Research Report

FREEFORM DYNAMICS

in association with

PURESTORAGE®

Modernizing database technology stacks

Remember the storage upgrade option

Freeform Dynamics Ltd, 2020

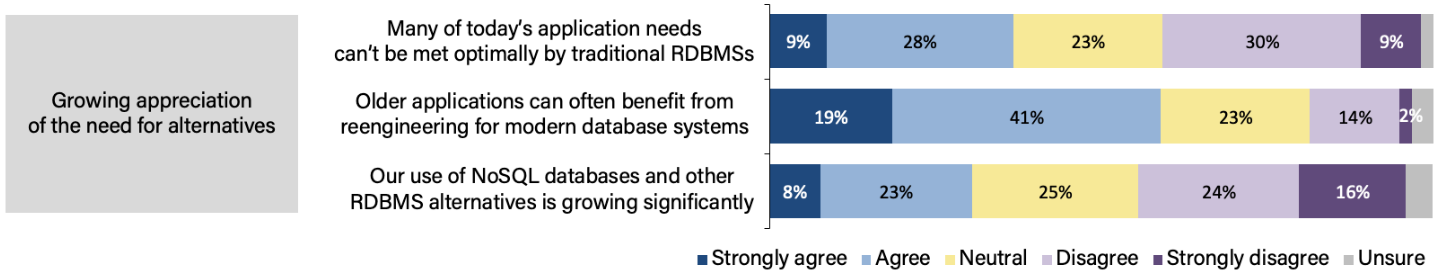
Database needs are evolving

Relational database management systems (RDBMSs) are incredibly versatile. It's one of the reasons they became the default for mainstream database requirements back in the early 1990s, and remain dominant now. Since they first emerged onto the scene, however, the application landscape has changed dramatically. In today's highly connected digital environment, data is often diverse, unstructured, widely-distributed, high-volume and/or fast moving. In many cases, we are also dealing with web-scale applications and services that dictate a level of performance and scalability that most would have thought inconceivable 30 years ago.

Against this background, while talented database designers and SQL programmers can make RDBMSs do pretty much anything - including things they were never designed to do - the old saying "just because you can, doesn't mean you should" is very pertinent. Given the 'Jack of all trades' nature of RDBMSs, alternative technologies have emerged to address specific types of need in a more targeted and optimal manner.

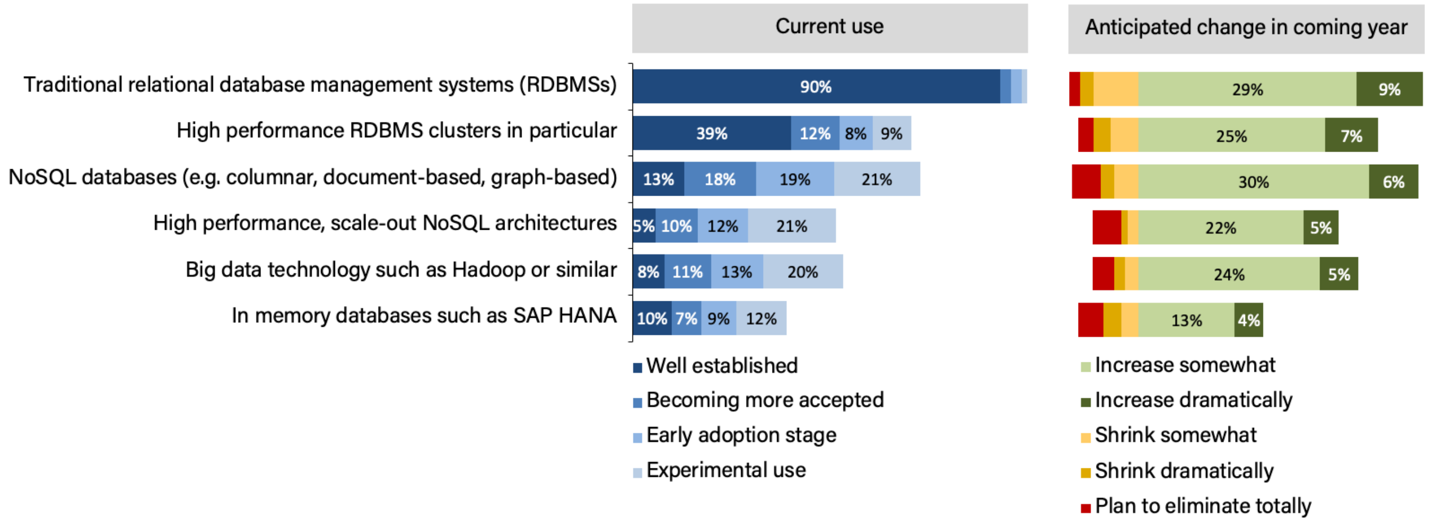
This evolution of both requirements and technology was acknowledged by many of the 225 IT professionals taking part in a recent survey conducted by Freeform Dynamics in collaboration with a major tech news site.

