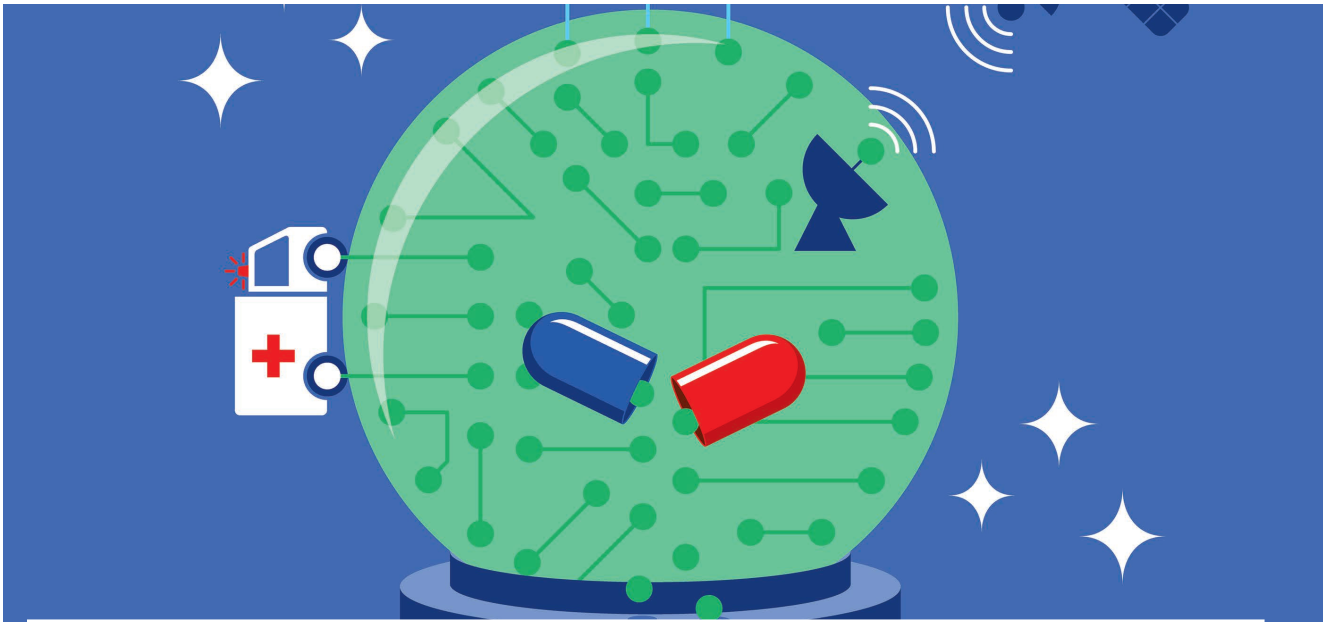

Ushering in the Low Earth Orbit Compute Cloud Era

Ketan Bhardwaj, **Vaibhav Bhosale**, and **Ada Gavrilovska**, Georgia Institute of Technology

The last few years have seen low Earth orbit satellites becoming an integral part of global connectivity, including high-speed broadband, cellular, and Internet of Things. This article outlines a vision in which this new infrastructure tier is used like today's clouds to flexibly deploy diverse services in space.

Today, we are witnessing a surge in the number of satellites in the low Earth orbit (LEO) as part of megaconstellations. These megaconstellations are providing a variety of commercial services, including connectivity, such as high-speed broadband and cellular, Internet of Things (IoT) analytics, Earth observation services, weather forecasting, managing natural disasters, and many more.

This surge in deploying compute infrastructure to LEO is driven by advancements in launch services, which are becoming more efficient and sustainable through reusable launch vehicles and commoditization of spacecraft production as well as in traditional space computing-related issues, such as shielding from cosmic radiation and handling associated failures. These advancements have poised LEO satellites to be a major player in global connectivity.

History suggests that the compute infrastructure used for connectivity is followed by new rich compute services. For example, content delivery networks are deployed within Internet service providers, and edge computing services are enabled by cellular infrastructure and wireless access.[1] We predict that history will repeat itself and the current LEO connectivity will soon be complemented with a LEO cloud. A LEO cloud is more than just physical

Digital Object Identifier 10.1109/MC.2025.3589214
Date of current version: 25 September 2025

compute infrastructure in the LEO. Rather, it is the software stack that binds the plethora of physical compute available in satellites to form a frictionless, elastic computing infrastructure similar to the terrestrial cloud.

