Achieving Scalability for Mission-Critical Systems in the Cloud

Dr. Bruce D. Holenstein >> **President & CEO** >> Gravic, Inc.
Dr. Bill Highleyman >> **Managing Editor** >> Availability Digest
Paul J. Holenstein >> **Executive Vice President** >> Gravic, Inc.

In our previous paper, "Improving Reliability via Redundant Processing,"[1] we discussed the reliability and the availability of mission-critical systems. We noted that 'reliability' is the probability that a system will produce correct outputs. 'Availability' is the probability that the system is operational. Reliability and availability are two of the three pillars of mission-critical systems. The third pillar is scalability, leading to the acronym RAS.

In this article, we look to add scalability to mission-critical systems. Scalability means that the system capacity can be expanded easily, and without taking an outage, if the workload on the system increases. Likewise, the system capacity can be scaled back if the workload diminishes.

Modifying system capacity based on a fluctuating workload precludes replacing the servers and databases with different hardware to match a new workload. Such an approach is very expensive and time-consuming, taking perhaps months to order, install, and test new hardware. Certainly, the scale-up or scale-down of the system will lag far behind the workload fluctuations, which might occur on an hourly or daily basis.

Rather, the system architecture must be flexible from a scalability viewpoint. Capacity must be able to increase or decrease easily, perhaps on a daily or weekly basis. For instance, a retail application may need additional capacity as the holiday season approaches. Once the holidays are over, the excess capacity can be retired.

A straightforward approach to achieving scalability is to burst the application to a cloud. Clouds comprise multiple physical servers, each capable of supporting several virtual servers. Thus, an application that suddenly needs additional capacity can be extended to a cloud and given access to one or more of the cloud's virtual servers. When the application's excess capacity needs have diminished, the application releases the virtual server resources that it consumed in the cloud.

Other ways to achieve such scalability include active/active systems and Pathway Domains. We look at these techniques first, and then move on to scalability using cloud resources.

[1] Improving Reliability via Redundant Processing, The Connection; March/April 2017.

Active/Active Systems

What Is An Active/Active System?

An active/active system comprises two or more servers and two or more copies of the application database. Each server has access to a copy of the database (Figure 1). The databases keep synchronized via bidirectional data replication. Whenever a change is made to one database, that change is replicated immediately to the other databases in the system.

Therefore, any server in the active/active system can process any transaction and achieve the same result that would occur if any other server had serviced the transaction in the system.

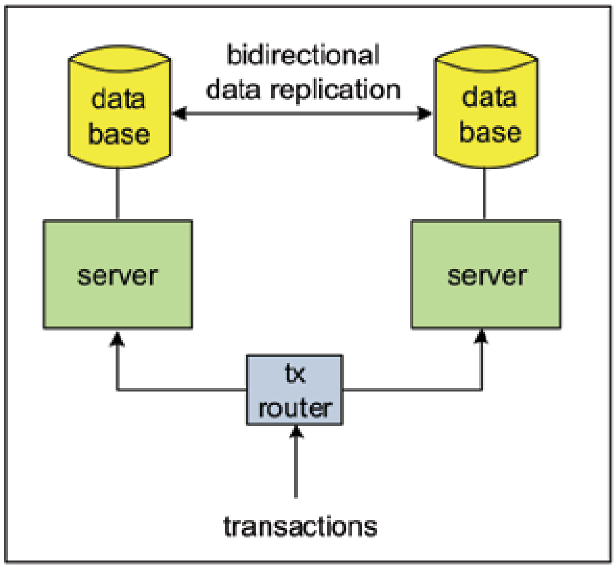

Figure 1: An Active/Active System

Consequently, an active/active system provides continuous availability. If a server fails in an active/active system, further transactions are simply routed to surviving servers. Recovery is so fast that users will be unaware that an outage took place. To restore a server to service, it is rebooted with the application(s), attached to a database copy, and added back into the transaction-routing facility.

To achieve continuous availability, there must be at least two copies of the application database in the system. If there are only two servers, each server typically will have its own copy of the database, as shown in Figure 1. To improve availability, each server in a two-server configuration also has access to the database on the other server in case it loses its own database, as shown in Figure 2.