Thinking about your use of database technology, how much do you agree or disagree with the following?



Drilling into the database landscape changes in more detail, we can see the kinds of technologies that have begun to gain traction, and how their use is anticipated to increase over time.

How would you describe your current use of the following, and how is this changing?

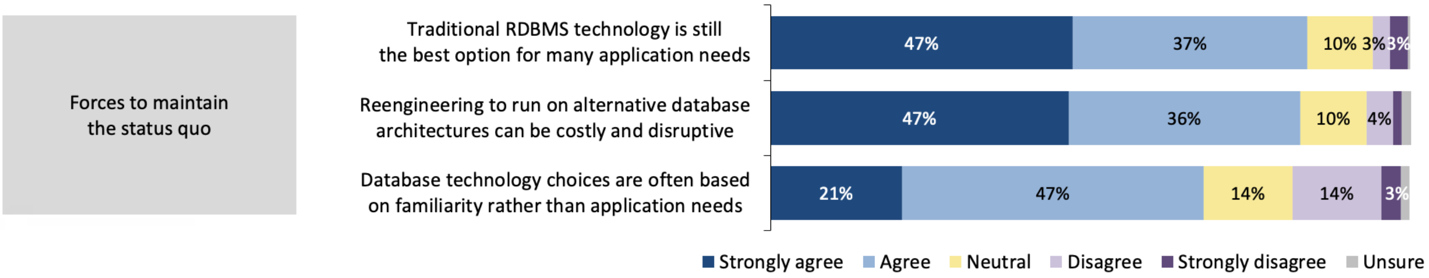


What's also notable from this chart, however, is that allegiance to relational technology is in no way diminishing. Indeed, according to our survey, RDBMS use is more likely to 'increase dramatically' than any other type of database technology, and that's clearly against an already dominant position. There are a few reasons for this.

Pain, force of habit and lack of awareness

The continued exclusive focus on relational technology by some could be because RDBMSs really can meet all of their application needs in an optimal manner. Given the way in which RDBMS vendors and open source projects have extended the core feature set over the years to deal with different types of data and use cases, this is certainly a plausible explanation. That being said, a couple of other possible explanations came through in the research to do with the pain of switching to alternative database architectures, and basic force of habit.

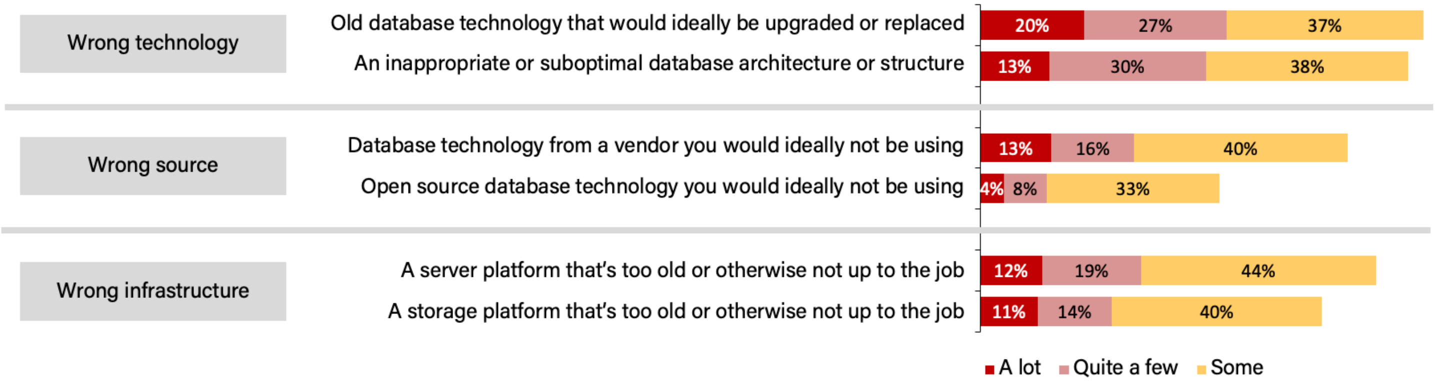
Thinking about your use of database technology, how much do you agree or disagree with the following?