However, terrestrial cloud software stacks are not equipped to handle the unique aspects that must be considered for a LEO cloud. Terrestrial software stacks are built assuming a stationary physical compute infrastructure and hence inadequate to handle the motion of compute nodes in a LEO cloud. There are open, often unasked, questions about software systems, such as: What does availability mean in a LEO cloud? What are the appropriate robustness guarantees? How should the limited and variable energy available on the LEO nodes be handled? What are the efficient architectures for communicating with command and control ground stations on Earth? How should cybersecurity concerns in a LEO cloud be addressed?

There is a silver lining. There is a high degree of predictability and structure in the seemingly chaotic LEO cloud. The orbits and paths of LEO satellites are well understood, and they follow predictable patterns, which can be leveraged to provide availability guarantees on Earth.[2] The solar energy available (and stored) in satellite batteries can also be predicted fairly accurately, opening up new energy-aware system design.[3] Leveraging this structured motion requires adopting fundamentally different principles in software systems that are proactive and selective.

In this article, we go over the trends informing our prediction of the LEO cloud, briefly outline recent work that answers some of the aforementioned questions concretely, and present a call to action for the research community in this new greenfield area.

TRENDS SUPPORTING A LEO CLOUD

Policy shifts

In the 2000s, U.S. policy in space underwent a significant change, leading to the allocation of billions of dollars in contracts to private companies. This policy shift catalyzed a remarkable transformation in resource allocation, supply chain dynamics, and innovation in LEO satellite technology. It also brought a substantial cultural and strategic evolution in the space industry. Companies like SpaceX and Blue Origin are emblematic of this new space age. Additionally, this era in space exploration is characterized by increasing global interest and participation. Nations around the world, including major players like the United States, Russia, China, the European Union, and India, are actively involved, often adopting competitive stances. The commercialization of space continues to evolve, paralleling established collaborations between agencies, such as the National Aeronautics and Space Administration (NASA), the European Space Agency, and other global entities, with new private companies, such as SpaceX, OneWeb, and Planet. This multifaceted development marks a new and complex era in space exploration and utilization.

THERE IS A HIGH DEGREE OF PREDICTABILITY AND STRUCTURE IN THE SEEMINGLY CHAOTIC LEO CLOUD.

Technological drivers

Reusable rockets and reduced satellite manufacturing costs have significantly boosted the number and frequency of launches. A range of industry players, including SpaceX, Amazon's Kuiper, OneWeb, and others, are deploying large-scale LEO constellations. For instance, SpaceX's Starlink currently operates nearly 8,000 LEO satellites supported by nearly 200 ground stations. Unlike the traditional satellite ecosystem where ground contact was intermittent, modern LEO constellations benefit from a dense ground station deployment and inter-satellite links (ISLs), extending ground connectivity of the satellites to near continuous coverage in certain cases. This enables fine-grained, dynamic control mechanisms driven by the ground infrastructure in contrast to the earlier sporadic rendezvous. Highlighting the growing interest in this domain, the Tiansuan constellation has been launched as a research platform exploring emerging directions in satellite networking, computing, and operating systems.[4]

Technology roadmap

The current satellite compute capabilities are also evolving, both in terms of the compute hardware available as well as their connectivity with ground stations and with other satellites. For example, every Starlink satellite consists of up to 250 nodes.[5] Further, accelerators, such as GPUs and field programmable gate arrays have been deployed on satellites to assist processing of terabytes of data generated by the satellites.[6] On the communications side, optical lasers provide high bandwidths of nearly 200 gigabits per second (Gb/s) on the ISLs. There are also improvements in ground-satellite links (GSLs) used to communicate with ground infrastructure using higher frequency bands, such as Ku, Ka, V, and E that provide 20–80 Gb/s bandwidth per link. Figure 1 showcases the different components of the LEO satellite ecosystem.

A RANGE OF INDUSTRY PLAYERS, INCLUDING SpaceX, AMAZON'S KUIPER, OneWeb, AND OTHERS, ARE DEPLOYING LARGE-SCALE LEO CONSTELLATIONS.

