Practical Evaluation Methods for Scientific Instruments

Kate Keahey
The University of Chicago
Chicago, IL, USA
Argonne National Laboratory
Lemont, IL, USA
keahey@uchicago.edu

Paul Marshall
The University of Chicago
Chicago, IL, USA
marshalp@uchicago.edu

Mark Powers
The University of Chicago
Chicago, IL, USA
markpowers@uchicago.edu

Marc Richardson
The University of Chicago
Chicago, IL, USA
mtrichardson@uchicago.edu

Michael Sherman
The University of Chicago
Chicago, IL, USA
msherman@uchicago.edu

Dan Stanzione
The University of Texas at Austin
Austin, Texas, USA
Texas Advanced Computing Center
Austin, Texas, USA
dan@tacc.utexas.edu

Abstract

Evaluating experimental cyberinfrastructure requires practical methodologies that balance assessment objectives with feasible data collection. This paper proposes such evaluation methods and demonstrates their value by analyzing usage and impacts based on 10 years of operation of the Chameleon project, an NSF-funded testbed for computer science research and education. We present three methodological contributions: (1) user engagement metrics that differentiate between login activity, project membership, and actual resource usage; (2) usage measurements for interactive systems, showing the impact of lease management on project usage; and (3) a multi-source bibliometric approach combining algorithmic discovery via Google Scholar, Scopus, and Semantic Scholar with user reporting to establish verifiable publication counts. Our methodology identified 1,200 publications produced by the Chameleon community by the end of 2025, with 54% discovered through citation, 31% through acknowledgment, and 15% through user reporting alone. We provide code, metric definitions, and data collection recommendations to enable reproducible evaluations and meaningful cross-project comparisons, helping differentiate between proof-of-concept systems and scientific instruments serving broad research communities.

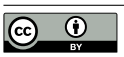
CCS Concepts

• **General and reference → Metrics**; **Measurement**; **Evaluation**;
• **Computer systems organization → Cloud computing**.

Keywords

cyberinfrastructure evaluation, multi-source bibliometrics, infrastructure impact

ACM Reference Format:
Kate Keahey, Paul Marshall, Mark Powers, Marc Richardson, Michael Sherman, and Dan Stanzione. 2026. Practical Evaluation Methods for Scientific Instruments. In *Practice and Experience in Advanced Research Computing (PEARC '26), July 26–30, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3785462.3815804



PEARC '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2377-3/26/07
https://doi.org/10.1145/3785462.3815804

1 Introduction

Building innovative or experimental infrastructure systems often carries significant risk. Since the system presents a new solution, it is sometimes difficult to set expectations for its success whether in terms of technology innovation, development of the community the system should serve, level of use it is fair to expect, or how much research impact the system should produce. In some cases, it is difficult to state up front what metrics may be the best measure of success and what metrics may be obtainable in practice. At the same time, the question of how to evaluate new and experimental systems is of particular importance now, as academic computing is increasingly augmenting traditional batch computing system presentation by exploring new models for both data-centric and interactive systems, and seeking the best methods to support AI workflows.

Measuring experimental systems is of value in multiple ways. First, keeping tabs on the system evolution and asking questions about its usage can help understand user community dynamics and keep pace with the development of new scientific interests to guide project evolution – in other words, it is essential to improving the system. Second, quantifying the system adoption or its results in terms of scientific impact allows us to capture how well the system meets its expectations and thereby define its success. Lastly, differentiating between outcomes a system may produce allows us to classify different experimental systems so that we can better articulate and set expectations for their value added whether in the breadth or depth of scientific appeal (community-scale impact) or in the innovative nature of the system itself (system-scale impact).

In this paper, we propose practical evaluation methods for scientific instruments implemented in cyberinfrastructure and illustrate and discuss their usefulness by evaluating Chameleon [16], an NSF-funded experimental infrastructure for computer science research and education. Chameleon is a slice-based resource that allows users to allocate a resource *slice* [22] for direct, interactive use. The system is configured as a heavily customized OpenStack [20] cloud to meet the requirements of our use case. Users provision resource slices

through advance reservations (including immediate/on-demand reservations) of time-bounded leases that they can then reconfigure via bare metal, virtual machine (VM), or container deployment (only the first two will be discussed here); this makes the usage recommendations we provide applicable to a broad range of interactive systems. Lastly, as an infrastructure that has been in operation for over 10 years, Chameleon provides an interesting case study of research impact and a data point in terms of expectations for this type of instrument.

The practices we propose are simple and intuitive, but we have seen them applied inconsistently across many systems, making cross-system comparison difficult. Some of the proposed evaluation methods finesse accepted practices – for example, we present user counts, but we also consider users in terms of how they engage with the system. We share the metric definitions, the infrastructure evaluation questions, and the code, where appropriate, that can be used to replicate similar evaluation for other systems, and we make specific recommendations for data collection to facilitate setting up new experimental infrastructure projects. We also provide recommendations for metric collection for new or experimental resources for interactive usage.

2 Related Work

Prior work in this domain can be organized into three main categories: comprehensive system evaluations, bibliometric method comparisons, and return on investment analyses.

On system evaluations, von Laszewski et al. [29] developed a comprehensive framework for evaluating XSEDE's scientific impact over the project period using bibliometric approaches and leveraging multiple data sources (e.g., ISI Web of Science and Microsoft Academic Graph) to calculate Field-Weighted Citation Impact (FWCI) and peer comparisons. Similarly, Hart [15] analyzed resource usage patterns across TeraGrid and XSEDE, although focusing on examining operations metrics, e.g., job submission patterns, user retention, and community demographics, rather than impact metrics. Lastly, Bayly et al. [2] presented tools for institutional-level publication impact exploration, using OpenAlex to track contributions over six years and providing interactive visualizations for stakeholder engagement.

