面向超级计算机系统的可视化综述

吕斐[1)], 陈长建[1)*], 张嘉鹏[1)], 冯冼[2)], 唐卓[1)]

[1)] (湖南大学信息科学与工程学院 长沙 410082)
[2)] (湖南省气象信息中心 长沙 410118)
(changjianchen@hnu.edu.cn)

摘 要: 随着科技飞速发展, 超级计算机系统不断向着更大规模迈进, 给系统使用和维护带来困难. 可视化技术以直观易懂的方式展示超级计算机系统的运行状态、异常情况等, 让用户更加深入地理解系统内部运行机制, 有助于更好地使用和优化系统. 通过文献调研, 分析了超级计算机系统可视化的相关工作, 并根据系统使用流程将其分为状态监控、性能优化和系统维护 3 个方面. 分析表明, 可视化在状态监控方面可帮助分析系统运行时内部进程和数据的状态; 在性能优化方面可帮助定位性能瓶颈并优化; 在系统维护方面可帮助发现运行异常以及对异常溯源. 最后, 详细阐述了未来超级计算机系统可视化在大规模数据处理、全局优化、下游任务迁移等方向上面临的机遇和挑战.

关键词: 超级计算机; 可视化; 状态监控; 性能优化; 系统维护
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024.2023-00791

Visualization for Supercomputer System: A Survey

Lyu Fei[1)], Chen Changjian[1)*], Zhang Jiapeng[1)], Feng Xian[2)], and Tang Zhuo[1)]

[1)] (*College of Computer Science and Electronic Engineering, Hunan University, Changsha* 410082)
[2)] (*Hunan Meteorological Information Center, Changsha* 410118)

Abstract: With the rapid development of science and technology, supercomputer systems are growing in size and complexity, which poses challenges for their usage and maintenance. Visualization provides an intuitive way to reveal the status of supercomputer systems. It enables users to gain a deeper understanding of the internal mechanisms of the systems, making it easier to use and optimize them. This survey summarizes the related work on supercomputer system visualization and classifies them into three categories: status monitoring, performance optimization, and system maintenance. Our survey shows that visualization helps analyze the status of internal processes and data for status monitoring; identify and optimize performance bottlenecks for performance optimization; and discover and analyze anomalies for system maintenance. Finally, we discuss future research opportunities, such as large-scale data processing, global optimization, and downstream task migration.

Key words: supercomputer; visualization; status monitoring; performance optimization; system maintenance

---

收稿日期: 2023-12-09; 修回日期: 2024-03-05. **基金项目**: 国家重点研发计划(2021ZD40303); 国家自然科学基金(62225205, 92055213, 62302160); 湖南省自然科学基金(2021JJ10023); 湖南省科技创新计划(2023ZJ1080); 长沙市科技重大专项(kh2301011); 国家自然科学基金-深圳基础研究计划(JCYJ20210324140002006). **吕斐**(2002—), 男, 硕士研究生, 主要研究方向为可视分析; **陈长建**(1994—), 男, 博士, 助理教授, 硕士生导师, CCF 会员, 论文通信作者, 主要研究方向为弱监督机器学习、可视分析; **张嘉鹏**(1995—), 男, 博士, 助理教授, 博士生导师, CCF 会员, 主要研究方向为分布式计算、并行计算; **冯冼**(1978—), 男, 硕士, 高级工程师, 主要研究方向为数据处理、人工智能; **唐卓**(1981—), 男, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为高性能计算、云计算.

1 相关工作

随着科学技术的发展，现代计算机的计算能力也在不断进步. 最新一代的超级计算机浮点计算处理能力达到了每秒百亿亿次级别[1]，在气象气候监测预警、高精度航空航天材料模拟、生物基因工程等数据规模庞大、计算效率要求极高的任务中起到了至关重要的作用[2-3]. 为了实现高性能高效率的计算，超级计算机采取并行计算的处理方式，利用多个处理器同时执行不同的任务，并通过通信实现不同处理器之间的同步[4]. 这些机制使得超级计算机系统的内部结构远比普通计算机复杂，给系统的使用带来了困难.

参考 Asaadi 等[5]总结的超级计算机系统应该具备的功能，本文凝练了其典型使用流程，如图 1 所示，并发现主要存在 3 个方面的困难. (1) 状态监控. 超级计算机系统规模庞大，内部情况错综复杂，对其内部计算过程中的状态变化进行高效实时监控极具挑战性. (2) 性能优化. 超级计算机系统拥有众多模块，彼此相互影响，系统的计算能力未被最大化利用，存在性能瓶颈. 然而，优化系统中潜在的性能瓶颈是一个十分费时费力的过程. (3) 系统维护. 超级计算机系统日常使用过程中需要进行维护，保障其能够持续高效稳定运行. 检测异常情况的常用方法是分析日志文件数据，但由于日志文件数据规模庞大且结构复杂，往往难以及时准确地检测出异常情况.

针对以上问题，可视化技术以其直观易懂的优势[6-11]展现系统运行过程中的运行状态，高效地分析大规模日志数据，帮助用户深入理解系统内部运行机制、探索潜在的性能瓶颈、及时准确地检测出异常情况. 近年来，已经有一些对面向计算机系统的可视化进行总结和探讨的综述文献. Isaacs 等[12]总结计算执行过程中的性能数据可视化方法，涵盖了硬件系统、程序调用路径、故障检测以及节点通信网络等多个方面；张炜等[13]总结面向计算机程序的可视化方法，将计算机程序划分为执行前、执行中、执行后 3 个阶段，并对每个阶段详细介绍了当前已有的可视化方法. 然而，由于超级计算机系统的数据规模庞大、数据结构复杂[14]，将这些面向普通计算机系统的方法应用于超级计算机系统时，会遇到视觉混乱、数据处理效率低等问题，因此不适用.

为了全面了解面向超级计算机系统的可视化研究进展，本文采取三阶段的检索方法[15-21]. (1) 利用 Google 学术搜索引擎在可视化知名刊物上以“High Performance Computing” “Supercomputer” “Cloud Computing”之一与“Visualization”组合作为关键词进行检索，得到 400 余篇文献；(2) 通过人工筛选，去除与“并行处理” “大规模计算” “可视化”无关的内容，得到 22 篇相关文献；(3) 对 22 篇相关文献中引用的参考文献进行人工筛查，最终得到 34 篇文献，并对它们进行系统性的总结与分类. 收集流程如图 2 所示.

本文基于超级计算机系统的典型使用流程，对该 34 篇文献进行分析与归纳总结，将使用流程分为状态监控、性能优化与系统维护 3 个方面；然后分别对每个方面的可视化工作进行系统性的介绍与总结；最后阐述面向超级计算机系统可视化的研究机遇与挑战. 本文将搜集的文献整理在 https://github.com/hnu-vis/supercomputer-system-survey，供研究人员参考.

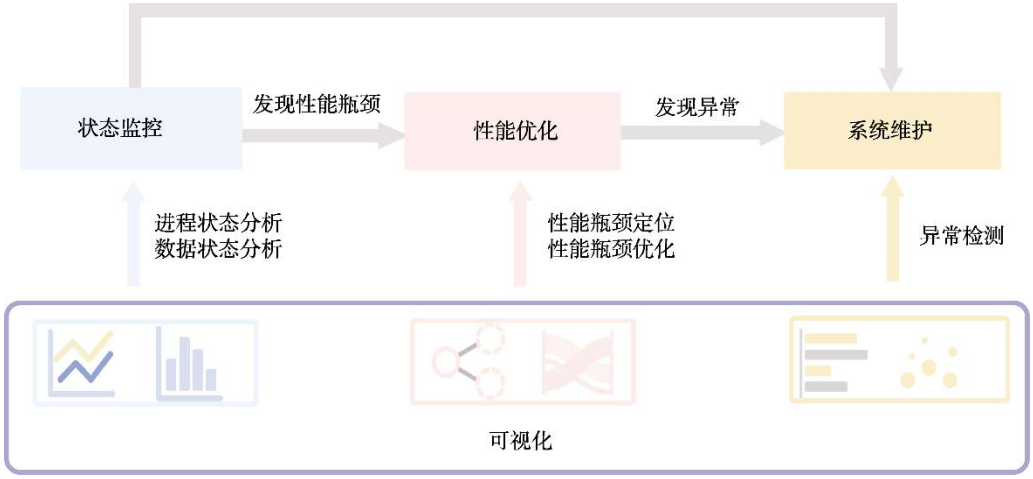


图 1 超级计算机系统的典型使用流程

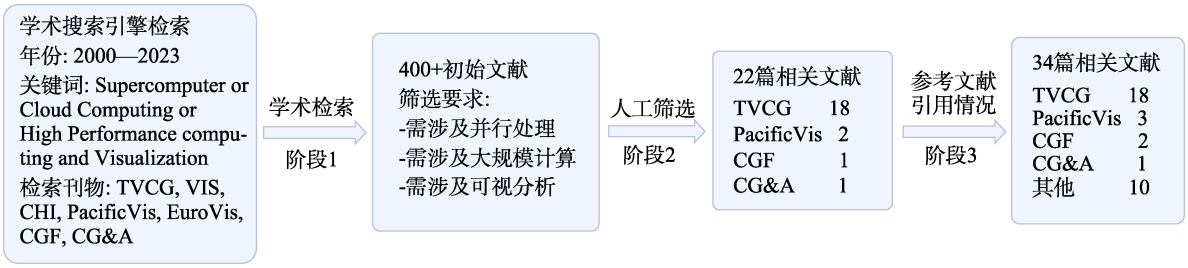


图 2　相关文献检索流程

2　状态监控

超级计算机系统的状态监控十分关键，能够让用户在系统运行过程中实时监控系统内部情况，保证系统高效稳定地运行．高效的可视化技术能够提供直观的图像和图表展示系统内部复杂的运行情况，有助于理解和分析系统的运行过程．根据监控的内容，状态监控可视化分为进程状态可视化与数据状态可视化．