RECENT ADVANCES

Richness of LEO compute applications

These trends point to a new, emerging cloud frontier propelled by the increased availability of compute infrastructure. As satellites process increasingly large volumes of data and expand their roles in communication, integrating computing capabilities directly onboard enhances overall service quality. This approach not only reduces latency but also alleviates the burden on constrained satellite-to-ground links that would otherwise be overwhelmed by terrestrial compute offloading.

Recent commercial and research advances have demonstrated a number of use cases, such as enabling global cellular coverage,[7] deploying machine learning inference[8] for environmental monitoring (for example, detecting forest fires), surveilling infrastructure (for example, oil rigs), and extending connectivity to remote areas. Additionally, there have been proposals for deploying space data centers[9] to offload the compute requirements of other constellations. We envision such services to be hosted in a LEO cloud which will provide ubiquitous access to compute due to the geographically distributed presence of LEO satellites.

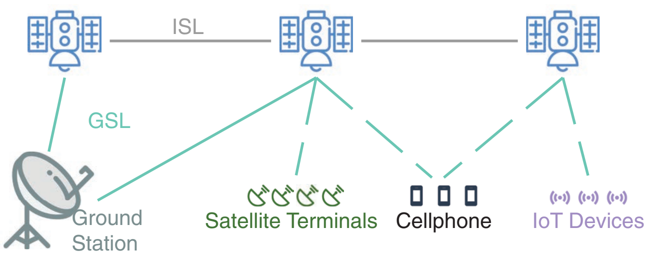

FIGURE 1. LEO satellite ecosystem.

Deployment considerations and constraints

To realize the full potential of a LEO cloud, it is essential that applications are deployed on satellites that can serve the specific geographic areas they are intended for to ensure the lowest application latencies and minimal network load on the satellite links. Seemingly straightforward, this proves a significant challenge as LEO satellites are accessible for a maximum of 4.5 min (median 3.5 min) from any geographical location due to their rapid motion.[10] Therefore, once a satellite moves out of range, direct access to the application is lost, requiring the use of additional ISL or GSL hops, thus obviating the latency benefits expected from the onboard compute of a LEO cloud.

A trivial approach for continuous availability is to deploy the same application on all the satellites so that there is at least one satellite always directly accessible. This approach is suitable for some types of applications, such as deploying stateless cellular network functions[7] where the state resides with the client, or machine learning applications aimed at reducing the data transmission on the constrained GSLs.[8] However, this is wasteful for applications relevant only to specific geographies. This limits the number of applications that can be efficiently served by a LEO cloud and thus reduces its overall utility. Further, this approach effectively rules out the deployment of any stateful applications as maintaining state consistency across hundreds or thousands of application instances will be prohibitively complex and expensive.

Recent work has attempted to address this challenge by pushing toward localized orchestration logic. EagleEye[11] uses inter-satellite coordination in a mixed-resolution, leader–follower satellite constellation where a leader satellite using a low-resolution camera continuously scans the Earth to dynamically capture regions of interest that are captured by the followers with high-resolution cameras. Such selective targeting enables greater coverage of higher-value targets while also reducing redundant data collection and thus increasing the efficacy of the constellation. Serval[12] takes a different approach by splitting the application logic (referred to as query bifurcation) across the ground stations and the satellites. It splits the application logic into lightweight filtering-like queries deployed on satellites and complex processing reserved for ground stations. This reduces the latency for time-sensitive Earth observation tasks. However, both these systems are specifically curated for observation tasks and require significant changes to the application logic, which do not generalize to other user-facing or stateful applications deployed on a LEO cloud. Further, EagleEye also requires a specific constellation, which may be prohibitively expensive or infeasible for many already deployed constellations.

This limitation exists due to *missing necessary abstractions* needed to specify where applications need to be available and *missing systems support* to ensure availability by accurately identifying deployment targets. To provide this functionality, terrestrial cloud systems rely on a zone abstraction, which is statically determined based on the location of infrastructure, but this assumption breaks down for a LEO cloud where infrastructure is not stationary.

Consider this illustrative example. Researchers from two universities, Georgia Tech and University of Georgia (UGA), want to deploy an edge application catering to both campuses. If zones are statically defined as per terrestrial zones, it will result in Atlanta and Athens having their own zones. However, neither of them alone will suffice since they cannot provide continuous coverage to all the intended end users. If the application is deployed in only one zone, users in the other zone will intermittently require multiple hops to reach the satellite, thus leading to increased bandwidth and latency (Figure 2). To mitigate this problem, the same application would need to be deployed in both zones. An augmented definition of LEO zones would enable the application provider to define a zone that can cover both campuses so that, instead of two instances, a single application instance would suffice.