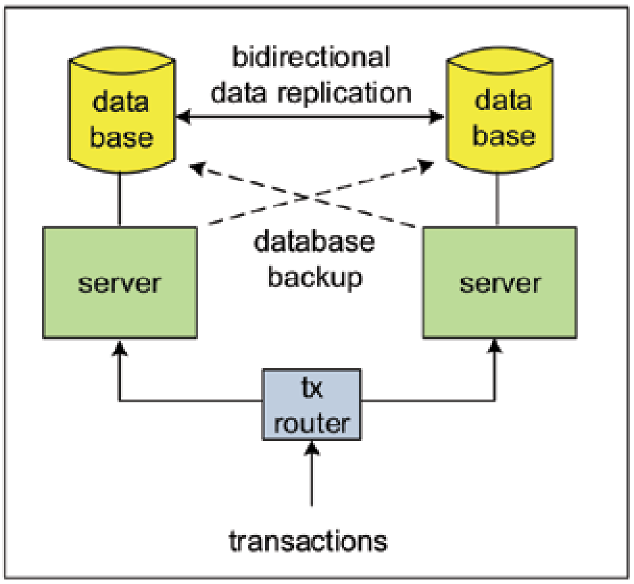

Figure 2: Active/Active System with Database Backup

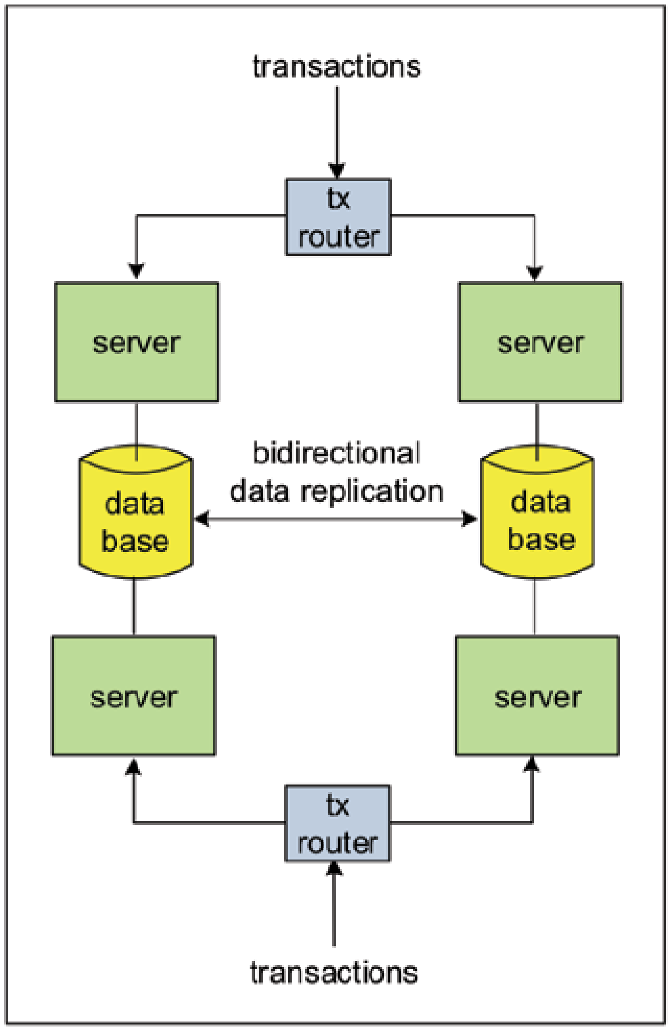

Figure 3: Scaling an Active/Active System

Active/Active Systems Are Inherently Scalable

Capacity can be added to an active/active system by adding additional servers, as shown in Figure 3. Each server must have access to a copy of the database so that a transaction can be sent to any server. If there is excess capacity, some of the servers can be retired and moved to other applications. Consequently, an active/active system is scalable, both in terms of adding capacity and in terms of reducing capacity.

In general, there is only a need for two copies of the application database to ensure continuous availability, though more copies can be provided. Each of the multiple servers in the active/active system must have access to at least one copy of the application database. However, if access is provided to only one database, the attached server becomes inoperable if the database fails.

System availability can be improved by giving every server the capability to reconnect to an operational database. Thus, if a database fails, the affected servers can reconnect to another copy of the database and then continue to operate.