Understanding the coverage and limitations of different bibliometric databases is essential for comprehensive impact assessment. Yang and Meho [31] provided an early comparison of Google Scholar, Scopus, and Web of Science, demonstrating significant differences in citation coverage across these sources and highlighting the value of using multiple databases for complete citation analysis. Martín-Martín et al. [19] similarly conducted a large-scale multidisciplinary comparison of six citation sources (Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI), also finding that Google Scholar captured significantly more citations than others, with substantial subject-specific variations. Lastly, Stewart et al. [25–27] and Snapp-Childs et al. [24] examined financial effectiveness and return on investment for XSEDE, demonstrating measurable value exceeding federal funding costs and providing frameworks for evaluating both direct financial returns and broader research impact. Their work established methodologies for calculating ROI that account for multiple types of value delivered by cyberinfrastructure beyond simple cost-benefit ratios.

3 Measuring Users and Projects

An assessment of user/project numbers, as well as their variety, give early and direct feedback on the activity on a project. Access to Chameleon is managed on the user/project system similarly to ACCESS [3] (the project migrated to federated identity only in 2020 [1]); any user with valid credentials can log into the system using federated identity [28], but they can actually use the system resources only once they get added to a project with an allocation. To request an allocation, users are first certified for Principal Investigator (PI) status and can then propose a project to which, if approved, they can add members. A Chameleon allocation represents a fixed commitment of 20,000 of service units (SUs) for 6 months; once an allocation expires the PI has the option to extend or recharge it.

User and project information is obtained through two main sources: the Keycloak database which is the point of authentication to the system, and the portal database synchronized with Keycloak, which maintains information about PI eligibility, project memberships, and user profiles. We modified the Keycloak database to track if users were added to a project and to persist login events so that we can tell how many unique users logged into the system over a period of time (active users). To track whether a user actually used the resources of the system, we use OpenStack events.

The overall cumulative unique user and project number as of the end of 2025 is 13,463 and 1,351, respectively; both have been rising steadily over time – with the number of new additions also rising year on year, e.g., from roughly 500 new users in 2015 to roughly 1,000 in 2020, and exceeding roughly 2,000 new users for the last two years of the project (the growth in new user numbers is leveling which may be a sign of system saturation).

Understanding how many users a system supports on an ongoing basis can be challenging as there are many definitions of what constitutes a user of the system. The definition that we typically use – because it is the one that everybody else uses, which allows for meaningful comparisons across projects – is the number of users who logged into the system (definition A). However, not all the users who log into the system end up actually using it. For this reason, similarly to XDMoD [21], we also track the number of users who were added to a project (definition B). Lastly, it is also interesting to see how many users actually allocate resources on the system via OpenStack calls (definition C).

Figure 1 compares the number of active users falling under those different definitions over the last two years since we started tracking active user information. First, we see that on average about 700 users log into the system each month during the academic year, with a sharp drop to about 400-500 users during the summer and winter holidays. Comparing the users active in one month to the ones active in the month immediately following, we find that there is roughly 50% of overlap, which shows a good balance between retention and new user activity. Next, we see that every month roughly 10% of users who log into the system never get added to a project, and roughly 20% of the users who log into the system do not ever make an OpenStack call. These differences will grow

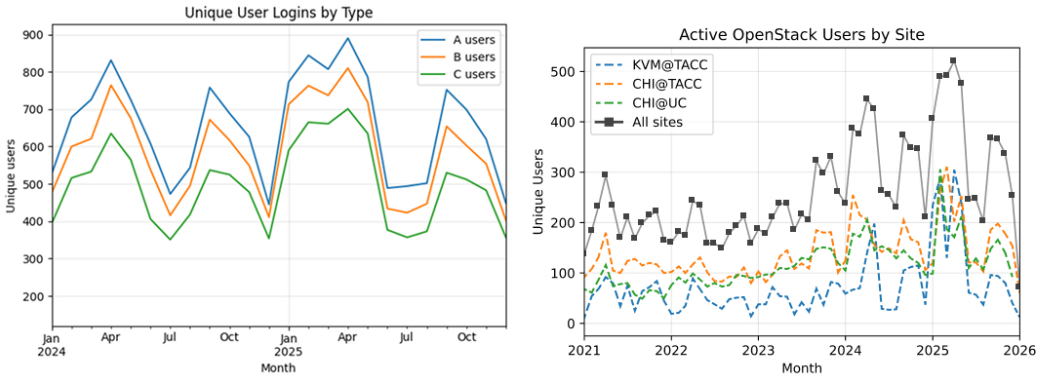


Figure 1: (left) Comparison over the last two years of the number of users who logged into Chameleon each month (A); the subset of A who were added to a Chameleon project at any time (B); the subset of B who made OpenStack calls at any time (C). (right) Numbers of users using OpenStack commands (C) over the last 5 years broken down by the core bare metal sites and virtualized component.