We can probably add lack of knowledge and awareness to this list, not because of negligence or disinterest, but because it's hard for resource-constrained IT teams with constant deadlines and other pressures to find time to explore new options. Whatever the cause, the upshot is that RDBMSs frequently get used in situations where an alternative database technology would be more appropriate.

If we bring this together with the activities of those who like to experiment, e.g. developers dipping into the pool of open source database offerings, then perhaps unilaterally moving to the commercially supported editions of solutions they have discovered, the upshot is a range of different mismatches within application stacks.

Looking across your systems landscape, how many application workloads are sitting on the following?



The degree to which these kinds of mismatches matter will vary depending on the situation. If it's an old version of database software underpinning an application that's not particularly demanding, the problem might simply be the vendor pulling support. If it's the software driving a high-volume consumer-facing website that's bottlenecking because the relational model can't handle the access patterns, that's a serious business issue.

It's also worth highlighting that mismatches are not mutually exclusive. As a simple example, you might have a core business application that's both struggling and costing a lot to maintain because it's served by an aging database engine sitting on the original hardware it first went live with over a decade ago.

The mention of hardware brings us to the main topic of our paper, which is how modernizing the storage layer potentially opens up some interesting 'quick(er) win' opportunities in relation to database-driven workloads.

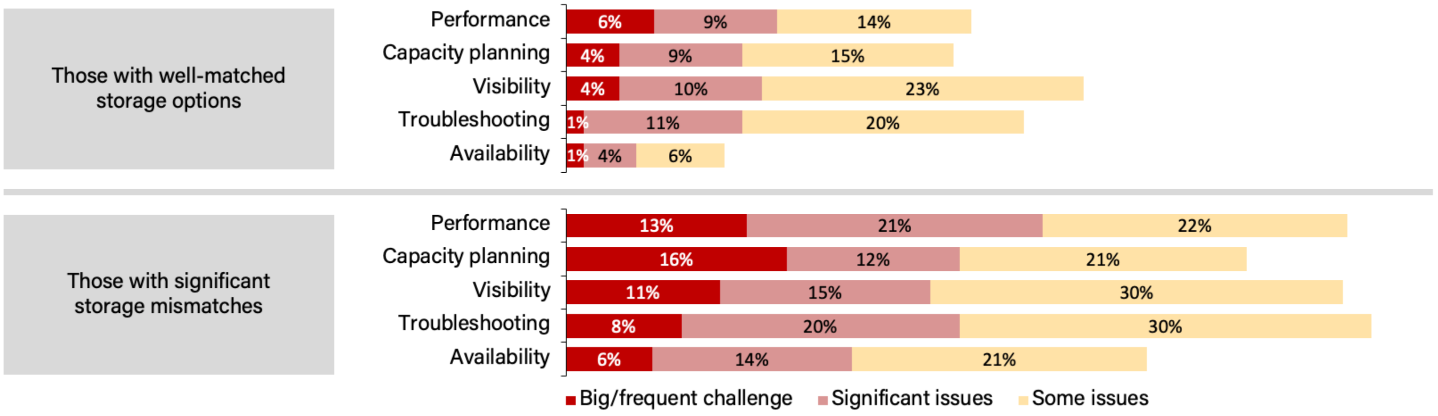
Why spend time talking about the storage layer?