2.1　进程状态可视化

为了提高计算效率，超级计算机系统通常采取进程并行和分布式计算等方法，利用大量进程共同协作完成任务，导致进程分布结构变得极其复杂[22]．可视化技术可以直观地展示进程执行情况和进程间的通信情况，帮助用户清晰地观察和分析不同进程的活动状态，从而更好地管理和优化整个大型计算系统[23]．

考虑大规模超级计算机系统中存在大量进程，单独分析每个进程的消息传递接口(message passing interface, MPI)会严重超出屏幕分辨率，影响可视化的使用，Muelder 等[23]提出可拓展的 MPI 可视化工具，如图 3 所示．该工具从 MPI 调用的角度出发，使用时间线概述[24-25]，其中，横轴表示时间，纵轴的高度与持续的长度分别表示调用 MPI 的进程比例和 MPI 调用的持续时间，每个染色点代表一个常用的 MPI 调用．通过这种可视化方式，用户能观察到不同密度的 MPI 调用，理解分析并行计算系统中的进程状态．

针对进程之间的行为，Karran 等[26]采用围绕公共圆心的弯曲冰柱图的可视化方法分析进程执行轨迹，并展示多个进程之间的交互行为，提出针对进程之间交互行为的可视分析工具 SYNCTRACE，如图 4 所示．SYNCTRACE 分为概览视图(Overview window)、主视图(Main View)和上下文视图(Context view) 3 部分，当用户选定需要分析的进程后，概览视图中会呈现进程间的同步关系，同时，主视图中的冰柱图呈现更详细的进程轨迹，上下文视图则使冰柱图弯曲并围绕圆心以呈现多进程之间的联系．这项技术旨在分析进程之间的关系，有助于更好地理解进程的大规模并发过程．在此基础上，为了解决进程数量大造成渲染效率过慢、可视化元素重叠等问题，Sigovan 等[27]将相关进程合并成为进程组，通过粒子动画(Particle animation)技术展示并行程序执行过程中进程之间的动态通信模式；经过测试，该方法能够胜任同时分析超过 16 000 个进程的要求．然而，上述工作只能分析当前时刻或较短时间段内的进程行为，无法对长时间跨度下的进程行为进行分析．

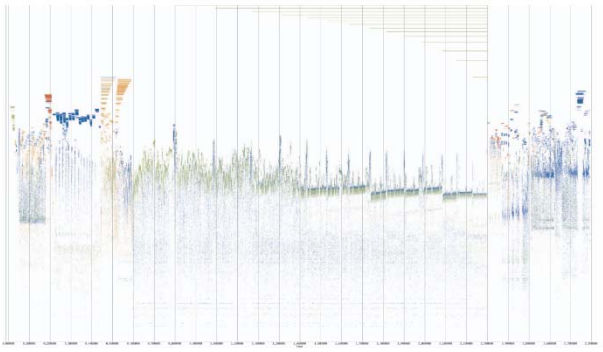


图 3　可拓展的 MPI 可视化工具[23]

为此，Sakin 等[28]与高性能计算专家合作，开发了一个用于任务并行跟踪的集成可视化平台 Traveler，如图 5 所示．Traveler 关注执行过程中产生的痕迹数据，通过利用率视图展示所选任务类型占程序的比例，如图 5 中的 a 所示；甘特图展示系统单一任务和任务组的调度过程，如图 5 中的 b 和图 5 中的 c 所示；功能框绘图视图总结跨资源和跨时间变化的性能指标，如图 5 中的 d 所示；信息视图和代码视图展示所选任务的具体描述信息和源代码，如图 5 中的 e 和图 5 中的 f 所示；依赖树视图展示任务间的依赖关系，如图 5 中的 g 所示；区间直方图视图展示任务持续时间分布，支持用户在时间跨度较大的情况下分析进程状态，如图 5 中的 h 所示．

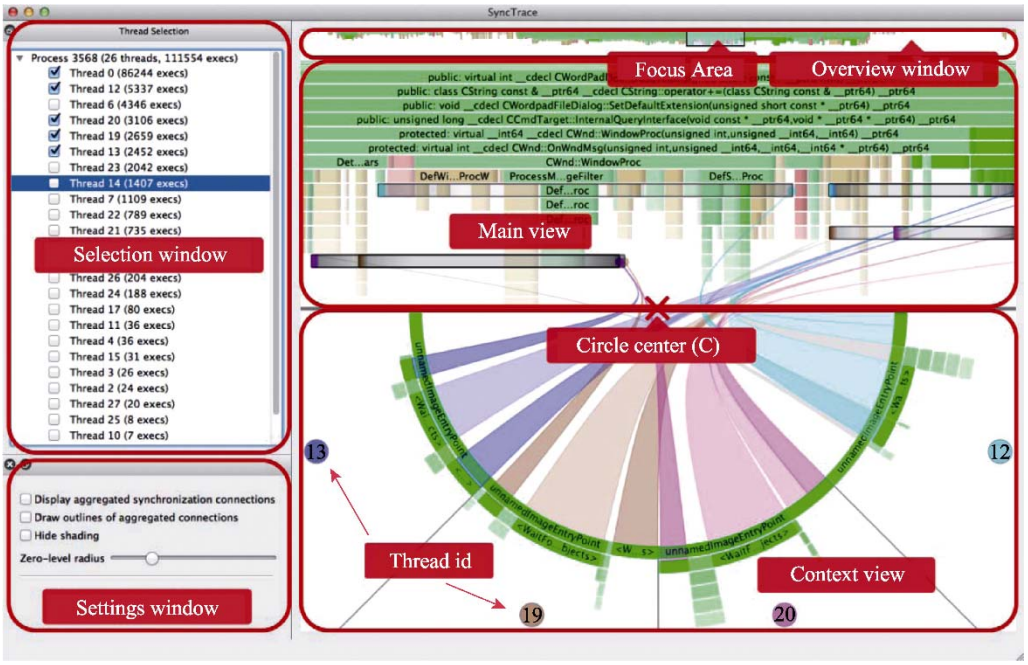


图 4 SYNCTRACE 工具[26]

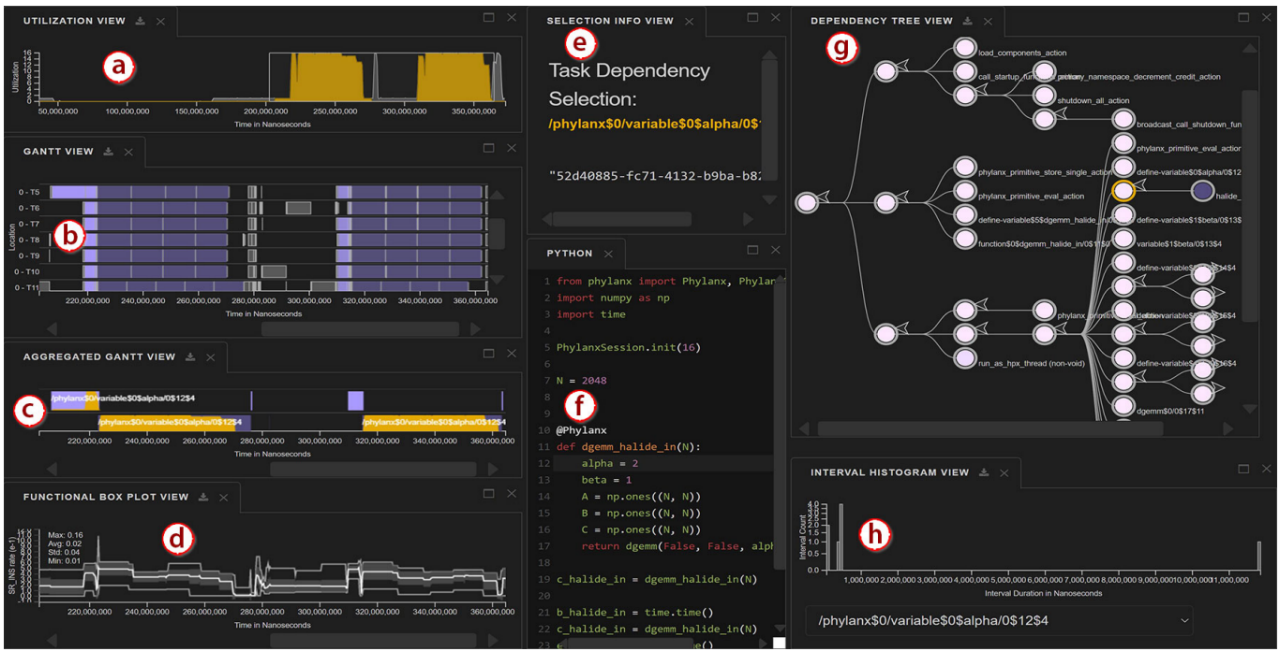


图 5 Traveler 平台[28]

2.2 数据状态可视化

超级计算机系统通常面向大规模数据处理，通过可视化直观地展示数据变化，用户能够更深入地理解超级计算机系统任务执行中数据的传输、处理等过程，并通过可视化高效地分析超级计算机系统的运行状态. 根据超级计算机系统中数据可视化的处理方式，可将其归纳为数据状态原位可视化与数据状态离线可视化[29].

数据状态原位可视化. 随着超级计算机在处理科学模拟任务时的数据量不断增大，对数据可视化会耗费大量的存储与数据传输资源[30]. 为了解决这一问题，在超级计算机处理大规模任务的同时，原位可视化技术将数据在原位进行可视化处理，降低对数据传输和存储的需求[31]. Dutta 等[32]指出，目前，原位可视化技术可分为原位可视化表示方法和交互式原位可视化方法.