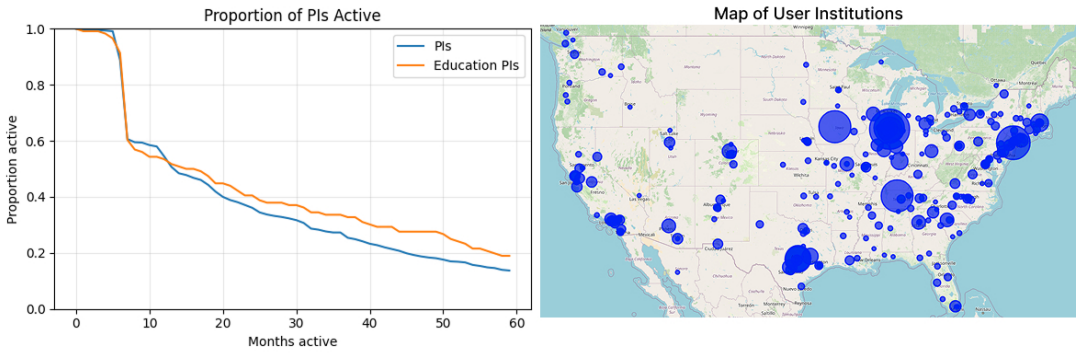


Figure 2: (left) The proportion of PIs using the system for a given number of months. (right) A map of Chameleon user institutions. The size of the bubble is proportional to the number of users at that institution.

larger over larger periods of time because few of the users who never get added to a project return: we found that, measured over the two-year period, roughly 25% of users who logged into the system have never been added to a project. This means that while portal logins are a good measure of interest in the system, they are not a good measure of actual activity on the system.

To find out who the users in these respective categories are, we examined their logins for the year 2025. We found that about 80% of users in group A – but not B – belong to unique institutions, which suggests that users logged in, found that they could not use the system without an allocation, and were unwilling or unable to obtain one. The remaining 20% consist of logins from institutions that heavily use Chameleon for education which suggests that they may have been students who decided to drop a class. Over a quarter of users in group B – but not C – are PIs (we found that only about 40% of PIs overall actually use OpenStack). Based on interacting with our users over the lifetime of the project, we suspect that the remainder use Chameleon through resources provisioned by others; for example, students using Chameleon for a group project in a class may organize their work so that one student configures resources on Chameleon while others work on other parts of the project.

The right graph in Figure 1 shows numbers of users who make OpenStack calls, i.e., provision and configure hardware resources on the system broken down into activity on the two core bare metal sites (UChicago and TACC) as well as the virtualized (KVM) partition (definition C). We see a repeat of the same academic year variability pattern as on the left side, but we also see a significant uptick of activity as the project goes on, especially in the last two years, consistent with the increase of annual growth in user numbers.

PIs are a special category of users as typically they are the ones who decide whether to use Chameleon for their project; their continued use of the system can thus be seen as a measure of user satisfaction. At the end of 2025, we have supported 800 PIs, 730 in research projects, and 117 in educational projects, with an overlap of 47 involved in both. Since every Chameleon project ends in 6 months, we would expect that any PI with unsatisfactory experience would not continue their association with the system beyond that – though of course it is also possible to have a good experience

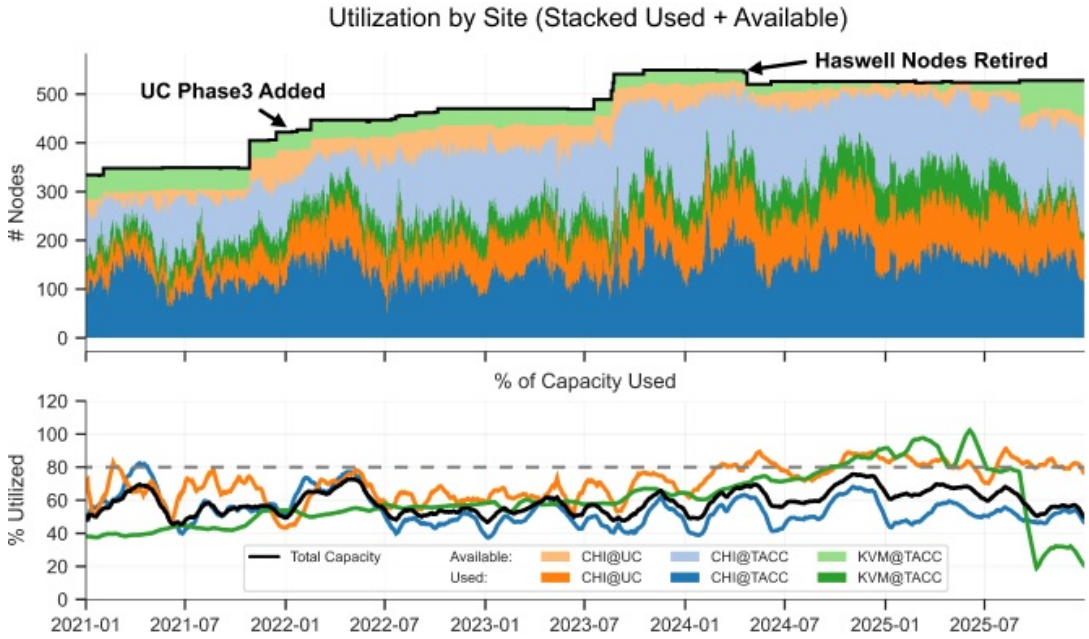


Figure 3: Usage and utilization over the last 5 years of the project; Top graph uses bright colors to show TACC and UC bare metal and KVM usage, while muted colors show availability.

and simply finish the project in 6 months. The left side of Figure 2 shows what proportion of PIs was using the system for what length of time throughout the project. We see that almost all PIs choose to continue their use of the system beyond their first experience and that 15-20% of PIs chose to continue using the system for as long as 5 years. Lastly, the right side of Figure 2 provides a graphical illustration of the reach of the project. Unsurprisingly, user number concentrations are dominated by institutions that use Chameleon for teaching (top 6 are: Iowa State, Vanderbilt, NYU, IIT, Austin Community College, and DePaul).