LEO zones

LEO zones, similar to terrestrial edge/cloud zones, are incorporated as part of the application deployment configuration to specify the geographical region of availability. Just like how a terrestrial zone guides the orchestrator to use nodes from a designated cluster, the definition of LEO zones guides the LEO edge orchestrator about which satellites can serve a specific LEO zone and, consequently, to select the satellite for deploying an application. However, unlike in terrestrial orchestrators, the choice is made not among the nodes known to a zone a priori, but rather from a dynamically determined set of satellite nodes based on their present (and anticipated) geopositioning. The selection is based on the LEO zone's *space projected region*, which is a cone that is calculated based on the allowed elevation angle. The elevation angle determines a LEO satellite's instantaneous accessibility from a given point on Earth and also the duration for which a particular satellite remains accessible. For instance, in Figure 3, the LEO zone definition would restrict the orchestrator to only deploy applications on the blue-colored satellites. While, for simplicity, in this illustration we use circular regions, similar calculations can be done for nontrivially shaped regions. The LEO zones abstraction is different from the way zones have been defined conventionally for terrestrial infrastructure: that is, instead of having the presence of infrastructure define the availability zones (and the ensuing tight coupling), LEO zones enable application providers to define their own availability zones based on their end users.

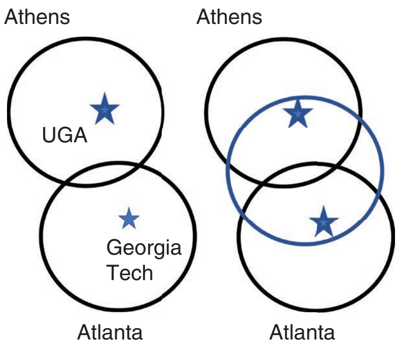

FIGURE 2. Comparing coverage of statically defined regions with flexibly defined LEO zones. The flexible definition of a LEO zone enables zone definition based on the target users in Georgia Tech and UGA.

Orchestrating for satellite mobility

Challenges due to satellite mobility. Figure 4 shows that an application instance needs to run on at least 475 satellites (in the first Starlink shell with 1,584 satellites) to ensure continuous accessibility to a single stationary end user for a 100-minute interval. Alternatively, continuous accessibility of LEO applications can be achieved through continuous application handovers, akin to handovers in cellular networks. However, in cellular networks, handovers take 30–60 milliseconds (ms) and are triggered *reactively*, that is, when a user moves out of range of a stationary base station. When the infrastructure itself is moving at high speeds of 27,000 km/hr, such a reactive approach can be detrimental to availability. Fortunately, satellite mobility can be predicted by established path models, such as Simplified General Perturbations 4 (SGP4). While this predictability and the ensuing structure has been studied and utilized in other settings, such as network routing,[10] system support to treat it as a first-order concern is missing from the terrestrial cloud system and is necessary for the LEO cloud.

Orchestration system for the LEO cloud. We built a software orchestration system, Krios,[2] that enables application providers to deploy their applications on LEO satellites without having to worry about the mobility of the infrastructure, that is, the satellite nodes. Unlike terrestrial orchestrators, Krios uses LEO zones to ensure that applications remain available in those zones despite the motion of satellites. This benefits LEO applications by affording them with reduced bandwidth usage and latency, enabling a practical deployment. For every application deployment on satellites, Krios determines when the satellite will no longer be visible from the geographical region and initiates a handover of the application to the next set of suitable satellites. This ensures that the application is always directly accessible to all end users in the specified LEO zones, thus maintaining geostationarity of the application relative to end users in the LEO zones. Our experience in building Krios informed us of two core aspects of system software design that require reconsideration: mechanisms and policies.

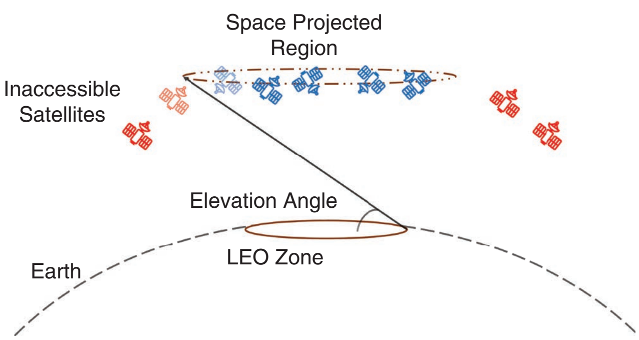

FIGURE 3. Defining the space projected region for a LEO zone based on elevation angle constraints.