早期的原位可视化主要关注于如何高效地对数据生成可视化表示(Visual representation)，此类

方法可以方便用户查看当前超级计算机系统中的数据状态. 原位可视化最初的概念由 Haimes[33]提出，用来可视化科学模拟过程中产生的大规模数据；此项工作推动了后续很多开源的原位可视化工具和代码库的研发，包括 VisIt[34]，ParaView[35]，ADIOS[36]和 GLEAN[37]等. 针对一些具体应用场景，研究者也有针对性地对原位可视化技术进行改进，如海洋气候[38]、海洋涡旋[39]、社交媒体[40-41]和粒子加速器[42]等. 此外，部分工作对原位可视化表示方法进行归纳总结. Ma[43]在探讨大规模复杂计算数据可视化问题时，着重强调数据原位可视化的必要性，并提出数据原位处理的基本过程，包括对数据进行压缩、提取关键特征、构建数据层次结构或创建快速索引；单桂华等[44]将科学计算过程的数据原位可视化分为数据压缩、特征处理与跟踪以及可视化绘制 3 个关键步骤，并针对每个步骤介绍了相关算法或可视化处理方法；Bauer 等[45]总结在科学和工程问题中，针对超级计算机系统使用原位可视化的方法，并详细阐述了在气候建模、结构工程与能量建模等存在大规模复杂数据模拟问题中的原位可视化应用.

近年来，研究者逐渐发现，要完成超级计算机系统的分析任务，不仅仅需要将数据用可视化形式展示，还需要支持用户交互式的探索、分析和验证[32]. 为此，交互式原位可视化方法被提出，此类方法使用原位处理技术将数据进行缩减，用户可以在缩减后的数据上进行交互，帮助他们交互式探索超级计算机系统中的数据状态[46]. Woodring 等[47]提出一种对大规模粒子模拟进行采样的方法，并将采样后的数据进行保存，供用户交互. Ahrens 等[31]将数据转化为图像并存储，在减少存储量的同时也能支持用户交互探索；Dutta 等[32]通过概率密度函数拟合数据，让用户通过该函数对数据进行交互分析；在此基础上，He 等[48]使用深度神经网络代替概率密度函数拟合数据. Li 等[49]等针对深度神经网络模型的分析任务，利用原位可视化技术高效率、低成本的特点，构建一个基于原位可视化的分析框架，通过特征提取算法获取训练过程中的时序数据的主要特征，并通过原位可视化技术将其呈现，最后通过桑基图与网格布局，帮助用户高效地分析模型每个训练轮次、每层网络的训练状态的变化过程.

数据状态离线可视化. 除了对超级计算机系统中被处理的数据感兴趣，用户也希望分析系统中数据日志信息，帮助他们更好地理解数据模式、内在联系和全局数据的变化趋势. 与原位可视化处理的大规模数据不同，数据日志信息数据量相对较少，数据存储和传输不再是瓶颈. 为此，离线可视化主要关注于设计有效的可视表示和交互来更好地帮助用户分析数据日志信息. 为了帮助用户理解系统计算过程中的网络活动，Landge 等[50]通过一个直观无重叠的网格视图[51-55]展示节点之间的网络连接关系，以及一个用户熟悉的三维节点-链接图来展示节点之间的物理网络结构，帮助用户分析网络流量的趋势和模式；Shilpika 等[56]则针对计算过程中硬件系统产生的时序数据进行可视分析，提出一个增量渐进式算法来加速幅度-形状图的生成以呈现时间序列数据的变化趋势，并结合函数型数据主成分分析(functional principal component analysis, FPCA)结果视图帮助用户识别出时间序列数据中的错误信息；Osmari 等[57]针对应用程序执行过程中产生的轨迹数据，提出一个高效率的数据轨迹可视分析框架 SmartTrace，如图 6 所示. SmartTrace 中，任务并行执行完成，如图 6 中的 a 所示，对任务执行生成的多类型轨迹数据进行收集，如图 6 中的 b 所示，用户根据需求选择合适的可视化和数据流，如图 6 中的 c 所示，该框架自动生成可视化，供用户进行探索和分析，如图 6 中的 d 所示.

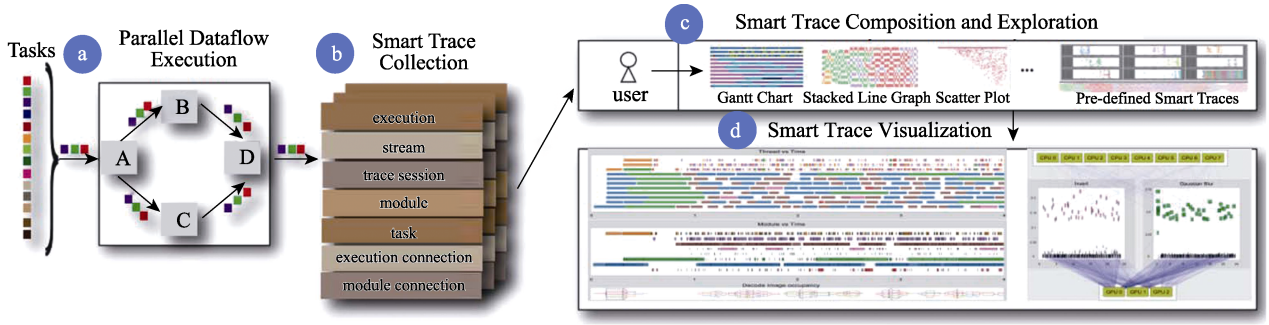


图 6　SmartTrace 框架[57]

3 性能优化

目前，超级计算机的运算速度已经达到了新的高度，但是突破性能瓶颈一直是领域内的难题. 超级计算机系统常常有极其复杂的架构，涉及多个处理器核心、大规模的内存系统等，其复杂性导致在优化过程中会面临“牵一发而动全身”的问题，使得优化性能变得越来越具有挑战性. 可视化技术能够利用计算过程中的数据帮助用户进行分析、探索性能瓶颈，从而帮助性能优化. 根据性能优化的过程，性能优化可视化可以分为性能瓶颈定位与性能瓶颈优化.

3.1 性能瓶颈定位

随着大规模并行系统与分布式计算的复杂性日益增高，进一步提高超级计算机系统的性能十分具有挑战性，需要高效地探索系统存在的性能瓶颈. Isaacs 等[58]从分析进程事件的角度出发，提出一种研究进程事件与进程间消息记录的可视化方法，将事件历史记录中的物理时间转化为逻辑时间，并根据逻辑时间线的性质对进程进行聚类，得到一个并行交互式逻辑时间线可视化，帮助用户分析潜在的性能瓶颈. 通过 2 个案例研究，他们发现并解释了在并行应用程序和大规模科学模拟中的性能瓶颈问题，验证了该方法的有效性；但受限于逻辑时间线长度，这种方法不能准确地定位性能瓶颈的位置与产生的原因.

调用文本树(Call text tree)能够有效地解决这个问题. 调用文本树是一种常见的用来展示代码结构和函数调用关系的方式，其通过文本的形式用树状结构呈现程序中函数调用的序列，并提供层级视图，清晰地展现了代码中函数之间的嵌套和调用关系. 在此基础上，Nguyen 等[59]使用调用上下文树(Call context tree)的数据结构，除了基本的函数调用序列外，还额外收集了函数调用时的上下文信息，如运行时的环境变量、函数参数以及执行状态等，结合超图的概念，设计并实现了一个用户可控的可视分析系统 CallFlow，用于分析大规模的调用上下文树；其不仅提供对调用上下文树的整体概览[60-61]，还运用桑基图与基于流的视觉隐喻[62]，将执行时间视为调用序列中消耗的资源，帮助用户准确地定位存在性能瓶颈的位置. 此外，Nguyen 等[63]针对传统可视化方法在性能分析过程中丢失进程的行为的问题，开发了可视化分析工具 VIPACT，对所有进程的运行时分布进行编码，渲染成光环的形式，并将光环作为调用树的节点，确保用户能够在保持进程行为过程完整的情况下分析进程的性能数据. 这些方法使用户能够高效地探索超级计算机系统中哪些函数被频繁调用或消耗了大量时间，从而更快地发现导致性能瓶颈的原因并准确定位需要优化的位置. 另一种表示函数调用关系的可视化方法——调用图(Call graph)也十分常用，其采用节点与线段相连的可视化方式，能够更为直观地呈现函数的调用全貌与依赖关系以帮助用户分析. 但是，在大规模并行程序中函数调用关系极其复杂，生成的调用图的规模十分庞大. 针对大规模图的可视化，Archambault 等[64]和 Gansner 等[65]利用有向图中存在的拓扑关系，提出大规模图高效可视化的方法；Kesavan 等[66]则在 CallFlow 的基础上进行扩展，开发出新的交互式分析工具 Ensemble CallFlow，并提出视觉设计 ensenmble-Sankey，如图 7 所示. Ensemble CallFlow 工具利用桑基图呈现调用的上下文和相关性能指标，如图 7 中的 a 所示；使用箱线图呈现调用的执行时间等分布情况，如图 7 中的 b 所示；并将其与多个关联视图相结合，帮助用户探索分析大规模调用图，并对性能瓶颈进行准确定位.

为了应对日益具有挑战性的性能优化任务，对高效率性能分析工具的要求也在不断提高. Shende 等[67]详细阐述了调试和分析实用程序(TAU)并行性能系统的结构与实现过程，以应对大规模并行计算程序的性能瓶颈定位与可视化的要求；Adhianto 等[68]推出一套集成工具 HPCToolkit，支持对并行应用程序的计算性能进行测量、分析以及可视化展示. 这些工具能够满足特定规模的性能分析需求，并提供相应的可视化呈现，帮助用户定位系统中可能存在的性能瓶颈.

3.2 性能瓶颈优化

在准确地定位系统存在的性能瓶颈之后，需要对其进行优化. 优化超级计算机系统的性能瓶颈涉及多个系统模块，如优化数据处理与分析流程、优化内存性能等，这些综合性的优化策略确保系统中每个关键模块都能处于最佳运行状态，为超级计算机的整体性能提升奠定了坚实的基础.