4 Measuring Cloud Usage

Systems supporting interactive usage typically operate by making a *slice* of a resource [22] available for temporary ownership, usually via either a container (e.g., NRP [30]), VM (e.g., commercial clouds), bare metal – or all of the above, as is the case with Chameleon. We propose that usage of such systems should be measured in terms of bare metal capacity, normalizing resource slices to the nearest node, as such representation gives the best idea of resource saturation. The usage expectations for such slice-based, interactive systems are different from typical batch-operated systems in that resources are allocated "on demand" (i.e., when the user needs the resource which may or may not occur when the resource is available) rather than "on availability" of a resource, allowing to maximize resource usage [17]. In other words, the goal of interactive systems is to make resources available as much as possible *at the user's convenience*, an objective that is hard to meet if a resource is 100% utilized. A rule of thumb that is sometimes used for on-demand systems is that 50% utilization means that a user who comes to the system expecting to provision resources on-demand has a 50% chance of finding resources. Advance reservations transform demand peaks into future use which means that in a system that – like Chameleon – supports advance reservation that threshold is higher. At the same time, because of its mission, Chameleon supports a wide range of different resource types which means that even with low overall utilization a user still may not be able to find a specific resource.

We use the total capacity representation obtained from the OpenStack Blazar database and databases managed by OpenStack services (Nova and Ironic) as a source of tracking information for resource usage. A database snapshot is taken every day and modified in the following ways. First, the information is filtered to include "usable capacity" only, defined as any node that is not broken and marked as available for reservation (i.e., not in maintenance or otherwise taken out of service). For resources that support advance reservations (bare metal and VMs after 08/25), a resource is considered "used" if it has been marked as reserved in OpenStack Blazar; in [16] we point out that to be useful the node also has to be reconfigured and explain how we ensure that the user either does that or gives up the node. For resources available on-demand only (i.e., VMs prior to 08/25), "used" corresponds to the deployment of a VM on the resource.

The top graph in Figure 3 shows how the total usage measured in terms of nodes evolved over the last 5 years of the project, flagging the most significant additions and retirements, while the bottom graph in the figure shows the corresponding resource utilization. We are only able to show the last 5 years due to gaps in data availability: the historical information about total capacity has not been consistently preserved so we only have reliable data going back to 2022 (the 2021 data has been reconstructed and should be considered approximate). Data prior to 2021 has been discarded since early

OpenStack upgrades have introduced semantic discontinuity due to changes in how data propagates through the system, and system reconfiguration in 2018 caused some data loss. Since the system combines bare metal resources and VMs we normalized each VM's vCPUs to the host's total vCPUs (e.g., a 4-core VM on a 12-core host counts as one third of that host) in the case of Haswell nodes and made a similar "by GPU" calculation in the case of instances on GPU nodes.

We see an overall growth in usage and utilization at both UChicago and TACC (especially the KVM site) frequently approaching and at times exceeding 80%, especially in the last two years of the project, a finding correlating with user activity presented in Figure 1 in Section 3. This is uncomfortably high for a system meant to support interactive usage; from interacting with our users we know that some are being turned off of the system because it was not possible to find available resources for their project or class. Capturing exact user metrics of this is hard: users typically verify availability through the availability calendar so that our main way of capturing data would be a voluntary user survey which is an inexact measure.

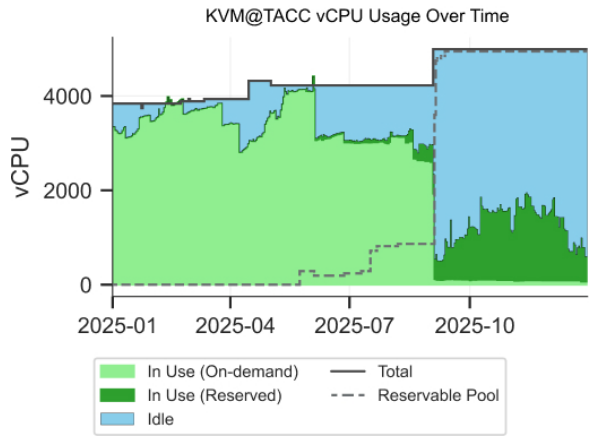


Figure 4: KVM@TACC Chameleon site usage over the last five years.

To relieve the pressure on bare metal nodes with GPUs – increasingly popular over the last several years – in 2025 we introduced virtualized GPU flavors on 6 nodes with H100 GPUs. This necessitated a system change: until 2025, in contrast to bare metal, KVM instances were operated as on-demand (instead of by advance reservation) and unbounded (instead of limited time only). This would not work for GPU instances: if they were unbounded nobody would ever give one up; and if they were on-demand only, nobody would ever be able to get hold of one. For this reason, before virtualized GPU flavors were released, we brought the configuration of our KVM site into compliance with bare metal by transitioning capacity from on demand to reservable/bounded use. The right graph in Figure 4 zooms in on KVM site usage and illustrates this shift with a dashed vertical line. One important feature illustrated by this Figure is the unbounded "runaway VM" problem: as part of the transition, many long-running, unbounded VM instances were stopped and archived – but the system did not immediately go back to the same levels of utilization, which means that many of them were not needed. This shows that quotas and allocation limits are an insufficient incentive to motivate users to terminate VM instances that are no longer needed. At the same time, operators have no insight into whether a VM is still needed or not, and cannot terminate VMs on active projects, a problem that bounded leases solve.