Scaling an active/active system can be accomplished with no

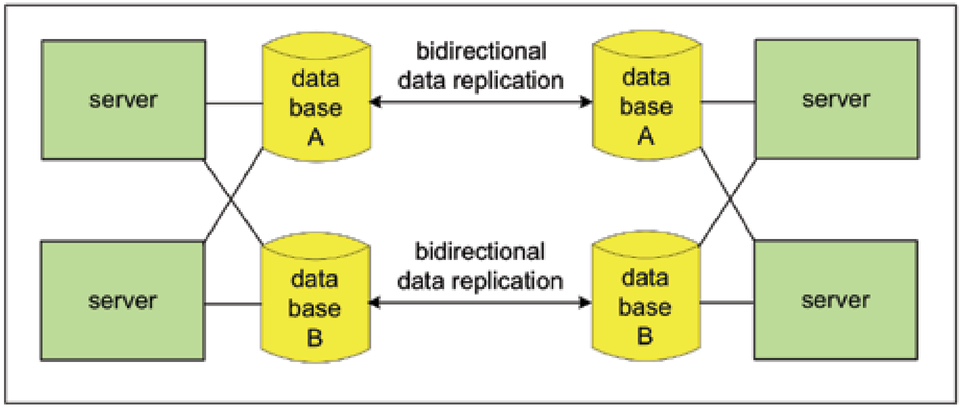

Figure 4: Active/Active System with Direct Access to a Partitioned Database

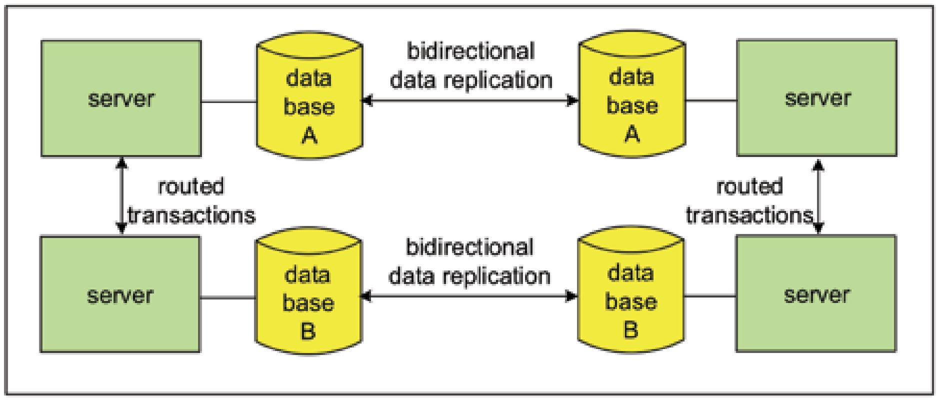

Figure 5: Active/Active System with Indirect Access to a Partitioned Database

downtime. If a server is added to increase capacity, it is brought online and allowed to attach to a copy of the database. Then the transaction routing table is expanded to include the new server, and it can begin to process transactions along with the other servers.

If a server is removed from the system to decrease capacity, its identity is first removed from the transaction routing table. Next, the server completes the processing of any transactions with which it already is involved. Only then, is the server shut down and removed from the active/active system.

Scaling an Active/Active Database via Partitioning

The database in an active/active system can be scaled via partitioning. Many application databases are large and partitioned across several disk volumes. As with a single database disk, every partition must be redundant. There must be at least two copies of each partition to maintain availability of the system in the event of a disk failure.

Every server in the active/active system must have access to all partitions. One method of access is shown in Figure 4. Each server attaches to one copy of every database partition (partition A and partition B in Figure 4). In this way, each server has direct access to the entire database.

An alternative technique is for a server to access a copy of an unattached partition by routing requests to a server attached to the partition, as shown in Figure 5.

Pathway Domains

What Is Pathway?

A software facility known as 'Pathway' provides the facility for fault tolerance and scalability for application programs in HPE NonStop systems. Applications written in a Pathway environment provide fault tolerance and scalability with little effort on the part of the application programmer.