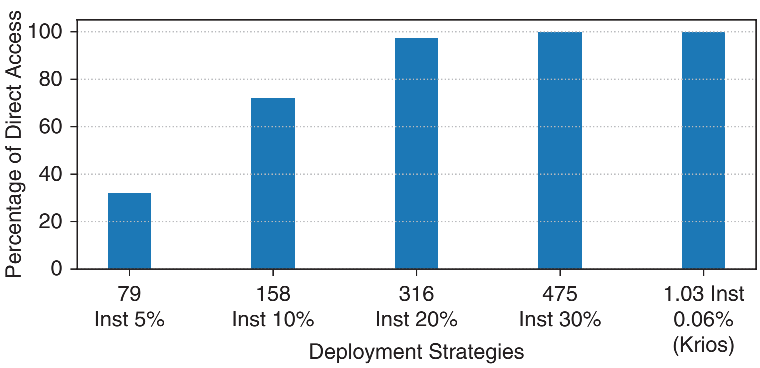

FIGURE 4. Availability of an application deployed on 5%, 10%, 20%, and 30% satellites in the first shell of Starlink during an orbital period (100 minutes) compared to just 1.03 instances with appropriate abstraction and accompanying system.

Proactive mechanisms. The dynamism introduced due to the LEO satellite movement requires frequent orchestration actions. Embedding this in the critical path of application performance leads to performance variability and downtime. Instead, a LEO cloud requires proactive, ahead-of-time management. Concerning orchestrating application deployment, Krios incorporates new proactive mechanisms for application handover from a satellite currently leaving a zone to a satellite capable of serving a particular zone. Specifically, it introduces a LEO controller to perform handovers transparently to applications. Initializing a new application instance on the target node takes a nontrivial amount of time, and using the built-in proactive mechanisms hides this time.

Mobility-aware policies. Along with new mechanisms, new policies are needed in the system to determine where to hand over a particular application. Terrestrial orchestrators use load and health metrics to distinguish among the nodes they manage. A LEO orchestrator must consider the movement of its nodes to minimize churn and overheads. Krios augments the filtering and scoring stages of the orchestration scheduler with new policies that rank nodes based on their presence in the space-projected region of the LEO zone, in addition to the default resource availability requirements. This is done by computing the position of the satellites using satellite path models. In doing so, Krios prioritizes nodes that will remain accessible to the LEO zone for the longest period of time. This is in contrast to other approaches, such as in the cellular context, where the user is generally handed over to the base station that is closest.

The evaluation of the impact of these proactive mechanisms and mobility-aware policies with respect to concrete metrics, such as availability, latency, bytes transferred, and system overhead, provides evidence to support our prediction that LEO compute clouds can be practically realized.

OPEN CHALLENGES

The recent advances summarized previously provide evidence that the LEO cloud is feasible and beneficial. However, a number of open challenges must be addressed to make it practical, ranging from control plane architectures that enable satellites to function as compute nodes of a LEO cloud, to challenges in energy availability in space, storage designs that can support state management across satellites, application or software system design and implementation suitable for LEO cloud deployment, to unique security and reliability. All of these furnish interesting research questions for the computing community to address. The following sections provide details on some of the open questions that we forecast will form a new exciting area of computing research.

ENERGY IS SOMETHING THAT IS ASSUMED TO BE AVAILABLE IN TERRESTRIAL CLOUDS, WHEREAS SATELLITES RELY ON SOLAR ENERGY FOR CONTINUOUS OPERATION.