5 Measuring Science Impact

Ultimately the measure of the impact of scientific instruments is how much science they support. This is traditionally measured by bibliometrics [4], i.e., the research products, such as papers, datasets, or reproducible artifacts that were produced using a system, and sometimes the impact of those research products themselves. In this section, we describe the methods we used to collect this information for Chameleon.

The first challenge of such collection is *attribution*: how can we reliably know which publications were produced using the system? A method that is often used is to assume that, if a system serves a community, any publication produced by this community should count as, e.g., in [2]; this leads to an over-count as the community members may have produced publications that did not use the system. We posit that the only acceptable proof that an instrument was used in advancing research presented in a publication is the author's formal statement that this was the case. In the best case, this statement is explicitly expressed as part of the publication text. However, users often forget this acknowledgment; in these cases, a formal user attestation is sufficient.

The attribution challenge is related to that of *identification*: how do we find the publications that were produced using the system? In the initial years of the project, the primary method was through self-reporting when users sought a renewal of their allocation. However, publications are often produced after the allocation is complete, when there are few incentives for users to report them to the project. To get around this issue, we developed a web page allowing users to report publications at any time [5]. This led to over-reporting from users who, e.g., sometimes included, in error, publications that did not use the system. We responded in two ways: (1) we modified the web page to ask users for an explicit attestation that the publication was produced using Chameleon, and (2) revised all 139 self-reported publications since the beginning of the project asking for an explicit attestation. This resulted in 86 confirmations, 33 denials, 20 non-response which we do not include in our count.

As the project progressed and we ourselves published papers describing its contributions, our user community started referencing those papers. These references are the best form of acknowledgment as they both explicitly *attribute* the work to the use of the resource and make *identification* easier through using scholarly databases and search engines, specifically Scopus [12], Semantic Scholar [18], and Google Scholar [13] though they raise yet other challenges. First, some publications reference Chameleon for its contribution rather than to indicate that a publication used the system (roughly 6% of considered publications); should those publications be *included* in the count? What about publications written by testbed operators on the operation of the testbed? We decided not to include those either. To implement this inclusion policy every publication is manually reviewed and potentially rejected if it

does not meet our criteria. Another challenge is that of *determining duplicates* of publications. Part of the problem is simply that many conference names have multiple synonyms, and part is deciding whether a publication with a similar title and authors is a journal version of a conference paper (distinct contribution) or a duplicate. We address this in two ways; when importing a publication from a source we check its source-specific identifier against already imported publications; second, we use Gestalt pattern matching [23] to identify similar publications for manual review.

Using these methods, we identified 1,200 user publications produced since the beginning of the project to the end of 2025; all of them are published on our web page [6]. While this certainly constitutes an under-count we are confident that it is a strong lower bound, i.e., all publications have been verified via an explicit user statement as using the system.

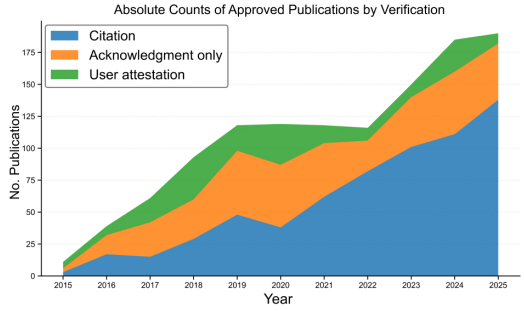


Figure 5: Absolute publication counts by year, broken down by attribution method.

The first set of questions centers on how the number of publications supported by the system is growing, how users acknowledge Chameleon, and how this practice changed over the last 10 years. In Figure 5, we see that the number of reported publications has been growing annually, with the exception of 3 years (2020 – 2022) following the COVID-19 pandemic. The graph also shows method of attribution broken down into the categories of citation, acknowledgment in text (but no citation), and user attestation only (but no explicit acknowledgment or citation). Proportionally, user attestation was more popular for acknowledging Chameleon in early years of the project, roughly averaging 27% in the first three years of the project, declining in share to 8% (averaged) in the last three years of the project. Similarly, we see a growing proportion of publications referencing Chameleon by citation; this is correlated with the first Chameleon papers published in 2019 and 2020. Overall, citation is the most common acknowledgment method in aggregate: 54% of publications are identified through citation, versus 31% for acknowledgment only, and 15% for user attestation.

Next, we ask how useful – versus how effort-intensive – publication identification is through a specific source, an analysis similar to the one carried out in [19, 31]. In Figure 6, we look at the absolute numbers of publications identified by each source and the extent of overlap between our sources, noting that 744 publications (or 62%) have more than one. Google Scholar identified not only the most publications overall (1,079 or 90%), but also the most unique publications (348), including conference papers (141), PhD/MS dissertations (128), and journal articles (34) – high-quality publications that we would not want to miss. At the same time, Google Scholar is also the most effort-intensive method of data collection, as it does not provide programmable interfaces, forbids automated scraping in its terms and conditions, and has made clear through legal action [14] that it will not condone such misuse. For this reason, in order to produce a list of candidate publications, it is necessary to first run manual searches, and then manually add each of the results to a list, which then can be exported – a process that can take hours for large numbers of publications as compared to the instantaneous response from an API-based system.

