

# 一种基于机器学习的大规模计算机系统 IO性能优化方法

北京应用物理与计算数学研究所  
刘文清，景翠萍，罗红兵，武林平

2024-8-18

# 报告内容

---

一、需求背景

二、研究内容

三、关键技术

四、应用效果

五、总结

# 需求背景：高性能计算机“IO墙”问题

- I/O墙：系统规模不断扩大，计算性能不断提升，程序计算过程中产生的浮点数据越多，对存储系统的并行度和带宽要求越高，但是存储性能未跟得上计算性能，存储与计算之间性能落差日益增大，这一落差被称为“IO”墙，严重限制系统的整体发展。现象为“IO耗时占比大，IO带宽利用率低（应用IO带宽/系统峰值带宽）”。

太湖之光系统应用IO耗时占比

应用名称	IO耗时占比
dndc	18%
lammps	11%
AWP	10%

xx机应用IO带宽（持续写）

应用名称	IO带宽/磁盘	IO耗时占比
a xxx	0.95MB/s	20.69%
g xxx	0.88MB/s	19.40%
l xxx	0.36MB/s	26.40%
A xxx	0.44Mb/s	22.28%

系统理论值：~600MB/s/磁盘

相差2个量级！！  
系统IO带宽利用率不足1%

急需通过提升系统IO带宽利用率来提升数值模拟程序性能！！

一种基于机器学习的系统IO性能优化方法

特点：1) 系统性能优化 2) 可移植性强，适合多种平台 3) 普适性

注：lammps(分子动力学程序)，dndc(地球海洋模拟)，AWP(地震模拟)

# 报告内容

---

一、需求背景

二、研究内容

三、关键技术

四、应用效果

五、总结

# 研究内容：一种基于机器学习的系统IO性能优化方法

## 技术问题

## 技术难点

如何发现性能问题

规律挖掘

IO特征数据：

<时间、IO特征1、……>

特点：时序、高维

难点：高维度数据，数据量大，挖掘规律难

系统性能瓶颈点  
定位

诊断

- IO软件栈复杂，IO硬件链路长，定位瓶颈难
- 系统配置参数>10000，IO的参数>100个。
- 遍历测试参数代价高，耗时长，不现实。

优化

# 研究内容：一种基于机器学习的系统IO性能优化方法

## 技术问题

## 技术难点

规律挖掘

诊断

优化

需保证性能开销低

系统规模大（2千+节点），  
“牵一发而动全身”，  
更改需最小化系统干扰

作业模型不同，如何高效  
快速评估400+用户2万个  
作业的性能优化幅度

# 研究内容：一种基于机器学习的系统存储性能优化方法

## 关键技术

基于聚类分析的IO运行时规律挖掘

基于回归分析的系统瓶颈参数定位与优化

规律挖掘

诊断

优化

## 应用效果

提升  
数值模拟程序运行效率

提升系统核时占比  
TOP7用户程序的效率

# 报告内容

---

- 一、需求背景
- 二、研究内容
- 三、关键技术
- 四、应用效果
- 五、总结

# 关键技术1：基于聚类分析的IO运行时规律挖掘

## 数据收集

## 基于聚类分析挖掘规律



输入：样本集 $X=\{x_1, x_2, \dots\}$ ， $X_i$ 为磁盘A一周所有作业的IO数据量/次，簇数 $k$ ，输出簇划分

step1: 随机选 $k$ 个初始簇中心；  
step2: 计算各数到各簇中心欧式距离；

step3: 按照距离公式将各数划分到最近的簇；

step4: 计算簇均值，若中心变化更新簇中心，若没有，结束



60GB性能数据  
挖掘规律难

发现20.45% (5.6万) 个  
作业“IO块”存在最大值  
(4MB/次)

规律：  
“IO块最大值”存在上限（  
4MB/次）

注：IO块=IO数据量/次

# 关键技术2：基于回归分析算法的系统瓶颈定位与优化方法

## 诊断

Lasso回归分析，是一种数据挖掘方法，即在常用的多元线性回归中，添加惩罚函数，不断压缩系数，从而达到精简模型的目的，当系数为0时，达到筛选变量的效果。

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \dots + \beta_n X_n$$



引入惩罚函数，压缩系数，当 $\beta_1=0$ 时， $X_1$ 被剔除方程



$$Y = \beta_0 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \dots + \beta_n X_n$$

## 优化

### 修改系统配置

IOR运行参数 (总IO数据量/ 每次IO数据量)	IOR (4个点)
	IO性能提升幅度
8G/8M	28%
8G/16M	24%
8G/32M	27%
16G/16M	54%
16G/32M	26%

### 瓶颈：

文件系统的配置影响程序IO带宽，进一步影响了系统IO带宽利用率

### 性能开销：

CPU 0.42% ↑  
内存 5.58% ↓

# 报告内容

---

一、需求背景

二、研究内容

三、关键技术

四、应用效果

五、总结

# 应用效果1：提升IO密集型数值模拟程序运行效率

## 可提升程序类型

lxxx	sxx
jxxx	gxxx
mxxx	axxx
Axx	rxx
Cxxx	

## IO密集型数值模拟程序运行效率提升示例

程序	并行规模	IO数据量 (GB)	优化前时间 (小时)	优化后时间 (小时)	性能提升百分比(%)
rxx	48	52	27.32	20.2	26.22
lxxx	480	245	16.1	12.3	22.66
axxx	960	4128	24.17	21.43	9.6
sxxx	32	499.2	39	36	7.69
axxx	256	1090	212.21	200	5.82