Control plane designs

Traditionally, the control operations on satellites involve detailed a priori planning across ground stations encountered in a satellite's orbit. The UNIFORM 1 mission[13] required planning for actions, such as when images will be captured and when they will be downlinked to the ground. However, such a planning-based solution becomes infeasible with the growing number of satellites. Terrestrial control plane systems designed for large-scale and dynamic control actions also fail to suffice. The compute nodes (LEO satellites) are ubiquitous, geo-distributed, and constantly in motion, while the control nodes (ground stations) are geo-distributed, but not collocated with the compute nodes they manage like in terrestrial control planes. As satellites move, they connect to different ground stations. This represents a shared pool of compute nodes that change their affinity to control nodes, something that does not exist in terrestrial infrastructure. While current versions of control planes are mostly based on centralized designs (for example, Starlink), they suffer from multiple challenges, including higher control-plane latencies and potential management or scalability hotspots. Moreover, even other intuitive choices, such as terrestrial federated control plane or highly available distributed control plane deployments, suffer from their own shortcomings, such as lack of support for dynamic partitions, or high consensus latencies when applied to this dynamic compute infrastructure. As the number of LEO clouds and their use cases increase, we posit that the control plane design needs to be evolved as well, to consider distributed designs that behave like its terrestrial counterparts but address the aforementioned shortcomings.

Dynamic energy availability in space

Similarly, energy is something that is assumed to be available in terrestrial clouds, whereas satellites rely on solar energy for continuous operation. As they move in their orbits, their solar exposure changes, which results in varying amounts of energy being harvested. During eclipses, when the Earth prevents sunlight from reaching the satellite directly, the satellite harvests zero energy and relies on stored battery.

In our prior work,[3] we investigated the total harvested energy of satellites and gathered insights about the energy variations. We studied the power harvesting patterns of 185 Planet Labs[6] satellites. Figure 5 shows the solar power harvested by three satellites over a period of 100 min. Sat A (green) has continuous access to sunlight and hence is able to harvest similar power over time. However, Sat B (blue) and Sat C (red) do not have continuous access to the Sun and hence have times when no energy is harvested. Interestingly though, they harvest greater energy during their sunlit intervals, thus also experiencing larger variations compared to Sat A. In addition to this, we also showed that a number of other factors have significant impact on harvested solar power. For satellites of different sizes (for example, from 1 U to 3 U configurations), the harvested energy varies substantially (from 20 kW to over 70 kW over a single orbit). The satellite inclination angle and its position in the orbital plane also result in up to 2× variation in harvested energy.

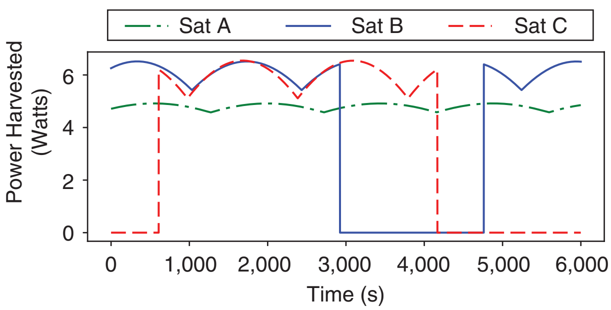

FIGURE 5. Variations in the solar power harvested by three satellites in 100 min.

These observations suggest that a LEO cloud exhibits significant levels of spatial and temporal variations in energy (and thus, compute resource) availability that must be explicitly considered. For instance, a satellite about to enter an eclipse state needs to conserve its energy compared to a satellite that has no eclipse period, even if their current energy is similar. LEO cloud schedulers should capture the current and future energy availability of a satellite to ensure that the satellite does not run out of energy at any point when applications are deployed. Additionally, they should manage application energy budgets to restrict the energy consumption of applications—similarly to how cgroups provide CPU/memory budgets. Doing so would require deploying support for online energy metering on LEO satellites to estimate the instantaneous energy consumed by an application. An energy budget, as an abstraction, can also be exposed to applications and runtimes, to allow for energy-aware application adaptation mechanisms, for example, which trade resource utilization for accuracy.

Storage system designs

Another challenge is the lifetime of storage devices in LEO satellites and how the storage software systems could drive new requirements. To support stateful applications in a LEO cloud, state handovers must be triggered along with application handovers. In terrestrial clouds, those states are part of a virtual storage volume, and, on handover, need to be migrated to a different node. However, continuous migrations, such as what would be needed in LEO cloud, significantly impact the disk lifetimes. For instance, most write-heavy solid-state drives have drive writes per day in the range of 30–50. In contrast, a LEO application requires more than 400–600 handovers in a day. Incurring 10–20× more state transfers for multiple applications will reduce the lifetime of a hard drive from five years to mere months (assuming the state of all the applications completely fill the hard drive). One alternative is to support only stateless applications in the LEO cloud. This severely diminishes its potential use, as has been evident with serverless deployments where a significant amount of work has gone in to addressing state for serverless applications. Another alternative is to store the state terrestrially. While this eliminates the storage system challenge, it also limits the benefits afforded by a LEO cloud.