User reporting identified the second most unique publications (96), including conference papers (54) and journal articles (18), and is irreplicable since some of the publications are identifiable only through user attestation. Unfortunately it is also the second most effort-intensive method, requiring the development of a reliable attestation mechanism and in some cases interaction with users. Otherwise, Scopus had 6 unique identifications, Semantic Scholar had 4, and Science Direct and OpenAlex had 1 each. The latter two did not yield much in either absolute numbers or unique results which suggests that they could be omitted – but are so easy to operate that it is a moot point. We note however that while Semantic Scholar, Scopus, and user reporting each identified ~600 publications, ~50% of the total, together they account for ~70% of publications. Therefore, while in our case the overall best identification method is Google Scholar and user reporting (unique contributions from other aggregators are negligible), Semantic Scholar, Scopus, and user reporting yield ~70% of the result but with much less effort.

Since not all publications are considered to have the same impact, e.g., peer-reviewed journal articles and conference papers are generally considered more impactful than preprints, we also break down the publications by category in Figure 7. We find that most of our publications are published in conferences (~55%); followed by journal articles (~20%), theses/dissertations (~12%), self-published artifacts (e.g., arXiv [10], SSRN [11], or personal website) (~9%); or "other" which include unique artifacts such as datasets, patents, software, presentations, and book chapters (~3%). We also examined how different sources contribute to identifying publications across categories. Of note is that while conference papers and journal articles are represented relatively evenly across all sources (~17-35% in each), by far the most of the dissertations and theses (~90%) are found through Google Scholar only while other sources contribute only minimally to this category – an additional reason to use Google Scholar. Lastly, we also looked at the number of unique venues (defined by conference name, organizer, and year) where these publications appeared and found that publications are published in a wide array of venues with no significant concentrations or patterns, but with the most popular being Supercomputing, Future Generation Computer Systems, HPDC, SIGMOD, INFOCOM, EMNLP, CCGrid, IPDPS, and EuroSys.

6 Discussion and Recommendations

Characterizing the evolution and impact of a system over a long period of time carries an inherent challenge: the more the system develops, the more interesting it becomes to analyze, but the data collected and the way it was collected are fixed, which limits the

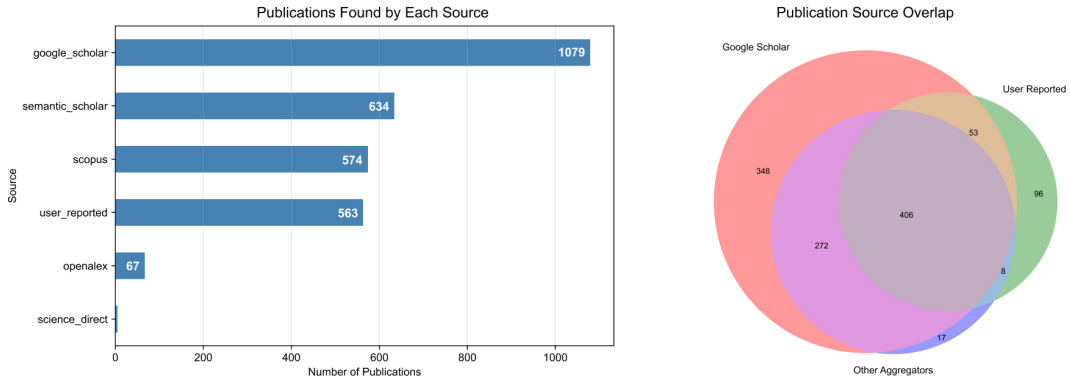


Figure 6: (left) Number of publications identified by each source. (right) Overlap and unique identifications across sources.

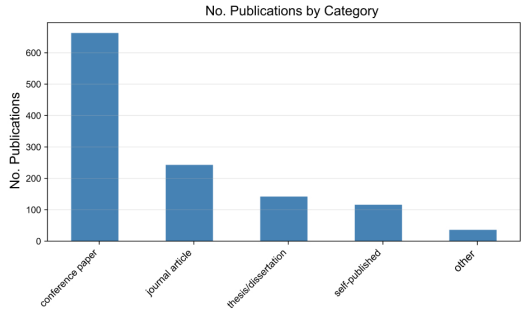


Figure 7: Count of publications by category.

questions we can ask. The preceding sections shared the questions we asked about the system; here we summarize recommendations on both the data to collect and systems to employ.

Measuring users and projects. By and large, the type of information tracked here is tracked by other projects as well and, in particular, by the XDMoD database [21] in project-specific ways that usually involve database dump tools. To support active user tracking, we built a simple exporter tool that runs as a cron job and periodically saves login events to a reporting database. This, as well as Python scripts and processes that analyze the data and generate the plots included here, along with queries for other metrics, are available in [9].

The main recommendation we make is to track and share user information that goes beyond unique user logins and allows testbed operators to differentiate user engagement with the system (the A, B, and C definitions of system user in Section 3). In particular, the ability to track active users and PIs with cumulative user numbers provides powerful insight into the system's usage patterns. Further, adding information and timing of what projects a user joined as opposed to just project membership, would have allowed us to answer questions about the number of users that belong to both education and research projects and how they move between those projects; a modification that we intend to implement. Lastly, visualizing and reviewing on a regular basis the properties of user community in terms of most active institutions, projects, and research categories – as well as visualizing the user membership via tools like Grafana or regular project email containing the most important project metrics – helps keep the finger on the pulse of how the community develops.

Measuring usage of clouds. We propose that usage of systems that explicitly make available a slice of resource to a user should be measured in terms of bare metal capacity. Further, closely tracking state transitions of those resources supports better understanding of system use and creates the opportunity to implement policies that promote its more efficient use (e.g., ensure that all reserved resources are at least reconfigured). While the data collection will be system-specific, our approach generalizes to any input which provides a "start" and "end" for resource values. For OpenStack in particular, our usage reporting collection, analysis, and plotting code are available in [8].