针对大规模数据处理与分析任务的性能优化问题，为了提高并行计算过程中大规模流式数据分析任务的效率，Kesavan 等[69]开发了一个可视化分析框架，包括数据管理、数据分析和交互式可视化 3 个模块，帮助用户高效地分析流式数据的变

化；其中，数据管理模块负责收集数据并反馈到其余 2 个模块，数据分析与交互式可视化模块则分别对流式数据进行渐进式的分析与可视化呈现．但是，大规模数据不仅数据量庞大，通常还包括多项随时间变化的性能指标数据，导致其结构变得异常复杂．Jo 等[70]提出对大规模多维数据进行交互式数据可视化的系统 SwiftTuna，利用内存计算引擎 Apache Spark 加速可视化渲染过程，并提供即时的低保真响应以及延迟的高保真响应实现高效率可视化；此外，他们还设计了新的交互式尾部图表(Tailed chart)，如图 8 所示．其中，当用户选定需要分析的变量数据时，系统会展示出与之对应的尾部图表来呈现可视化估计值的置信区间，如图 8 中的 A 所示；以及密度图来呈现不同变量间的关系，如图 8 中的 B 所示；以优化多维数据的探索过程；经过测试，SwiftTuna 在包含 40 亿条记录的数据集进行探索时，响应延迟保持在几秒钟之内．

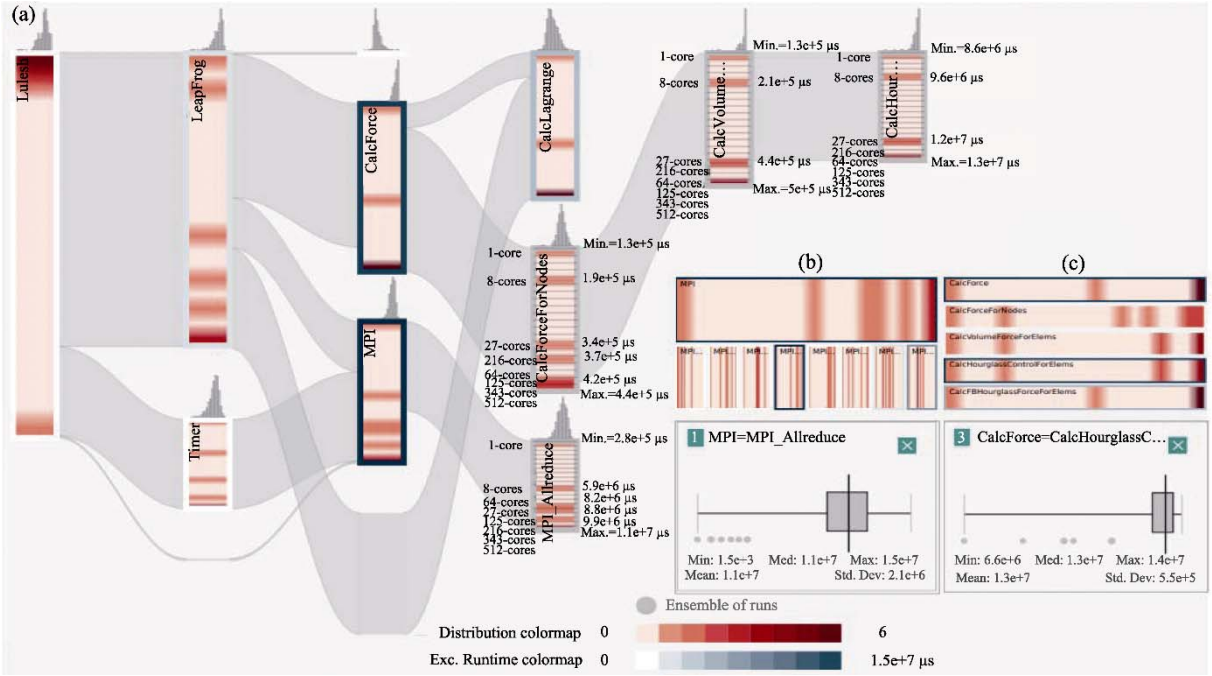


图 7　Ensemble CallFlow 工具[66]

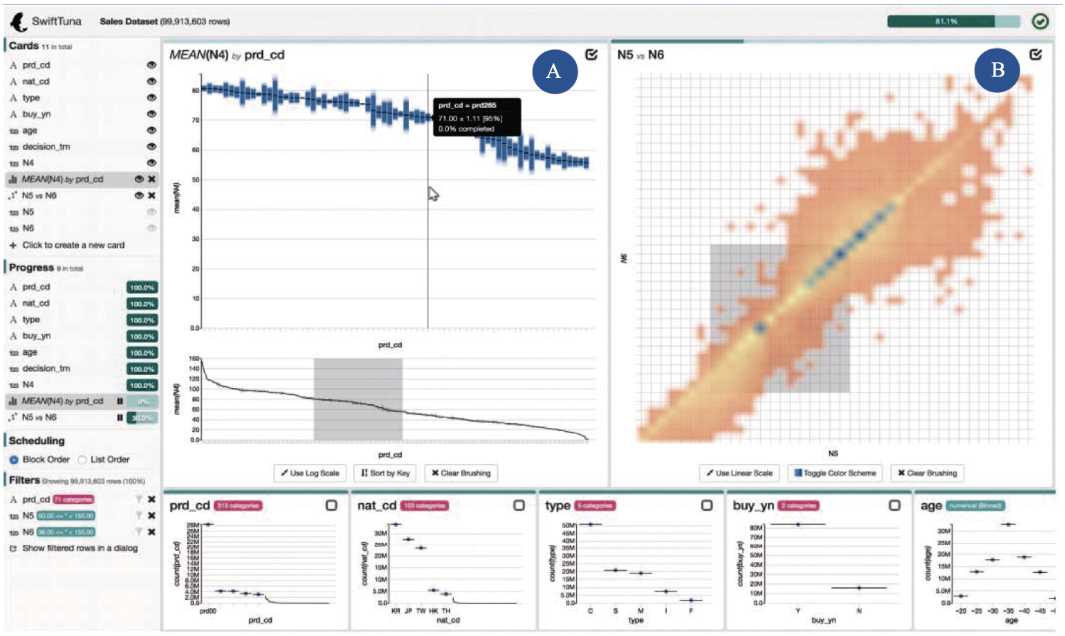


图 8　SwiftTuna 系统[70]

为了加速分析高性能计算节点产生的多元时间序列数据，Li 等[71]提出一个可视化框架，使用分层循环图、时间线图等可视化方法呈现高性能计算性能指标数据随时间的变化过程，允许用户探索高性能计算应用程序的性能瓶颈，并根据具体问题制定相应的优化策略．Muelder 等[72]则通过定期采样云计算系统执行过程的内存使用情况、网络使用情况、CPU 负载等数据指标，将其构建成为一组多元时间序列数据，并通过相似性度量的方法将每个计算节点的行为用时间线表示出来，最终从这些大规模的行为时间线中分析不同的行为模式，针对性能瓶颈制定优化策略．不仅如此，对于并行计算过程中的数据和任务划分(负载均衡)的过程，Zhang 等[73]使用节点-链接图[74-75]可视化每个进程的行为、数据的变化．其中，节点显示进程的行为，链接则代表进程之间的数据传输过程，以此促进性能分析任务并制定优化策略．

针对内存性能优化，Giménez 等[76]与高性能计算领域专家合作，设计一个内存性能数据可视化工具 MemAxes，将采样得到的内存访问数据映射到全新的视觉表达方式旭日图中，如图 9 所示．其中，将 L_1 缓存、L_2 缓存、L_3 缓存以及主内存资源分层进行染色，呈现硬件资源与内存性能数据之间的深层联系，帮助用户进行内存分析任务，探索突破内存性能瓶颈的方法．

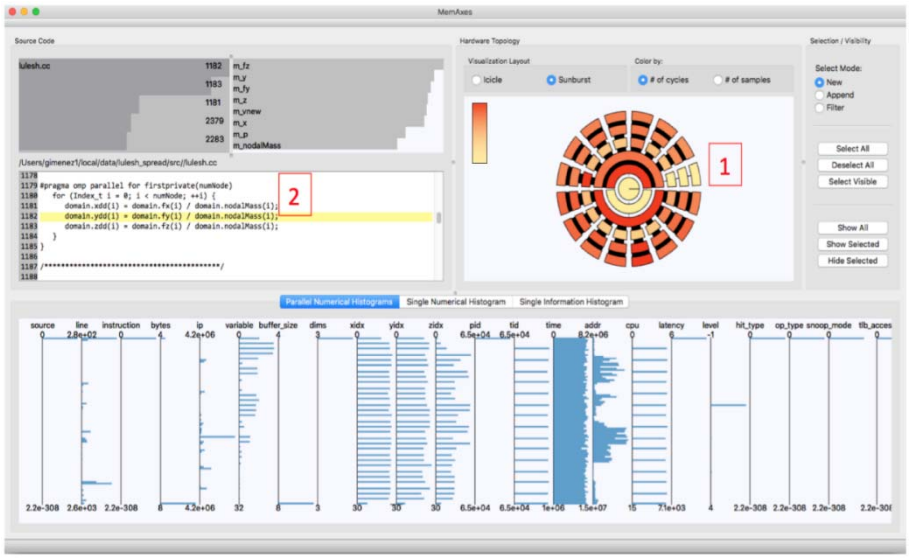


图 9 MemAxes 工具[76]

4 系统维护

对超级计算机系统进行日常维护是确保系统能够持续高效稳定运行的重要任务．在日常维护中，异常检测至关重要，其可以监测和识别系统运行中的不寻常行为、性能变化以及可能的故障．通过异常检测，用户能够及时发现超级计算机系统中潜在的问题，预防异常演变成更严重的故障，如系统计算性能下降或系统崩溃等情况．

对超级计算机系统的异常检测通常是通过分析系统运行时产生的日志信息来实现[77]．这些日志信息包括不同时间节点的各种运行指标和性能数据，结构复杂并且规模较大，用户需要借助传统的模式匹配或机器学习等技术，并结合可视化方法直观展示数据，识别出与正常行为不符的模式或事件，从而高效地发现系统中潜在的异常．