Therefore, there is a need to rethink the design of the underlying storage systems in the context of the diverse LEO cloud use cases and applications. We forecast that for a LEO cloud to be sustainable and reliable, a concerted effort is needed in the storage area that may include increasing the storage redundancy in LEO satellites, lazy state migrations at the storage layer, and specialized storage disks with longer lifetimes.

Software engineering

Software designed for satellites (and for space, in general) mandates different standards (for example, NASA software engineering requirements)[14] and frameworks (for example, F prime)[15] compared to terrestrial software. This stems from the emphasis of terrestrial software on scalability, flexibility, and rapid iteration, compared to space software focusing on reliability, safety-critical compliance, and long-term operability under extreme conditions. To open up space infrastructure to a broader range of developers and users through a LEO cloud, these two approaches need to be interoperable to lower the barrier for developing LEO cloud applications that offer both agility and required safety. We predict that advances made in the use of generative artificial intelligence (AI) for software development and engineering could potentially bridge this gap by enabling quick conversion of existing terrestrial software implementations to be space-compliant or by providing easier access to compliance tests for the new software developed for the LEO cloud using generative AI. Regardless, we forecast a need for concerted effort to bridge the gaps in compliance-driven space software stack development and the current agile practices of terrestrial developments.

Security aspects

Beyond the cybersecurity concerns faced by the terrestrial compute clouds, LEO compute clouds are exposed to unique cybersecurity and safety concerns. These stem from the lack of available information about closed LEO satellite megaconstellation deployments that has created an unclear perimeter, obfuscated, and perhaps increased attack surface, new communication channel attack models, and many others. Moreover, recent studies have identified emerging cybersecurity threats, including reception-based geolocation attack,[16] satellite link saturation attack,[17] and novel attack vectors that exploit solar energy availability in satellites to launch denial-of-service attacks—threats largely unaccounted for in terrestrial settings. Additional challenges from the perspective of LEO cloud orchestrators are due to variability in vulnerabilities as satellites cross specific regions (for example, over politically conflicted zones), dynamic evaluation of security-hardening tradeoffs, and the current uncertainties in regulatory policies for computing in space. We forecast that addressing these challenges requires new frameworks for collaboration among satellite operators, cloud service providers, cybersecurity experts, and regulatory bodies, in addition to research on new attacks and defenses.

The ongoing growth in the LEO satellite ecosystem propelled by the latest advances in launches and space hardware has led to increased demand for communication and computational services on these satellites. We envision future LEO compute clouds that provide similar abstractions and guarantees as terrestrial clouds and enable ease of application deployment and management. Recent advances provide evidence of the feasibility of this vision and raise a number of pressing challenges that must be addressed.

ABOUT THE AUTHORS

KETAN BHARDWAJ is a senior research engineer at the College of Computing, Georgia Institute of Technology, Atlanta, GA 30333 USA. His research interests include computer systems distributed–decentralized to low-level operating system support, functional correctness and performance, and systems support for security. Bhardwaj received his Ph.D. in computer science from Georgia Institute of Technology. Contact him at ketanbj@cc.gatech.edu.

VAIBHAV BHOSALE is a Ph.D. student at the School of Computer Science at the Georgia Institute of Technology, Atlanta, GA 30333 USA. His research interests include building systems and networking solutions for LEO satellite networks. Bhosale received his master's in computer science from Georgia Institute of Technology. He is a Student Member of IEEE. Contact him at vbhosale6@gatech.edu.

ADA GAVRILOVSKA is a professor at the School of Computer Science at the Georgia Institute of Technology, Atlanta, GA 30333 USA. Her research interests include designing systems for emerging technologies. Gavrilovska received her Ph.D. in computer science from Georgia Institute of Technology. She is a Member of IEEE. Contact her at ada@cc.gatech.edu.

Importantly, the impact of developing these technologies can extend beyond LEO.[18] Long-term settlement missions to the Moon (for example, Artemis) and Mars (for example, Lightship) envision using satellites in the respective orbits for observation, communication and navigation. Transmitting all the information back to Earth for these satellites is even more expensive compared to LEO and necessitates onboard compute. We posit that these satellite platforms represent an extension of a LEO cloud, and the lessons learned and experience in enabling and managing a dynamic LEO cloud offering would greatly help in designing cloud solutions for lunar and Martian environments.