When considering the role of storage we need to acknowledge the hard fact that most people, even within IT, don't think about it that much. Do they even care about it at all? The truth is they do, but in the same way they care about the plumbing in their home - they take it for granted until it fails to work as it should.

'Failing to work as it should' in the context of the storage technology underpinning database management systems most often takes the form of performance problems, excessive outages and hitting capacity limits. From an ops perspective, when issues arise without any warning, that makes things even worse. Most painful of all is when something goes horribly wrong that impacts users and/or customers, and you don't have the means to troubleshoot quickly and effectively.

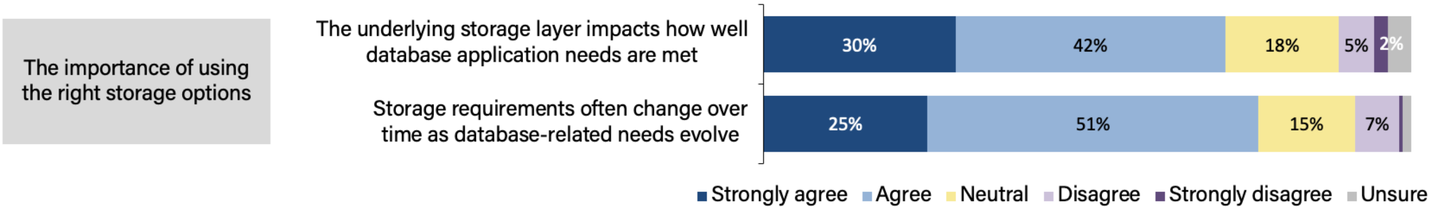
As a simple illustration of why storage matters, we divided our survey sample into two groups based on whether they said they had databases running on storage platforms that were 'too old or otherwise not up to the job' (see the last item on the previous chart). Those who cited no issues of this kind (approx a third of the sample), were considered to have 'well-matched' storage options in place, with the remainder regarded as having 'significant storage mismatches'. The difference in levels of challenge experienced by the two groups are very noticeable.

How much do you experience the following issues with your existing database and storage infrastructure?



Now as with all such analysis, we have to be careful not to blindly infer causation from correlation, and there are clearly lots of factors (skills, tools, best practices, etc) that will have an impact in addition to whether the storage layer is fit for purpose. The differences we see, though, are probably no coincidence, and most people taking part in the research seemed to intuitively understand the dependencies.

How much do you agree or disagree with the following?



The second bar on that last chart is a reminder that gaps between what the database layer needs and what the storage layer is capable of delivering can appear and grow over time as business requirements, application functionality and usage patterns continue to evolve.

This brings us back to the storage modernization opportunity.

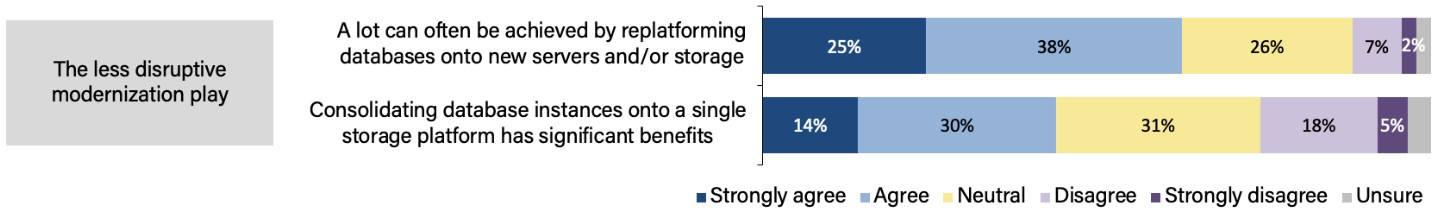
The ‘lift and shift’ alternative to full re-engineering

Based on what we have discussed so far, it should be clear that if you are designing and/or implementing a new application, it makes sense to spend time thinking not just about the database architecture, but the nature of the storage platform that will underpin it - also, of course, taking into account how things might evolve over time. But what about existing database environments that are troublesome or under-performing?

There will be occasions when it makes sense to embark on a full re-engineering exercise and switch to an alternative database architecture. This could happen if the nature of the application has morphed in a way that means it would now be better served by different database technology. Full re-engineering may also be the best option if a style of database has moved into the mainstream that is a much better fit than the original.