王晓东等[78]从高性能计算节点计算过程中生成的大量日志出发，首先运用主成分分析法进行异常检测，将日志的有序排列作为一种流量模式；然后采用层次聚类方法对提取出的异常模式进行简化，最后进行可视化呈现．Xu 等[79]对云计算系统计算过程中生成的多元复杂时间序列数据进行异常分析，提出一种无监督算法用于捕获时间序列中的异常模式，并设计了交互式可视化分析系统 CloudDet，如图 10 所示．CloudDet 系统能够挖掘出性能数据的异常模式，并提供异常排名视图呈现不同性能指标的平均异常分数和潜在的异常周期，从而帮助用户快速发现异常，如图 10 中的 a 所示；此外，还提供异常检查视图，如图 10 中的 b 所示；异常聚类视图用来分析异常发生的原因，如

图 10 中的 c 所示．基于这 3 个视图，用户可以通过交互的方式检查并诊断潜在的异常情况，提高对云计算系统的异常检测效率．Xie 等[80]则通过调用堆栈树对高性能计算中的异常行为进行分析，使用 Stack2vec 技术将调用堆栈树转换为向量格式，并通过降维技术和散点图展示调用树之间的相似关系[81-87]，将异常检测问题转化为在调用堆栈森林中查找异常的树形结构以简化异常检测工作，如图 11 所示．其中，提供一个散点图来展现调用堆栈树的分布情况，如图 11 中的 a 所示；在候选视图中呈现异常得分最高的树形结构，如图 11 中的 b 所示；对于被选定的树形结构，提供了节点-链接图，如图 11 中的 c 所示；以及时间线图来呈现树形结构的详细信息和调用的时间模式，如图 11 中的 d 所示；此外，还提供控制视图对选定的树形结构进行标注并更新异常检测模型，如图 11 中的 e 所示．针对大规模数据的异常检测任务，Xu 等[88]设计并实现了交互式视觉分析系统 ViDX，通过扩展的 Marey 图以及视觉聚合技术，帮助技术人员对工厂超级计算机中产生的大规模复杂性能数据进行诊断分析，并定位性能表现异常的位置．深度神经网络在训练过程也会产生大量复杂数据，Wei 等[89]针对大规模深度

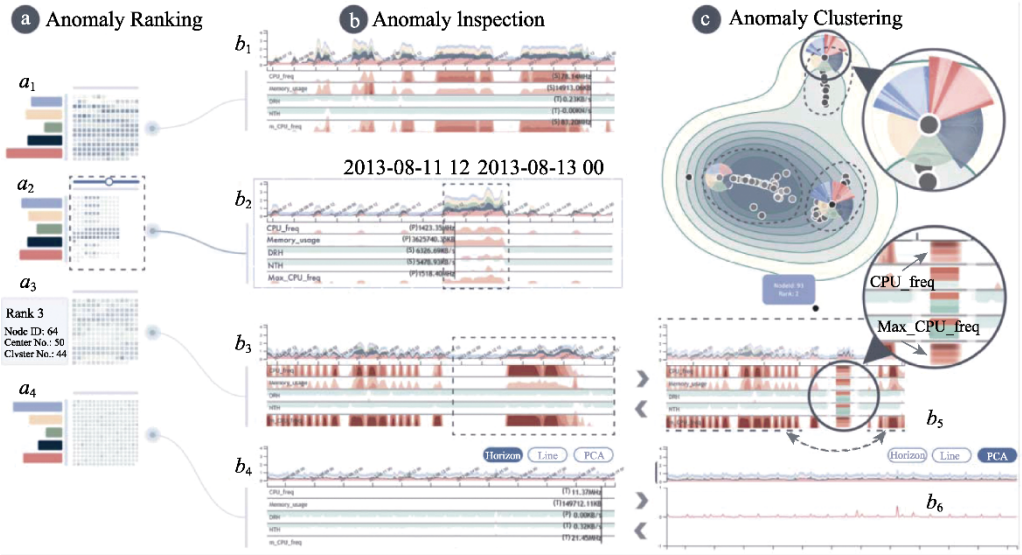


图 10　CloudDet 系统[79]

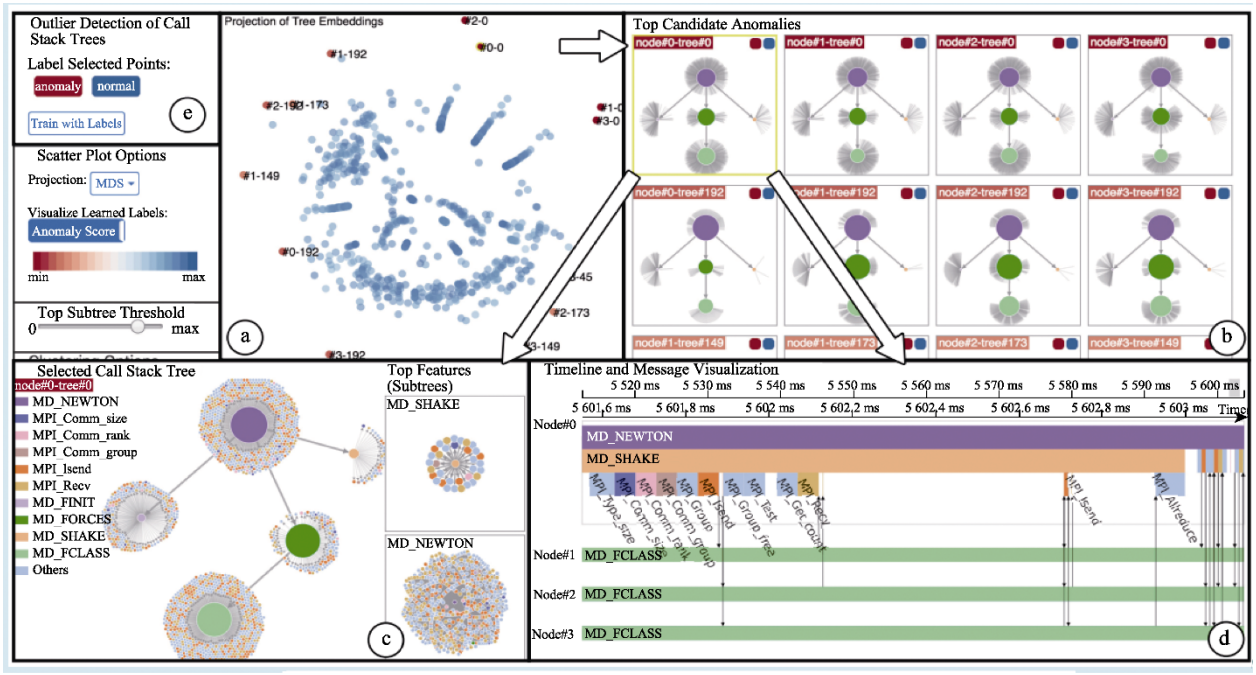


图 11　针对调用堆栈树中寻找异常树形结构的可视分析工具[80]

神经网络的训练低效问题进行研究，首先将执行序列引入计算图中，并使用并行策略视图进行呈现；然后将时间跨度与带状视觉隐喻结合在 Marey 图中，呈现复杂的数据指标之间的关系．结合这 2 种视图，能够有效帮助用户直观地探索深度神经网络模型的并行训练过程并辨识潜在的异常．

5 讨 论

结合以上对现有面向超级计算机系统的可视化工作的归纳和分析，本文对其中用到的可视化表示方法和交互方法行讨论，并对可视化技术发挥的作用进行总结，如表 1 所示．

表 1 可视化表示、交互方法和可视化技术的作用总结

超级计算机系统典型使用流程	文献	可视化表示	交互方法	可视化技术的作用
状态监控	[23,26-28,31-32,43-45,47-50,56-57]	时间线图(6)，散点图(3)，折线图(3)，节点-链接图(2)，条形图(1)，冰柱图(1)，桑基图(1)，甘特图(1)，网格视图(1)，幅度-形状图(1)，面积图(1)	探索(12)，选择(5)，关联(4)，编码(3)，抽象/具象(2)，过滤(2)	(1) 状态高效监控 (2) 数据实时分析
性能优化	[58-59,63-73,76]	散点图(5)，节点-链接图(4)，条形图(4)，网格视图(3)，时间线图(3)，桑基图(2)，折线图(2)，箱线图(1)，面积图(1)，密度图(1)，旭日图(1)，分层循环图(1)	选择(9)，关联(9)，探索(9)，过滤(6)，抽象/具象(2)，编码(1)	(1) 调用关系分析 (2) 优化策略定制
系统维护	[78-80,88-89]	散点图(4)，条形图(4)，折线图(3)，时间线图(3)，日历图(2)，节点-链接图(2)，地平线图(1)	选择(4)，关联(4)，探索(4)，抽象/具象(4)，过滤(3)	(1) 交互式异常检测 (2) 异常结果溯源