ACKNOWLEDGMENT

This project was partially supported by the National Science Foundation (NSF) Grant CNS-2345827 as well as funding from Cisco, VMware, and the Georgia Smart program.

REFERENCES

1. M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017, doi: 10.1109/MC.2017.9.
2. V. Bhosale, A. Gavrilovska, and K. Bhardwaj, "Krios: Scheduling abstractions and mechanisms for enabling a LEO compute cloud," in *Proc. ACM Symp. Cloud Comput.*, 2024, pp. 322–340.
3. V. Bhosale, K. Bhardwaj, and A. Gavrilovska, "Don't let your LEO edge fade at night," in *Proc. 1st Workshop Hot Topics Syst. Infrastructure*, 2023. [Online]. Available: https://hotinfra23.github.io/papers/hotinfra23-paper23.pdf
4. S. Wang, Q. Li, M. Xu, X. Ma, A. Zhou, and Q. Sun, "Tiansuan constellation: An open research platform," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, 2021, pp. 91–101, doi: 10.1109/EDGE53862.2021.00022.
5. A. Badshah, N. Morris, and M. Monson, "Over-the-vacuum update–Starlink's approach for reliably upgrading software on thousands of satellites," in *Proc. AIAA/USU Conf. Small Satell.*, SSC23-II-01, 2023. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2023/all2023/72/
6. "Planet: Daily satellite data and insights about earth." Planet Labs PBC. Accessed: Feb. 1, 2025. [Online]. Available: https://www.planet.com/
7. Y. Li et al., "A case for stateless mobile core network functions in space," in *Proc. ACM SIGCOMM Conf.*, 2022, pp. 298–313.
8. B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. 35th Int. Conf. Archit. Support Program. Lang. Operating Syst.*, 2020, pp. 939–954.
9. N. Bleier, M. H. Mubarik, G. R. Swenson, and R. Kumar, "Space microdatacenters," in *Proc. 56th Annu. IEEE/ACM Int. Symp. Microarchit.*, 2023, pp. 900–915.
10. V. Bhosale, A. Saeed, K. Bhardwaj, and A. Gavrilovska, "A characterization of route variability in LEO satellite networks," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2023, pp. 313–342.
11. Z. Cheng, B. Denby, K. McCleary, and B. Lucia, "EagleEye: Nanosatellite constellation design for high-coverage, high-resolution sensing," in *Proc. 29th ACM Int. Conf. Archit. Support Program. Lang. Operating Syst.*, vol. 1, 2024, pp. 117–132.
12. B. Tao, O. Chabra, I. Janveja, I. Gupta, and D. Vasisht, "Known knowns and unknowns: Near-realtime earth observation via query bifurcation in serval," in *Proc. 21st USENIX Symp. Netw. Syst. Des. Implementation (NSDI 24)*, 2024, pp. 809–824.
13. T. Hiramatsu et al., "UNIFORM-1 ground system: Mission planning, scheduling, control, and data processing," in *Proc. AIAA/USU Conf. Small Satell., SSC14-II-1*, 2014. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2014/IntellSoftware/1/
14. "NASA software engineering procedural requirements, standards, and related resources." NASA (.gov). Accessed: Jul. 24, 2025. [Online]. Available: https://www.nasa.gov/intelligent-systems-division/software-management-office/nasa-software-engineering-procedural-requirements-standards-and-related-resources/
15. "F-prime: A flight-proven, multi-platform, open-source flight software framework." GitHub. Accessed: Jul. 24, 2025. [Online]. Available: https://github.com/nasa/fprime
16. E. Jedermann, M. Strohmeier, V. Lenders, and J. Schmitt, "RECORD: A reception-only region determination attack on LEO satellite users," in *Proc. 33rd USENIX Secur. Symp. (USENIX Secur.)*, 2024, pp. 6113–6130.
17. S. Yoon, M. Kang, and J.-Y. Choi, "Security attacks against the availability of low earth orbit satellite networks," in *Proc. 12th Int. Conf. Netw., Commun. Comput.*, 2023, pp. 64–69.
18. T. Pfandzelter and D. Bermbach, "Can orbital servers provide mars-wide edge computing?" in *Proc. 1st ACM MobiCom Workshop Satell. Netw. Comput.*, 2023, pp. 7–12.