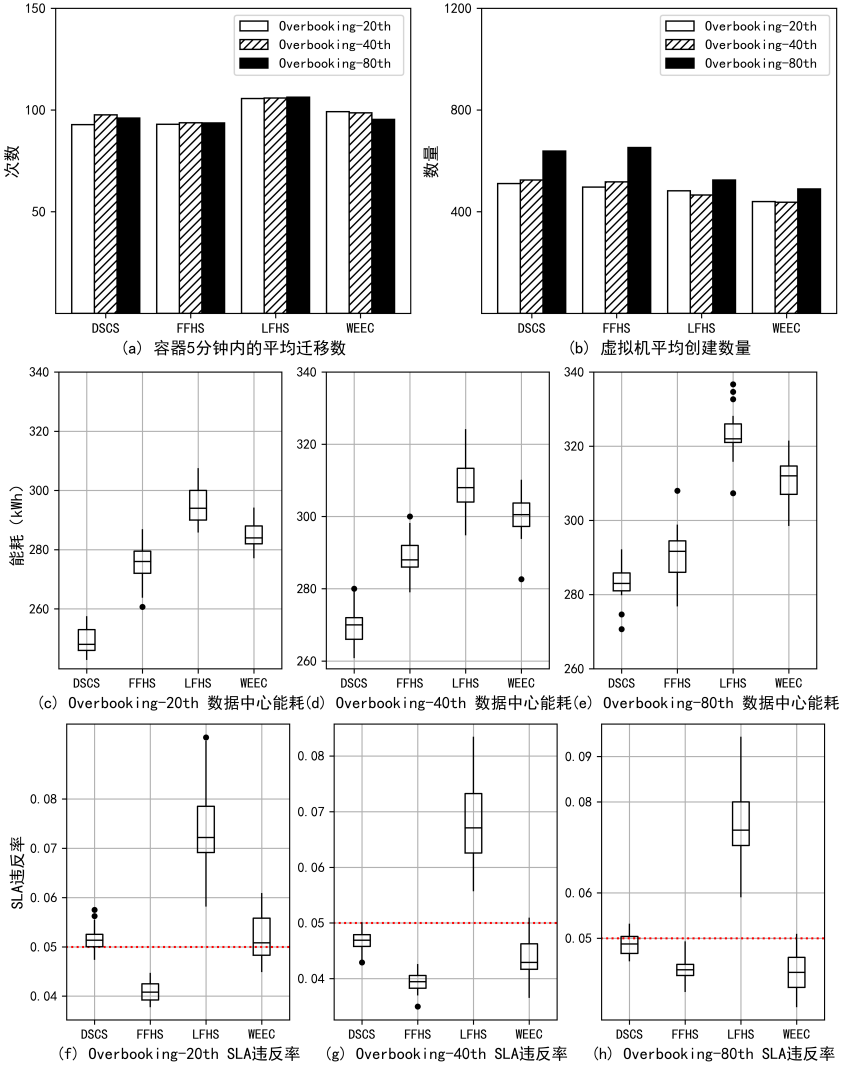

图 24: group.3 4 种调度策略容器迁移、虚拟机创建数、能耗和 SLA 对比

- 主流云服务商（GCP[2]，AWS[3]，阿里云[4]）SLA 违反补偿协议

赔偿（相对于租金比例）	SLA 违反率（未响应时间占比）
0	0 ∼ 5%
5	5% ∼ 15%
10	15% ∼ 20%
20	20% ∼ 25%
35	25% ∼ 30%
55	30% ∼ 45%
80	45% ∼ 50%
100	50% 以上

表 4: 主流云供应商 SLA 违反补偿金计算

[2] https://cloud.google.com/
[3] https://aws.amazon.com/
[4] https://cn.aliyun.com/

目录

1. 研究动机
2. 趋势感知的容器负载预测模型
3. 基于动态伸缩的容器调度策略
4. 总结与展望

总结与展望

总结

- 基于趋势感知的区间预测模型：SAC-GPSO-SVM
  - 单值预测模型 ⇒ **误差敏感，鲁棒性不足**
  - 传统区间预测模型 ⇒ **先验假设，不考虑负载类型异构**
  - 负载趋势感知（SAC）⇒ **平稳型、趋势型和周期型**
  - 针对类型构造训练区间 ⇒ **SVM 预测**（~~依赖分布假设~~）
  - GPSO 优化模型超参数 ⇒ **实时性、准确性要求**
- 基于动态伸缩的容器调度策略：DSCS
  - 动态调整容器副本数量 ⇒ **提升资源利用率，保证服务质量**
  - 基于负载相关性容器和主机选择算法 ⇒ **减少空闲主机，优化能耗**

展望

- 容器接入雾计算和边缘计算：任务跨平台可移植性
- 容器调度系统复杂性和异构性：负载预测、资源配置和节能调度

参考文献

[1] Wes Felter et al. "An updated performance comparison of virtual machines and linux containers". In: *2015 IEEE international symposium on performance analysis of systems and software (ISPASS)*. 94. IEEE, 2015, pp. 171–172.

[2] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. 95. Routledge, 2013.

[3] Joao Marques Lima. *Data Centres of the World will Consume 1/5 of Earth's Power by 2025*. 14. Desember, 2017.

[4] Nitin Anand Shrivastava, Abbas Khosravi, and Bijaya Ketan Panigrahi. "Prediction interval estimation of electricity prices using PSO-tuned support vector machines". In: *IEEE Transactions on Industrial Informatics* 11.2 (2015). 55, pp. 322–331.

Thanks