(1) 可视化表示．利用合适的方式展现超级计算机系统内部运行信息，帮助用户直观、快速地了解超级计算机系统的状态．

a. 状态监控．针对状态监控，使用较多的可视化表示方法为时间线图(6 篇)和折线图(3 篇)，这是因为超级计算机系统状态数据一般为时序数据，这 2 种图可以很好展示状态变化情况．散点图(3 篇)则是将进程或 MPI 调用以散点的形式呈现并以时间排序．此外，针对不同的数据情况与应用场景，也会采用或设计专用的可视化表示方法．例如，分析多个进程之间的交互行为时[26]，使用冰柱图展示进程执行轨迹；分析长时间跨度下的进程状态时[28]，使用甘特图、节点-链接图、条形图呈现进程的执行信息；分析深度神经网络中的数据时[49]，结合桑基图和基于网格视图的面积图来展现每层网络的数据情况；分析计算系统中的网络情况时[50]，使用节点-链接图展示网络结构；分析硬件系统产生的时序数据时[56]，使用渐进算法加速生成的幅度-形状图展示数据的变化过程．

b. 性能优化．与状态监控类似，性能优化的现有工作较多使用时间线图(3 篇)和折线图(2 篇)来分析系统执行过程中的性能指标数据；但在此方面，散点图使用更为频繁(5 篇)，这是因为除了将进程作为散点外，性能指标数据也被作为散点以帮助分析不同指标间的相关性．节点-链接图的表示方法(4 篇)也较多被应用，这是因为性能优化需要分析调用关系，其可建模为图结构，适合使用节点-链接图展示．条形图(4 篇)和网格视图(3 篇)虽然使用较多，但都是作为辅助视图来帮助更好地呈现数据分布和具体内容．例如，CallFlow[59]中，桑基图节点的条形图用于呈现调用的执行时间等分布情况；Muelder 等[72]通过网格的方式呈现多个时间线图，分析不同计算节点的时序数据．针对性能优化任务，研究者也专门设计了新颖的可视化，例如，在优化数据探索过程中[70]，使用尾部图表呈现估计值的置信区间，并使用密度图分析不同变量的关系；在分析高性能计算的时序数据时[71]，使用分层循环图呈现数据的变化过程．

c. 系统维护．与状态监控类似，由于系统维护需要分析日志信息中存在的时序数据，因此较多使用折线图(3 篇)和时间线图(3 篇)呈现数据随时间变化的过程，条形图(4 篇)也较多地作为辅助视图使用．例如，ViDX[88]中使用条形图呈现不同日期的异常情况．此外，与其他方面的散点图不同，该方面(4 篇)中的散点通常代表着数据点，从而帮助发现异常．例如，CloudDet[79]中将异常时序数据以散点的方式进行聚类；在分析调用堆栈树时[80]，将异常树形结构转换的向量视作散点并进

行聚类. 值得注意的是，时间线图中有一类特殊的可视化(Marey 图)被较多地用于系统维护. 例如，在分析生产线中生成的大规模异常数据时[88]，将 Marey 图与日历图、条形图结合，以呈现在不同的时间尺度下的异常模式；在分析神经网络模型的训练低效问题时[89]，通过节点-链接图呈现计算序列，并使用 Marey 图呈现复杂指标数据间的关系.

(2) 交互方法. 指用户与可视化系统之间的相互作用方式，帮助用户实现对数据或信息的探索、理解和操作[32]. 本文使用 Yi 等[90]提出的交互分类方法进行分类，如表 1 所示.

a. 状态监控. 针对状态监控的交互方法，探索使用最多(12 篇)，一个主要原因是部分文献属于交互式原位可视化，它们关注于如何实现对大规模数据的实时交互，为此仅使用简单的交互操作，如缩放、平移等操作进行探索. 选择(5 篇)与关联(4 篇)这 2 种方法用于多个视图之间的关联，帮助用户从多个相互关联的维度监控系统状态. 选择和关联使用占比较少的原因有 2 个：一是状态监控是最早被研究的方向，早期工作关注于设计单个有效的视图来呈现数据信息，交互较少；二是原位可视化用单个视图展示信息，交互同样较少. 值得注意的是，抽象/具象作为处理大规模数据的常用手段，仅有 2 篇被用于状态监控，这也是未来可以继续探索的一个方向. 此外，编码和过滤交互往往存在于系统中，但是功能较为常见，在正文中可能被忽略，因此数量较少. 该问题也同样存在于面向性能优化和系统维护的方法中.

b. 性能优化. 与状态监控相同，针对性能优化也有较多的工作支持探索(9 篇). 此外，选择(9 篇)和关联(9 篇)这 2 种交互方法也占多数，原因是在性能优化方面，大多数的可视化系统都采用多视图进行展示. 与面向状态监控的方法类似，支持抽象/具象(2 篇)的工作较少，这也是未来值得继续探索的一个方向.

c. 系统维护. 与性能优化相同，针对系统维护方面也有较多的工作支持选择(4 篇)、关联(4 篇)和探索(4 篇). 但与前 2 种方法不同，该方面方法较多地使用抽象/具象(4 篇)，一个可能的原因是面向系统维护的方法主要发展于近几年，同时，抽象/具象技术日渐成熟，被大家广为所用.

(3) 可视化技术的作用. 可视化在超级计算机系统的使用过程中发挥了重要作用. 对于不同阶段，可视化能起到一些共性的作用(如直观地展示系统内部数据)，也发挥了独特的作用. 下面针对不同阶段可视化所起的作用进行总结.

a. 状态监控. 可视化作用包括状态高效监控、数据实时分析. 状态高效监控中，超级计算机系统内部信息错综复杂、难以分析，可视化以直观的形式展示这些信息，帮助用户高效监控系统运行. 数据实时分析中，为了及时了解超级计算机系统内部数据状态，需要实时对数据进行分析. 为此，有很多工作关注原位可视化，从而实时地对系统内部数据进行展示，帮助用户进行探索.

b. 性能优化. 可视化作用包括调用关系分析、优化策略定制. 调用关系分析中，调用关系分析是定位性能瓶颈的关键[59]，而调用关系往往能被抽象成树或者图结构. 为此，树和图可视化能被用于调用关系分析，帮助挖掘大规模数据中的模式. 优化策略定制中，与调用关系分析类似，优化策略定制同样需要分析调用关系，然而为了定制优化策略，还需要对比策略变更前后对系统造成的影响. 因此，通常会额外设计可视化和交互方式来辅助这一对比过程.

c. 系统维护. 可视化作用包括交互式异常检测、异常结果溯源. 交互式异常检测中，与前 2 个方面不同，由于异常发生频率极低，仅通过基本的数据可视化很难检测. 为此，该方面的方法通常将异常检测模型和可视化进行结合，在节省用户时间的同时提高检测准确率，典型例子有 Xu 等[79]提出的 CloudDet. 异常结果溯源中，除了检测异常外，可视化还允许通过交互式探索的方式帮助用户溯源异常发生的原因，从而定位问题，典型例子有 Xu 等[88]提出的 ViDX.

6　研究机遇与挑战

在超级计算机系统中，可视化技术的应用范围非常广泛，不仅限于提供系统运算过程中的实时监控，协助理解系统内部复杂的运行机制；还能够帮助分析各项性能指标数据，帮助研究人员识别系统中潜在的性能瓶颈，并为进一步的系统性能优化提供重要线索. 针对日常使用过程中的系统维护，可视化技术在进行异常检测与解决潜在问题方面发挥着关键作用，在超级计算机系统中，为各阶段的操作和决策提供可视化分析支持，已然成为了系统运作、优化和维护的重要驱动力. 但是，在超级计算机系统可视化领域，仍然有一些挑战性的问题亟待解决：

(1) 大规模数据原位可视化工具. 海量计算数

据的传输成本惊人，而离线进行可视化所需的机器级别与超级计算机相当. 原位可视化能够在同一台机器上完成计算与可视化 2 个任务，但随着计算规模增大以及数据结构越来越复杂，现有的原位可视化工具在数据处理效率、可视化渲染等方面难以应对挑战；新一代的大规模数据可视化工具需要具备更强大的计算能力，利用并行计算和分布式系统处理大规模数据和复杂数据结构；同时，深度学习和机器学习技术也可以被整合到这些工具中，以确保其拥有足够的灵活性，适应不同类型和规模的计算任务，加深对数据的理解并提高可视化的质量. 因此，扩展原位可视化工具以适应复杂的大规模数据，是未来的一个重要研究方向.

(2) 系统整体性能优化. 目前，针对超级计算机系统的性能优化往往关注于计算核心的优化，而对其他模块，如负载均衡、进程调度等涉及较少. 这种现状可能会导致即使系统的计算核心效率很高，在面对负载激增或者复杂任务时仍然出现效率低下的问题；也可能导致部分节点过载，而其他节点可能处于闲置状态，从而限制了整个系统的性能表现；同时，优化不足的进程调度可能导致资源利用不均衡，影响任务的执行效率和系统的整体稳定性. 尽管这些未被充分重视的模块可能不是主要性能瓶颈，但它们的优化对于整个超级计算机系统的性能提升仍然起到了重要的作用. 因此，将可视化技术应用于这些模块的改进，是未来的一个重要研究方向.

(3) 多维度异常检测方法. 目前，绝大多数针对超级计算机系统的异常检测方法均是对系统运行过程中生成的日志文件进行分析. 然而，日志文件有时会缺乏上下文或遗漏重要信息，仅依赖日志文件可能无法完整地获取系统运行过程中的全部信息. 为了更全面地监测异常，需要结合多维度数据，如性能指标、网络流量数据、运行时上下文信息等，通过可视化的方法实时地进行呈现，从而有效提高异常检测的准确性和及时性. 因此，这种综合性的方法能为超级计算机系统提供更高效率的异常检测，有助于及时检测潜在问题并保障系统稳定运行，是未来的一个重要的研究方向.

(4) 下游任务的可视化需求. 目前，超级计算机系统在高精度数值模拟以及大规模科学计算等方面展现出卓越的性能，被广泛运用于各类下游任务，如量子计算、深度神经网络训练、地震灾害预测等，而每类下游任务所需的模型和数据结构都具有很大的差异，这些领域需要可视化技术直观地呈现复杂数据和不同的模型结构，帮助用户更好地理解特定问题并做出相应的决策. 已经有一些初步工作(如 P4[91], P5[92], P6[93])利用声明式语法提升定制可视化的构建效率. 如何针对不同下游任务更加高效地设计和构建可视化，是一个值得深入研究的方向.

7 结 语

本文首先总结与超级计算机系统可视化相关的研究工作，并将其归纳为状态监控、性能优化和系统维护 3 个方面，然后对不同方面的可视化工作进行分类与说明，着重分析了该领域目前所面临的机遇和挑战. 希望本文有助于读者深入了解可视化技术在超级计算机系统中的重要性以及多方面的应用，并为未来相关的研究工作提供启示.