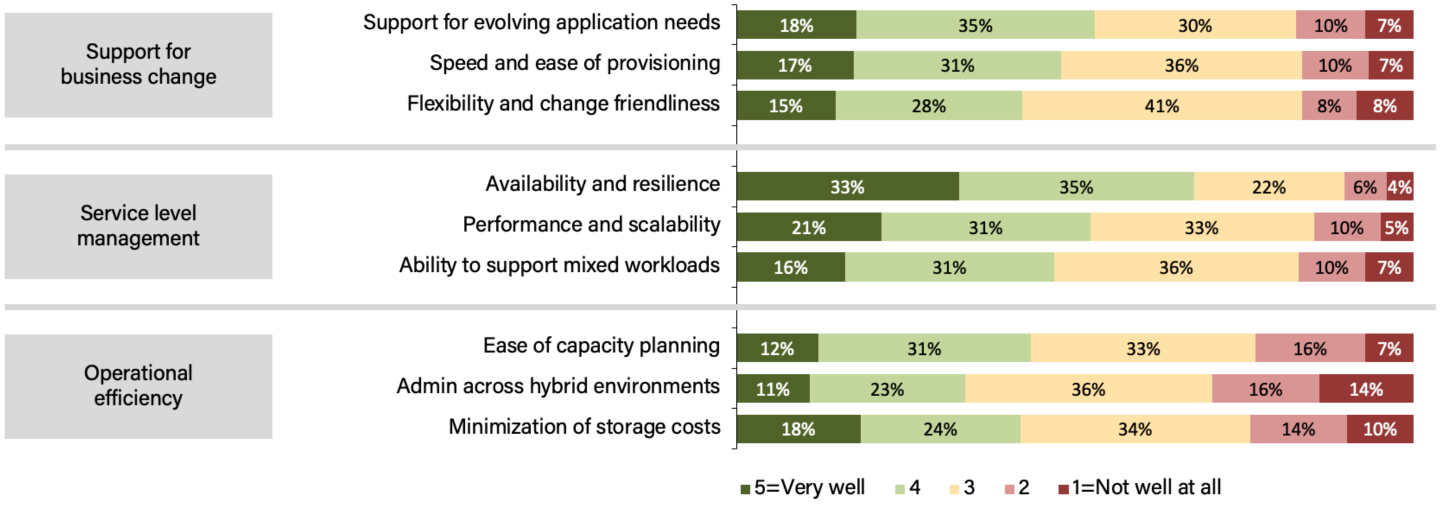
In a lot of cases, however, a storage refresh may be all that’s needed to boost performance, availability and efficiency. As a simple example, migration to a modern all-flash storage solution can totally transform an application in terms of user experience, reliability, scalability, simplicity and overall operational manageability.

How much do you agree or disagree with the following?

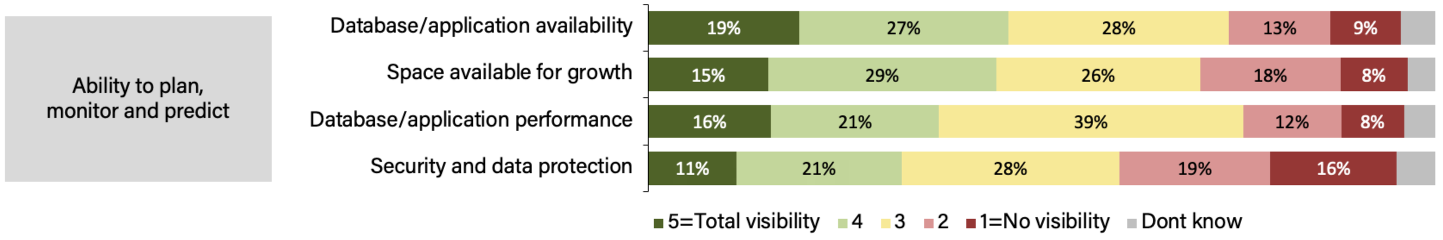


Opportunities exist to drive efficiency further by consolidating at a storage level. Whatever your aims, however, you may need to acquire new storage solutions given that existing infrastructure often falls short in key areas.