Pathway Architecture

Pathway applications comprise a collection of server classes, as shown in Figure 6. A server class contains one or more identical stateless application servers (processes) designed to process a particular transaction (or set of transactions). Typically, a server in a server class receives a transaction from a requesting client and routes the database updates to the appropriate databases. The HPE NonStop Transaction Management Facility (TMF) is responsible for maintaining the Atomicity, Consistency, Isolation, and Durability (ACID) properties of the transaction as it updates the database. NonStop systems support three databases –

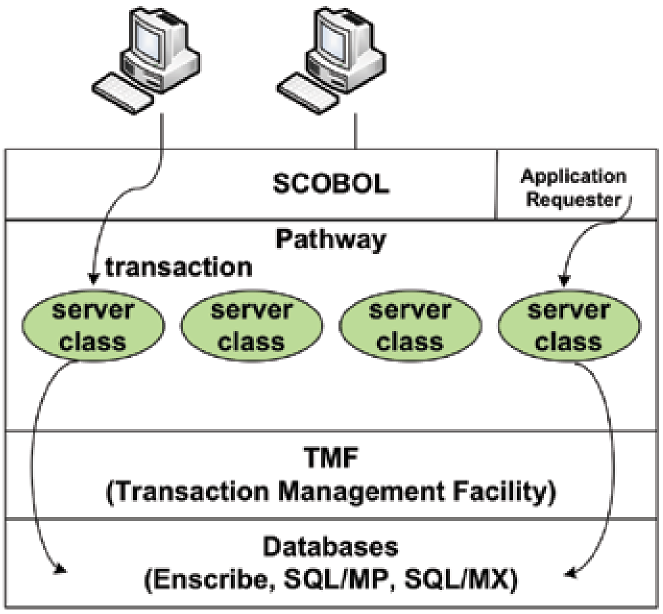

Figure 6: A Pathway Environment

Enscribe, a relational file system, SQL/MP, a SQL database, and SQL/MX, a newer SQL database that complies more closely with the ANSI SQL standard.

Pathway Scalability

The server classes provide the scalability of the Pathway environment. The servers in a server class can be spread among several processors in a NonStop system, with transaction requests automatically distributed between them. If a server class becomes overloaded, Pathway can spawn additional server processes for the server class. If the load on a server class falls to the point where not all of its servers are needed, it terminates some of them.

Communication between processes in the NonStop system is via interprocess messages, where there is no common memory used. Therefore, there is no fundamental limit to the expandability of a system, which is a near-linear function of the number of processors in the system. In fact, it is possible to link up to 255 systems (4,080 processors) in a cluster; each additional processor increases the system capacity by 98% of that processor's capacity, (which is not the case for symmetric multiprocessor (SMP) systems).

Pathway Domains

While Pathway provides scalability across a single NonStop system, Pathway Domains enables scalability across multiple NonStop systems[2]. Multiple identical Pathway environments can be configured as a Pathway Domain that behaves as a single, large Pathway application. Any of the Pathway environments within a domain can be taken down for maintenance while the remaining environments within the domain continue processing work with no interruption. This ability enables online updating of server code and rebalancing of Pathway environments without planned outages. It also relaxes the configuration limits of a single Pathway system, removing barriers to scalability. Server classes are replicated across all Pathway environments in the domain, and requests are automatically load-balanced across the domain.

Since a remote Pathway environment can be configured as part of a domain, a logical Pathway server class can span multiple (up to four) NonStop systems. This configuration allows a NonStop node in a domain to be taken down for maintenance while the application remains available on other nodes. Pathway automatically routes requests only to the available nodes in a domain. This routing also increases scalability because a Pathway application can span multiple NonStop nodes, which may actually be part of a bigger ServerNet or Infiniband cluster containing hundreds of NonStop processors. So if a Pathway application reaches the scalability limits of a single NonStop system, simply distribute it across multiple NonStop nodes using Pathway Domains.

Of course, since each server node in the Pathway Domain must have access to the application database, each system is provided access to a copy of the database. The database copies are kept synchronized via bidirectional replication.

Bursting to the Cloud for Scalability

With the advent of cloud technology, a new method of scalability has evolved. Applications can scale internally as described above; but if they need further capacity, they can be burst to a private cloud[3], a public cloud, or a hybrid mix of private and public clouds. A copy of the database must be in the cloud or be available to applications running in the cloud.

When an application requires more capacity than is available via its physical servers, a copy of the application can be sent to the cloud. The cloud spawns virtual servers on which the application can run. As the application demands even more capacity, the cloud provides further virtual server resources. If the capacity needs of the application diminish, the cloud releases some of the assigned resources. In this way, the application is scaled to handle its workload.