调优策略对大部分程序  
有普适性

应用：

- 策略有普适性，支撑2589个作业性能提升
- 性能提升最大26.22%，平均14.46%，节约核时62.67万/月

# 应用效果2：提升核时占比TOP7用户程序的效率

## TOP7用户程序效率提升幅度统计

用户	程序	提升作业个数占比	提升最大幅度% (IO)	提升最大幅度% (整体)
用户A	lxxx	64.2	48.1	22.66
用户B	Axxx	70.27	36.3	22.19
用户C	gxxx	87.18	38.79	26.1
用户D	gxxx	27.96	13.8	3.5
用户E	axxx	37.76	21.2	9.6
用户F	gxxx	91.16	24.3	9.5
用户G	lxxx	91.02	34.09	22.67

7个用户  
占总核时  
24.6%

作业个数  
占比平均  
67.08%

最大优化  
26.1%

### 应用效果：

- 提升TOP7用户性能
- 性能最大提升26.1%
- 平均提升67.08%作业的效率

注：

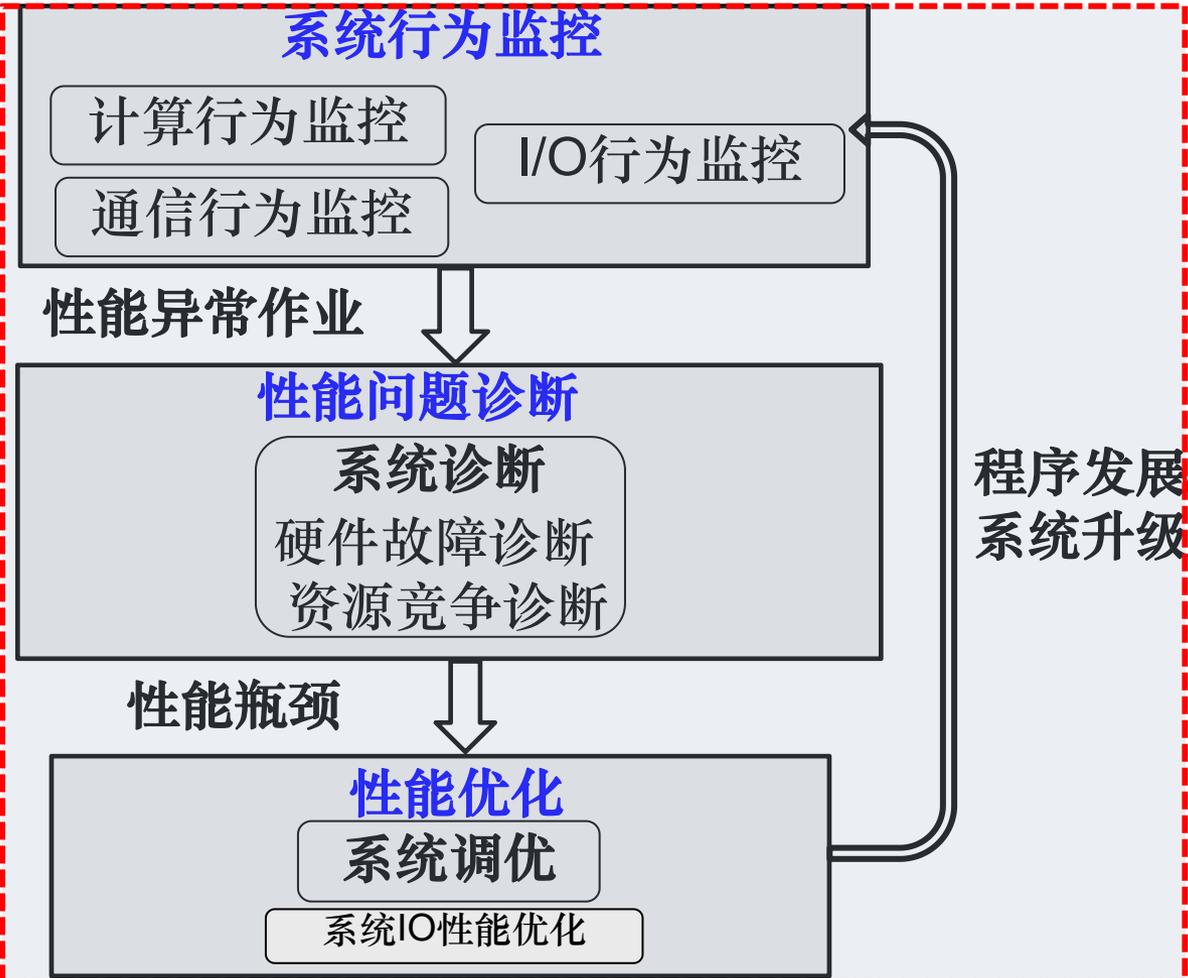
最大提升幅度 (IO) =  $\max(\text{节约时间} / \text{IO运行时间})$

最大提升幅度 (整体) =  $\max(\text{节约时间} / \text{运行总时间})$

# 集成到JSysCT平台

■ 面向高性能计算机上数值模拟程序的运行时效率提升需求，研制“系统运行时性能诊断与优化平台JSysCT”。将IO性能优化方法集成到了JSysCT平台

- 01 轻量级监控
- 02 细粒度定位
- 03 协同优化



研制软件产品 **JsysCT**

满足高性能计算机的性能诊断与优化需求

# 报告内容

---

一、需求背景

二、研究内容

三、关键技术

四、应用效果

五、总结

# 总结

- 一种基于机器学习的大规模计算机系统IO性能优化方法
  - 关键技术
    - 基于聚类分析的IO运行时规律挖掘
    - 基于回归分析算法的系统瓶颈定位与优化方法
  - 性能评估
    - 可接受性能系统开销下，提升2589个作业性能；
    - 作业IO性能最高提升48.1%，整体最大提升26.22%
    - 节约核时62.7万/月

# 致谢

---

谢谢聆听！