参考文献(References):

[1] Schneider D. The Exascale Era is Upon Us: the frontier supercomputer may be the first to reach 1 000 000 000 000 000 000 operations per second[J]. IEEE Spectrum, 2022, 59(1): 34-35

[2] Cao Yi, Wang Huawei, Xia Fang, *et al*. Interactive visual analysis engine for high-performance CAE simulations[J]. Journal of Computer-Aided Design & Computer Graphics, 2021, 33(12): 1803-1810(in Chinese)
(曹轶, 王华维, 夏芳, 等. 面向高性能工业仿真的交互可视分析引擎[J]. 计算机辅助设计与图形学学报, 2021, 33(12): 1803-1810)

[3] Lin Jiaqi, Xie Lijun, Ji Houfeng, *et al*. StreamLNet: in-situ compression of flow field combined with streamline visualization[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(7): 1127-1137(in Chinese)
(林佳琦, 解利军, 季候风, 等. StreamLNet: 结合流线可视化的流场原位压缩方法[J]. 计算机辅助设计与图形学学报, 2022, 34(7): 1127-1137)

[4] Robey R, Zamora Y. Parallel and high performance computing[M]. Shelter Island: Manning, 2021

[5] Asaadi H R, Khaldi D, Chapman B. A comparative survey of the HPC and big data paradigms: Analysis and experiments[C]//Proceedings of the IEEE International Conference on Cluster Computing. Los Alamitos: IEEE Computer Society Press, 2016: 423-432

[6] Fan Yilun, Lyu Shenglan, Guo Jie, *et al*. A survey of label placements in visualizations[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(8): 1162-1174(in Chinese)
(范毅伦, 吕胜蓝, 郭杰, 等. 可视化中的标签放置方法综述[J]. 计算机辅助设计与图形学学报, 2023, 35(8): 1162-1174)

[7] Zhu Min, Yao Lin, Wang Xiangkun, *et al*. A survey of visualization approaches for comparative genomics[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(6):

852-868(in Chinese)
(朱敏, 姚林, 王翔坤, 等. 面向比较基因组学的可视化方法综述[J]. 计算机辅助设计与图形学学报, 2022, 34(6): 852-868)

[8] Liu Yuhua, Zhai Ruyu, Zhang Xiang, *et al*. A survey on the visual analytics of knowledge graph[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(1): 23-36(in Chinese)
(刘玉华, 翟如钰, 张翔, 等. 知识图谱可视分析研究综述[J]. 计算机辅助设计与图形学学报, 2023, 35(1): 23-36)

[9] Chen Bin, Zhang Yuhan, Hu Ruizhen. Insight-guided visualization recommendation[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(1): 135-145(in Chinese)
(陈滨, 张钰涵, 胡瑞珍. 信息见解引导的可视化推荐[J]. 计算机辅助设计与图形学学报, 2023, 35(1): 135-145)

[10] Wang Hui, Sun Guodao, Ding Weijie, *et al*. Visual analysis of evolutionary patterns in judicial judgment documents[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(3): 470-481(in Chinese)
(王慧, 孙国道, 丁伟杰, 等. 司法判决文书数据中演化模式的挖掘和可视分析[J]. 计算机辅助设计与图形学学报, 2022, 34(3): 470-481)

[11] Zhu Haiyang, Qian Zhonghao, Yan Fan, *et al*. An interactive visual analysis method for multi-dimensional data deduplication[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(6): 841-851(in Chinese)
(朱海洋, 钱中昊, 严凡, 等. 支持多维度数据去重的交互式可视分析方法[J]. 计算机辅助设计与图形学学报, 2022, 34(6): 841-851)

[12] Isaacs K E, Giménez A, Jusufi I, *et al*. State of the art of performance visualization[OL]. [2023-12-09]. https://www.cs.umd.edu/~bhatele/pubs/pdf/2014/eurovis2014.pdf

[13] Zhang Wei, Wen Zhen, Pan Jiacheng, *et al*. Visualization for computer program: a survey[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(8): 1139-1149(in Chinese)
(张炜, 温圳, 潘嘉铖, 等. 面向计算机程序的可视化综述[J]. 计算机辅助设计与图形学学报, 2023, 35(8): 1139-1149)

[14] Liao Xiangke, Tan Yusong, Lu Yutong, *et al*. On the challenge for supercomputer design in the big data era[J]. Journal of Shanghai University(Natural Science), 2016, 22(1): 3-16(in Chinese)
(廖湘科, 谭郁松, 卢宇彤, 等. 面向大数据应用挑战的超级计算机设计[J]. 上海大学学报(自然科学版), 2016, 22(1): 3-16)

[15] Yang Weikai, Chen Changjian, Zhu Jiangning, *et al*. A survey of visual analytics research for improving training data quality[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(11): 1629-1642(in Chinese)
(杨维铠, 陈长建, 朱江宁, 等. 基于可视分析的训练数据质量提升综述[J]. 计算机辅助设计与图形学学报, 2023, 35(11): 1629-1642)

[16] Yang W K, Liu M C, Wang Z, *et al*. Foundation models meet visualizations: Challenges and opportunities[OL]. [2023-12-09]. https://arxiv.org/abs/2310.05771

[17] Yuan J, Chen C J, Yang W K, *et al*. A survey of visual analytics techniques for machine learning[J]. Computational Visual Media, 2021, 7(1): 3-36

[18] Liu S X, Wang X T, Collins C, *et al*. Bridging text visualization and mining: a task-driven survey[J]. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(7): 2482-2504

[19] Ding Weijie, Liang Ronghua, Sun Guodao, *et al*. Research progress of crime data visualization[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(7): 979-989(in Chinese)
(丁伟杰, 梁荣华, 孙国道, 等. 犯罪数据可视化研究进展[J]. 计算机辅助设计与图形学学报, 2023, 35(7): 979-989)

[20] Wen Zhen, Li Peng, Wang Xumeng, *et al*. Mixed reality based visual analytics[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(6): 821-829(in Chinese)
(温圳, 李鹏, 王叙萌, 等. 基于混合现实的可视分析综述[J]. 计算机辅助设计与图形学学报, 2022, 34(6): 821-829)

[21] Zhou Zhiguang, Wang Saqing, Liu Yuhua, *et al*. A survey on visual analysis of massive open online courses (MOOC) data[J]. Journal of Computer-Aided Design & Computer Graphics, 2022, 34(6): 830-840(in Chinese)
(周志光, 王飒清, 刘玉华, 等. 大型开放式在线课程(MOOC)数据可视分析综述[J]. 计算机辅助设计与图形学学报, 2022, 34(6): 830-840)

[22] Fu H H, Liao J F, Yang J Z, *et al*. The Sunway TaihuLight supercomputer: system and applications[J]. Science China Information Sciences, 2016, 59(7): Article No.072001

[23] Muelder C, Gygi F, Ma K L. Visual analysis of inter-process communication for large-scale parallel computing[J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15(6): 1129-1136

[24] Chen C J, Chen J S, Yang W K, *et al*. Enhancing single-frame supervision for better temporal action localization[OL]. [2023-12-09]. https://arxiv.org/abs/2312.05178

[25] Cao K L, Liu M C, Su H, *et al*. Analyzing the noise robustness of deep neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(7): 3289-3304

[26] Karran B, Trümper J, Döllner J. SYNCTRACE: visual thread-interplay analysis[C] //Proceedings of the First IEEE Working Conference on Software Visualization. Los Alamitos: IEEE Computer Society Press, 2013: 1-10

[27] Sigovan C, Muelder C W, Ma K L. Visualizing large-scale parallel communication traces using a particle animation technique[J]. Computer Graphics Forum, 2013, 32(3pt2): 141-150

[28] Sakin S A, Bigelow A, Tohid R, *et al*. Traveler: navigating task parallel traces for performance analysis[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(1): 788-797

[29] Wang Y, Agrawal G, Bicer T, *et al*. Smart: a mapReduce-like framework for in-situ scientific analytics[C] //Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. Los Alamitos: IEEE Computer Society Press, 2015: 1-12

[30] Kageyama A, Yamada T. An approach to exascale visualization: Interactive viewing of in-situ visualization[J]. Computer Physics Communications, 2014, 185(1): 79-85

[31] Ahrens J, Jourdain S, O'Leary P, *et al*. An image-based approach to extreme scale in situ visualization and analysis[C] //Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. Los Alamitos: IEEE Computer Society Press, 2014: 424-434

[32] Dutta S, Chen C M, Heinlein G, *et al*. In situ distribution guided analysis and visualization of transonic jet engine simulations[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 811-820

[33] Haimes R. pV3-a distributed system for large-scale unsteady CFD visualization[OL]. [2023-12-09]. https://doi.org/10.2514/6.1994-321

[34] Childs H, Brugger E S, Whitlock B, *et al*. VisIt: an end-user tool for visualizing and analyzing very large data[OL]. [2023-12-09]. https://www.osti.gov/biblio/1048735

[35] Ayachit U. The paraView guide: a parallel visualization application[M]. Clifton Park: Kitware, Inc., 2015

[36] Lofstead J F, Klasky S, Schwan K, *et al*. Flexible IO and integration for scientific codes through the adaptable IO system (ADIOS)[C] //Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments. New York: ACM Press, 2008: 15-24

[37] Vishwanath V, Hereld M, Papka M E. Toward simulation-time data analysis and I/O acceleration on leadership-class systems[C] //Proceedings of the IEEE Symposium on Large Data Analysis and Visualization. Los Alamitos: IEEE Computer Society Press, 2011

[38] Woodring J, Petersen M, Schmeißer A, *et al*. In situ eddy analysis in a high-resolution ocean climate model[J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(1): 857-866

[39] Han X Y, Yu X M, Li G, *et al*. Narrative *in situ* visual analysis for large-scale ocean eddy evolution[J]. IEEE Computer Graphics and Applications, 2022, 42(3): 65-73

[40] Hadlak S, Schulz H J, Schumann H. In situ exploration of large dynamic networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2334-2343