More than any other quality of the system, usage data collection over long stretches of time is susceptible to configuration and implementation changes as the system evolves. In our case, change in configuration to split the database between two sites, optimizations for database performance, and successive OpenStack upgrades introduced discontinuity in the tracked data. To ensure relative consistency of gathered data requires treating usage data collection as a "first class citizen" in successive migrations, something that is often both effort-intensive and not high enough priority in a system operated with few resources.

Measuring science impact. Measuring science impact is rewarding, but expensive considering the challenges of attribution, identification, de-duplication, and inclusion that may require manual steps including review. While attribution should always be rooted in a user statement, collecting user attestations can be expensive and require careful reporting structure as described in Section 5; therefore, in the best case attribution is included in the publication itself. We recommend giving users a clear and preferably unique (e.g., containing a reward number) way to acknowledge the use of the system from the beginning of the project and explicitly requiring its use in the terms and conditions. Referencing a publication describing the system is better, as it both serves as attribution and helps identify the publication through scholarly databases and search engines. Therefore, we recommend creating such a reference early in the project – we missed a trick here, having published our first paper five years after the start of the project and its impact can be seen in the graphs shown in Section 5. Periodic reminders to

acknowledge the system foster awareness of the importance of linking resource usage to scientific results and are usually effective.

Identifying publications can be effort intensive. Google Scholar yields the most comprehensive results – including information about dissertations and theses not available via any other sources – but requires a manual approach which can be expensive unless applied regularly. Less comprehensive (roughly 70%) but much cheaper results can be obtained by combining Semantic Scholar and Scopus with user reporting which is the only source of publications without explicit attribution. An independent avenue to reduce this effort is through leveraging tools and processes we published to obtain and analyze publications [7]. It is important to realize that despite all these efforts the science impact will still likely be an undercount – however, with proper handling of attribution, it will constitute a strong lower bound.

Lastly, we note that there are different types of impact that an *experimental resource/system* may generate: on one end of the spectrum, we have resources experimental in the sense that they explore the feasibility of infrastructure enabled by new technology – on the other, resources that serve as an instrument for community experimentation. The former are *experiments in themselves* that primarily explore infrastructure construction; the latter are *scientific instruments* that provide a platform for the experimentation of others (though in order to do so, they often also advance infrastructure construction). The definition of success and impact differs significantly between these two types of systems: the former will be defined by publications on system construction and its application, primarily from the construction team and its collaborators; the latter will be primarily defined by publications of platform users, not associated with the construction/operation team. Since there may be a spectrum between an experiment and a scientific instrument, we propose that as a rough distinguishing measure at least 90% of the identified scientific output (i.e., lower bound) of a scientific instrument should consist of community publications. As an example, in the case of Chameleon the (self-reported) publications produced by the team both to disseminate insights obtained in designing and operating the system (16) and explore its applications (19) were significantly under that barrier for every year of the project.

Acknowledgments

This work used resources available through the National Research Platform (NRP) at the University of California, San Diego. NRP has been developed and is supported in part by funding from the National Science Foundation, from awards 1730158, 1540112, 1541349, 1826967, 2112167, 2100237, and 2120019, as well as additional funding from community partners.

References

[1] Jason Anderson and Kate Keahey. 2022. Migrating towards Single Sign-On and Federated Identity. In *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 8, 8 pages. doi:10.1145/3491418.3530770

[2] Devin Bayly, Benjamin Kruse, and Christopher Reidy. 2024. Streamlit App for Cluster Publication Impact Exploration. In *Practice and Experience in Advanced Research Computing 2024: Human Powered Computing* (Providence, RI, USA) *(PEARC '24)*. Association for Computing Machinery, New York, NY, USA, Article 90, 4 pages. doi:10.1145/3626203.3670633

[3] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. 2023. ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good* (Portland, OR, USA) *(PEARC '23)*. Association for Computing Machinery, New York, NY, USA, 173–176. doi:10.1145/3569951.3597559

[4] Lutz Bornmann and Hans-Dieter Daniel. 2008. What do citation counts measure? A review of studies on citing behavior. *Journal of Documentation* 64, 1 (01 2008), 45–80. doi:10.1108/00220410810844150 arXiv:https://www.emerald.com/jd/article-pdf/64/1/45/1494385/00220410810844150.pdf

[5] Chameleon Cloud. 2026. Add Publication. https://chameleoncloud.org/user/projects/add/publications/. Accessed: 2025-02-06.

[6] Chameleon Cloud. 2026. Chameleon Used in Research. https://chameleoncloud.org/user/projects/chameleon-used-research/. Complete list of publications supported by Chameleon.

[7] Chameleon Cloud. 2026. Magpub - A Tool for Collecting Academic Publications. https://github.com/ChameleonCloud/magpub/. Accessed: 2026-06-17.

[8] Chameleon Cloud. 2026. Usage Exploration. https://github.com/ChameleonCloud/usage-exploration. Accessed: 2026-02-09.

[9] Chameleon Cloud. 2026. User and Project Reports. https://github.com/ChameleonCloud/user_project_reports. Accessed: 2026-02-09.

[10] Cornell University. 2024. arXiv.org e-Print Archive. https://arxiv.org. Accessed: 2026-06-16.

[11] Elsevier. 2024. Social Science Research Network (SSRN). https://www.ssrn.com. Accessed: 2026-06-16.

[12] Elsevier. 2026. Scopus. https://www.scopus.com. Abstract and citation database.

[13] Google. 2026. Google Scholar. https://scholar.google.com.