Clouds, whether public or private, comprise inexpensive commodity hardware, which supports scalability quite well. However, how does one support the reliability and availability aspects of RAS in clouds? Though not dealing with clouds, we explored reliability and availability to a great extent in our two-part paper "Improving Availability via Staggered Systems," published in the November/December 2016 and January/February 2017 issues of The Connection.[4] By staggering the start time of a system and its backup, the probability distributions of failure will not line up. Therefore, when one system is most likely to fail, the other system is much less likely to fail. If the starting times are not staggered, the times for the peak probability of failure will align; and increase the likelihood of both systems being down at the same time.

When bursting an application to the cloud for scalability, an important technique to guarantee reliability and availability is to run redundant systems in a Validation Configuration. In this architecture, redundant arrays of inexpensive computers (RAIC) are deployed in a cloud (public or private), as shown in Figure 7. As discussed in our previous paper, "Improving Reliability via Redundant Processing," each RAIC system in a redundant pair periodically exchanges 'indicia.' An indicium is a representation of the current state of the system. Each system in the redundant pair compares the indicia sent by its mate to its own indicia. If the indicia agree, the systems are behaving properly; and processing continues. If the indicia do not agree, the systems are halted; and diagnostics are run to identify the faulty system.

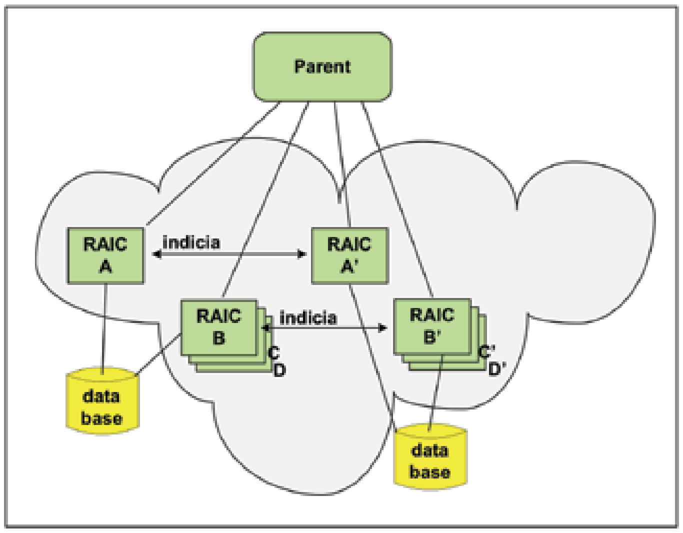

Figure 7: Bursting to a Single Cloud for Scalability

Figure 7 shows the RAIC systems running in the same cloud. To protect against cloud failures, RAIC redundant pairs could be run in separate clouds, as shown in Figure 8. Bursting to multiple clouds is most effective from an availability perspective if the clouds are provided by different vendors, such as Microsoft Azure, Amazon AWS, or the Google Cloud Platform. However, it requires that the application be written for deployment in multiple clouds.

Summary

Applications often require additional capacity during critical times. Scalability is an inherent and important attribute of mission-critical systems. Both the servers and the database must be scalable. Active/active systems can provide scalability to servers as well as to the database. However, adding physical servers and disk systems to achieve greater capacity is typically not feasible, especially if the additional capacity needs are temporary.

Scalability can be achieved via the use of Pathway Domains that span multiple NonStop systems, providing significant scalability. It can also be achieved by bursting applications to RAIC servers in one or more clouds. RAIC servers use inexpensive commodity hardware and can be reused between applications,

[2] See for background HP Pathway Domains and Data Replication – Perfect Together!, The Connection; September/October 2014.
[3] See for background Adding High Availability to the Cloud, The Connection; July/August 2014.
[4] Improving Availability via Staggered Systems Part 1: MTTF- Mean Time to Failure, The Connection; November/December 2016.
Improving Availability via Staggered Systems Part 2: Mitigating Redundant Failures via System Staggering, The Connection; January/February 2017.