[41] Goodall J R, Ragan E D, Steed C A, *et al*. Situ: identifying and explaining suspicious behavior in networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(1): 204-214

[42] Rübel O, Loring B, Vay J L, *et al*. Warpiv: in situ visualization and analysis of ion accelerator simulations[J]. IEEE Computer Graphics and Applications, 2016, 36(3): 22-35

[43] Ma K L. In situ visualization at extreme scale: challenges and opportunities[J]. IEEE Computer Graphics and Applications, 2009, 29(6): 14-19

[44] Shan Guihua, Tian Dong, Xie Maojin, *et al*. In-situ visualization for peta-scale scientific computation[J]. Journal of Computer-Aided Design & Computer Graphics, 2013, 25(3): 286-293(in Chinese)
(单桂华, 田东, 谢茂金, 等. 千万亿次科学计算的原位可视化[J]. 计算机辅助设计与图形学学报, 2013, 25(3): 286-293)

[45] Bauer A C, Abbasi H, Ahrens J, *et al*. *In situ* methods, infrastructures, and applications on high performance computing platforms[J]. Computer Graphics Forum, 2016, 35(3): 577-597

[46] Bruder V, Larsen M, Ertl T, *et al*. A hybrid in situ approach for cost efficient image database generation[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(9): 3788-3798

[47] Woodring J, Ahrens J, Figg J, *et al*. In-situ sampling of a large-scale particle simulation for interactive visualization and analysis[J]. Computer Graphics Forum, 2011, 30(3): 1151-1160

[48] He W B, Wang J P, Guo H Q, *et al*. InSituNet: deep image synthesis for parameter space exploration of ensemble simulations[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(1): 23-33

[49] Li G, Wang J P, Wang Y, *et al*. An in-situ visual analytics framework for deep neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2023: 1-17

[50] Landge A G, Levine J A, Bhatele A, *et al*. Visualizing network traffic to understand the performance of massively parallel simulations[J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(12): 2467-2476

[51] Chen C J, Guo Y K, Tian F Y, *et al*. A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision[J]. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(1): 76-86

[52] Zhou Y X, Yang W K, Chen J S, *et al*. Cluster-aware grid layout[J]. IEEE Transactions on Visualization and Computer Graphics, 2024, 30(1): 240-250

[53] Chen C J, Yuan J, Lu Y F, *et al*. OoDAnalyzer: interactive analysis of out-of-distribution samples[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(7): 3335-3349

[54] Chen C J, Wu J, Wang X H, *et al*. Towards better caption supervision for object detection[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(4): 1941-1954

[55] Yuan J, Liu M C, Tian F Y, *et al*. Visual analysis of neural architecture spaces for summarizing design principles[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(1): 288-298

[56] Shilpika F, Fujiwara T, Sakamoto N, *et al*. A visual analytics approach for hardware system monitoring with streaming functional data analysis[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(6): 2338-2349

[57] Osmari D K, Vo H T, Silva C T, *et al*. Visualization and analysis of parallel dataflow execution with smart traces[C] // Proceedings of the 27th SIBGRAPI Conference on Graphics, Patterns and Images. Los Alamitos: IEEE Computer Society Press, 2014: 165-172

[58] Isaacs K E, Bremer P T, Jusufi I, *et al*. Combing the communication hairball: Visualizing parallel execution traces using logical time[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(12): 2349-2358

[59] Nguyen H T, Bhatele A, Jain N, *et al*. Visualizing hierarchical performance profiles of parallel codes using callflow[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(4): 2455-2468

[60] Yang W K, Wang X T, Lu J, *et al*. Interactive steering of hierarchical clustering[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(10): 3953-3967

[61] Liu S X, Xiao J N, Liu J L, *et al*. Visual diagnosis of tree boosting methods[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(1): 163-173

[62] Li Z, Wang X T, Yang W K, *et al*. A unified understanding of deep NLP models for text classification[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(12): 4980-4994

[63] Nguyen H T, Wei L, Bhatele A, *et al*. VIPACT: a visualization interface for analyzing calling context trees[C] //Proceedings of the Third Workshop on Visual Performance Analysis. Los Alamitos: IEEE Computer Society Press, 2016: 25-28

[64] Archambault D, Munzner T, Auber D. TopoLayout: multilevel graph layout by topological features[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(2): 305-317

[65] Gansner E R, Koren Y, North S C. Topological fisheye views

for visualizing large graphs[J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11(4): 457-468

[66] Kesavan S P, Bhatia H, Bhatele A, *et al*. Scalable comparative visualization of ensembles of call graphs[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(3): 1691-1704

[67] Shende S S, Malony A D. The TAU parallel performance system[J]. The International Journal of High Performance Computing Applications, 2006, 20(2): 287-311

[68] Adhianto L, Banerjee S, Fagan M, *et al*. HPCTOOLKIT: tools for performance analysis of optimized parallel programs[J]. Concurrency and Computation: Practice and Experience, 2010, 22(6): 685-701

[69] Kesavan S P, Fujiwara T, Li J K, *et al*. A visual analytics framework for reviewing streaming performance data[C] //Proceedings of the Pacific Visualization Symposium. Los Alamitos: IEEE Computer Society Press, 2020: 206-215

[70] Jo J, Kim W, Yoo S, *et al*. SwiftTuna: responsive and incremental visual exploration of large-scale multidimensional data[C] //Proceedings of the Pacific Visualization Symposium. Los Alamitos: IEEE Computer Society Press, 2017: 131-140

[71] Li J K, Fujiwara T, Kesavan S P, *et al*. A visual analytics framework for analyzing parallel and distributed computing applications[C] //Proceedings of the Visualization in Data Science. Los Alamitos: IEEE Computer Society Press, 2019: 1-9

[72] Muelder C, Zhu B, Chen W, *et al*. Visual analysis of cloud computing performance using behavioral lines[J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(6): 1694-1704

[73] Zhang J, Yang C H, Li Y D, *et al*. LBVis: interactive dynamic load balancing visualization for parallel particle tracing[C] //Proceedings of the Pacific Visualization Symposium. Los Alamitos: IEEE Computer Society Press, 2020: 91-95

[74] Liu M C, Shi J X, Li Z, *et al*. Towards better analysis of deep convolutional neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 91-100

[75] Liu M C, Shi J X, Cao K L, *et al*. Analyzing the training processes of deep generative models[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(1): 77-87

[76] Giménez A, Gamblin T, Jusufi I, *et al*. MemAxes: visualization and analytics for characterizing complex memory performance behaviors[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(7): 2180-2193

[77] Fang Xiaoyu, Cao Chenhan, Xia Bin. Attention based log-level anomaly detection algorithm for large-scale system logs[J]. Journal of Nanjing University (Natural Science), 2021, 57(5): 785-792(in Chinese)
(房笑宇, 曹陈涵, 夏彬. 基于注意力机制的大规模系统日志异常检测方法[J]. 南京大学学报(自然科学), 2021, 57(5): 785-792)

[78] Wang Xiaodong, Zhao Yining, Xiao Haili, *et al*. Research on anomaly detection system of online multi-node log flow[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(11): 1828-1837(in Chinese)
(王晓东, 赵一宁, 肖海力, 等. 线上多节点日志流量异常检测系统的研究[J]. 计算机科学与探索, 2020, 14(11): 1828-1837)

[79] Xu K, Wang Y, Yang L N, *et al*. CloudDet: interactive visual analysis of anomalous performances in cloud computing systems[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(1): 1107-1117

[80] Xie C, Xu W, Mueller K. A visual analytics framework for the detection of anomalous call stack trees in high performance computing applications[J]. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(1): 215-224

[81] Yang W K, Ye X, Zhang X X, *et al*. Diagnosing ensemble few-shot classifiers[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(9): 3292-3306

[82] Chen C J, Wang Z W, Wu J, *et al*. Interactive graph construction for graph-based semi-supervised learning[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(9): 3701-3716

[83] Yuan J, Xiang S X, Xia J Z, *et al*. Evaluation of sampling methods for scatterplots[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 1720-1730

[84] Yang W K, Li Z, Liu M C, *et al*. Diagnosing concept drift with visual analytics[C] //Proceedings of the IEEE Conference on Visual Analytics Science and Technology. Los Alamitos: IEEE Computer Society Press, 2020: 12-23

[85] Xiang S X, Ye X, Xia J Z, *et al*. Interactive correction of mislabeled training data[C] //Proceedings of the IEEE Conference on Visual Analytics Science and Technology. Los Alamitos: IEEE Computer Society Press, 2019: 57-68

[86] Liu S X, Chen C J, Lu Y F, *et al*. An interactive method to improve crowdsourced annotations[J]. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(1): 235-245

[87] Xia Jiazhi, Li Jie, Chen Siming, *et al*. A survey on interdisciplinary research of visualization and artificial intelligence[J]. Scientia Sinica Informationis, 2021, 51(11): 1777-1801(in Chinese)
(夏佳志, 李杰, 陈思明, 等. 可视化与人工智能交叉研究综述[J]. 中国科学: 信息科学, 2021, 51(11): 1777-1801)

[88] Xu P P, Mei H H, Ren L, *et al*. ViDX: visual diagnostics of assembly line performance in smart factories[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 291-300

[89] Wei Y T, Wang Z Y, Wang Z W, *et al*. Visual diagnostics of parallel performance in training large-scale DNN models[J]. IEEE Transactions on Visualization and Computer Graphics, 2023: 1-14

[90] Yi J S, Kang Y A, Stasko J, *et al*. Toward a deeper understanding of the role of interaction in information visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6): 1224-1231

[91] Li J K, Ma K L. P4: portable parallel processing pipelines for interactive information visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(3): 1548-1561

[92] Li J K, Ma K L. P5: portable progressive parallel processing pipelines for interactive data analysis and visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(1): 1151-1160

[93] Li J K, Ma K L. P6: A declarative language for integrating machine learning in visual analytics[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 380-389