[14] Google LLC. 2025. Google LLC v. SerpApi, LLC. https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/Google_v._SerpApi__Complaint.pdf. Case No. 4:25-cv-10826, U.S. District Court for the Northern District of California, filed Dec. 19, 2025.

[15] David Hart. 2022. Measuring XSEDE: Usage Metrics for the XSEDE Federation of Resources: A comparison of evolving resource usage patterns across TeraGrid and XSEDE. In *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 12, 9 pages. doi:10.1145/3491418.3530765

[16] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons Learned from the Chameleon Testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association.

[17] Kate Keahey, Pierre Riteau, Jason Anderson, and Zhuo Zhen. 2019. Managing Allocatable Resources. In *Proceedings of the IEEE 2019 International Conference on Cloud Computing (CLOUD 2019)*.

[18] Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. 2023. Semantic Scholar. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. 106–112.

[19] Alberto Martín-Martín, Mike Thelwall, Enrique Orduna-Malea, and Emilio Delgado López-Cózar. 2021. Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations'COCI: a multidisciplinary comparison of coverage via citations. *Scientometrics* 126, 1 (2021), 871–906.

[20] OpenStack Contributors. 2026. *OpenStack Documentation*. OpenStack Foundation. https://docs.openstack.org/ Accessed: 2026-02-09.

[21] Jeffrey T. Palmer, Steven M. Gallo, Thomas R. Furlani, Matthew D. Jones, Robert L. DeLeon, Joseph P. White, Nikolay Simakov, Abani K. Patra, Jeanette Sperhac, Thomas Yearke, Ryan Rathsam, Martins Innus, Cynthia D. Cornelius, James C. Browne, William L. Barth, and Richard T. Evans. 2015. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. *Computing in Science & Engineering* 17, 4 (2015), 52–62. doi:10.1109/MCSE.2015.68

[22] Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir. 2006. Experiences building PlanetLab. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation* (Seattle, Washington) *(OSDI '06)*. USENIX Association, USA, 351–366.

[23] John W. Ratcliff and David E. Metzener. 1988. Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal* 13, 7 (1988), 46–72.

[24] Winona G. Snapp-Childs, David L. Hart, Claudia M. Costa, Julie A. Wernert, Harmony E. Jankowski, John W. Towns, and Craig A. Stewart. 2024. Evaluating Return on Investment for Cyberinfrastructure Using the International Integrated Reporting <IR> Framework. *SN Computer Science* 5, 5 (2024), 558.

[25] Craig Stewart, Julie Wernert, Claudia Costa, David Y. Hancock, Richard Knepper, and Winona Snapp-Childs. 2022. Return on Investment in Research Cyberinfrastructure: State of the Art. In *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article

60, 4 pages. doi:10.1145/3491418.3535131

[26] Craig A. Stewart, Claudia M. Costa, Julie A. Wernert, David Y. Hancock, Donald F. McMullen, Philip Blood, Robert Sinkovits, Susan Mehringer, Richard Knepper, Jeremy Fischer, Marques Bland, Gary Rogers, Peter Couvares, Terry Campbell, Harmony Jankowski, Winona Snapp-Childs, and John Towns. 2022. Metrics of financial effectiveness: Return On Investment in XSEDE, a national cyberinfrastructure coordination and support organization. In *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 5, 9 pages. doi:10.1145/3491418.3530287

[27] Craig A. Stewart, Claudia M. Costa, Julie A. Wernert, Winona Snapp-Childs, Marques Bland, Philip Blood, Terry Campbell, Peter Couvares, Jeremy Fischer, David Y. Hancock, David L. Hart, Harmony Jankowski, Richard Knepper, Donald F. McMullen, Susan Mehringer, Marlon Pierce, Gary Rogers, Robert S. Sinkovits, and John Towns. 2023. Use of accounting concepts to study research: return on investment in XSEDE, a US cyberinfrastructure service. *Scientometrics* 128, 6 (2023), 3225–3255.

[28] Steven Tuecke, Rachana Ananthakrishnan, Kyle Chard, Mattias Lidman, Brendan McCollam, Stephen Rosen, and Ian Foster. 2016. Globus auth: A research identity and access management platform. In *2016 IEEE 12th International Conference on e-Science (e-Science)*. 203–212. doi:10.1109/eScience.2016.7870901

[29] Gregor von Laszewski, Fugang Wang, and Geoffrey C. Fox. 2021. Comprehensive Evaluation of XSEDE's Scientific Impact using Semantic Scholar Data. In *Practice and Experience in Advanced Research Computing 2021: Evolution Across All Dimensions* (Boston, MA, USA) *(PEARC '21)*. Association for Computing Machinery, New York, NY, USA, Article 50, 4 pages. doi:10.1145/3437359.3465601

[30] Derek Weitzel, Ashton Graves, Sam Albin, Huijun Zhu, Frank Wuerthwein, Mahidhar Tatineni, Dmitry Mishin, Elham Khoda, Mohammad Sada, Larry Smarr, Thomas DeFanti, and John Graham. 2025. The National Research Platform: Stretched, Multi-Tenant, Scientific Kubernetes Cluster. In *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration (PEARC '25)*. Association for Computing Machinery, New York, NY, USA, Article 69, 5 pages. doi:10.1145/3708035.3736060

[31] Kiduk Yang and Lokman I. Meho. 2006. Citation Analysis: A Comparison of Google Scholar, Scopus, and Web of Science. *Proceedings of the American Society for Information Science and Technology* 43, 1 (2006), 1–15. doi:10.1002/meet.14504301185