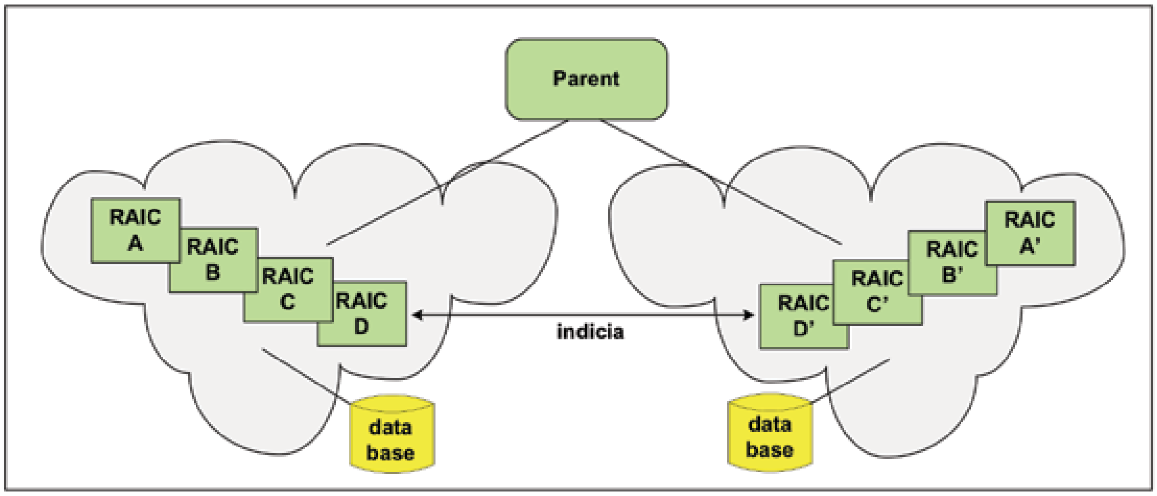

Figure 8: Bursting to Multiple Clouds for Scalability and Reliability

as scalability needs change, thus providing an economic cloud platform for scalability-on-demand.

We find that the attributes of RAS can be achieved as follows:

R	Reliability	Use Validation Configurations that match indicia to ensure that there are no data errors
A	Availability	Provide redundancy in all critical components. Use staggered starts to eliminate correlated failure modes.
S	Scalability	Use Pathway Domains to span NonStop systems. Burst applications needing additional capacity to a cloud.

For HPE NonStop users, maximizing the RAS for mission-critical systems can now be accomplished by leveraging the new Virtualized NonStop technology.

Dr. Bruce D. Holenstein, President and CEO. Dr. Holenstein leads all aspects of Gravic, Inc. as President and CEO. He started company operations with his brother, Paul, in 1980, and is presently leading the company through the changes needed to accommodate significant future growth. His technical fields of expertise include algorithms, mathematical modeling, availability architectures, data replication, pattern recognition systems, process control and turnkey software development. Dr. Holenstein is a well-known author of articles and books on high availability systems. He received his BSEE from Bucknell University and his Ph.D. in Astronomy and Astrophysics from the University of Pennsylvania.

Dr. Bill Highleyman is the Managing Editor of The Availability Digest (www.availabilitydigest.com), a monthly, online publication and a resource of information on high- and continuous availability topics. His years of experience in the design and implementation of mission-critical systems have made him a popular seminar speaker and a sought-after technical writer. Dr. Highleyman is a past chairman of ITUG, the former HP NonStop Users' Group, the holder of numerous U.S. patents, the author of Performance Analysis of Transaction Processing Systems, and the co-author of the three-volume series, Breaking the Availability Barrier.

Paul J. Holenstein is Executive Vice President, Gravic, Inc. He has direct responsibility for the Gravic, Inc. Shadowbase Products Group and is a Senior Fellow at Gravic Labs, the company's intellectual property group. He has previously held various positions in technology consulting companies, from software engineer through technical management to business development, beginning his career as a Tandem (HPE NonStop) developer in 1980. His technical areas of expertise include high availability designs and architectures, data replication technologies, heterogeneous application and data integration, and communications and performance analysis. Mr. Holenstein holds many patents in the field of data replication and synchronization, writes extensively on high and continuous availability topics, and co-authored Breaking the Availability Barrier, a three-volume book series. He received his BSCE from Bucknell University, a MSCS from Villanova University, and is an HPE Master Accredited Systems Engineer (MASE). To contact the author, please email: SBProductManagement@gravic.com . Hewlett Packard Enterprise directly sells and supports HPE Shadowbase Solutions (www.ShadowbaseSoftware.com); please contact your local HPE account team.