How well would you say your current storage infrastructure delivers against the following?

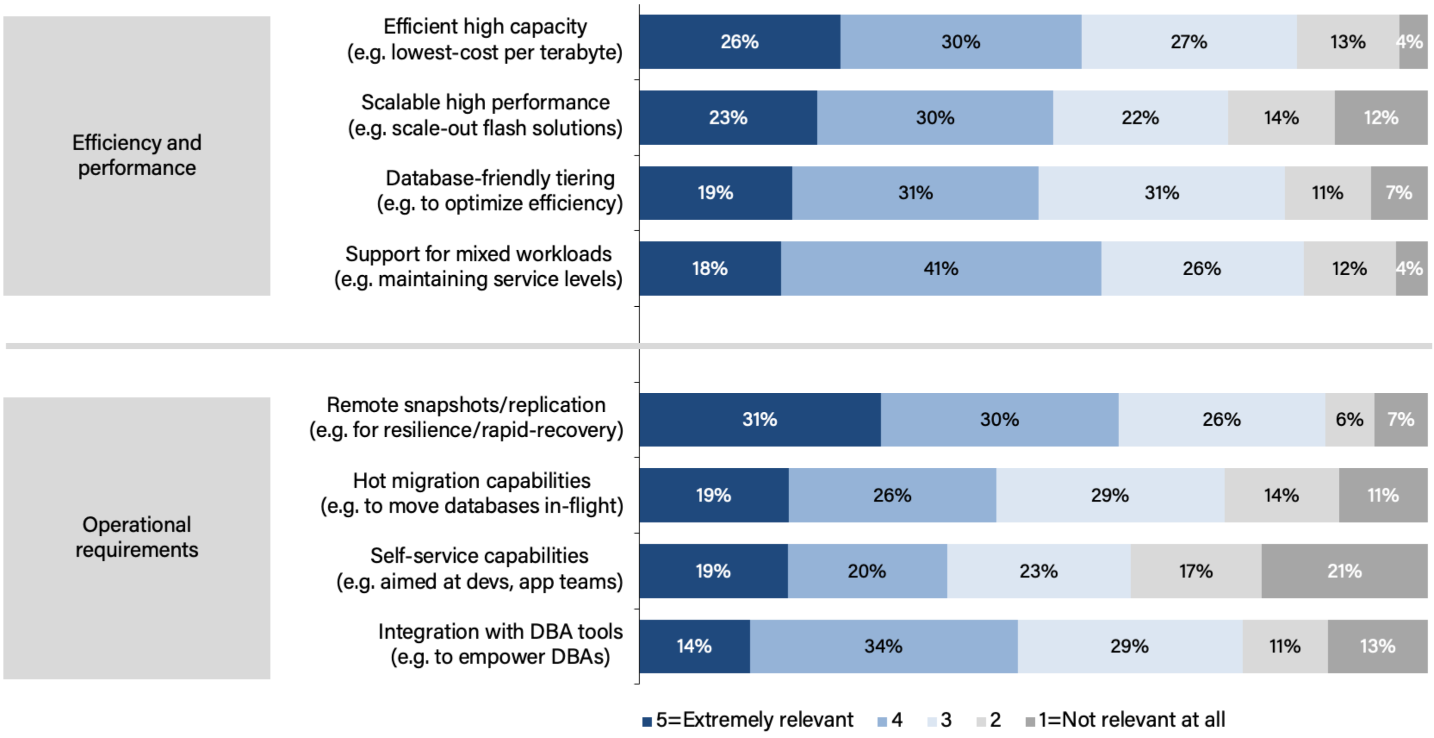


How much visibility do your current storage systems give into the following?



Making the right storage choices

So what should you look for when procuring storage specifically to underpin database management systems? Well, this will be influenced to a degree by application requirements, and/or whether you are looking to support multiple database engines or instances on the same platform (i.e. mixed workloads). Here are some ideas to consider, however, when either reviewing your existing storage platforms or selecting replacements for them.



Most of what we see here is self-explanatory, but it's worth highlighting a number of requirements that are particularly important to keep pace with the trends we are seeing more broadly across IT. We've already mentioned support for mixed workloads, for example, and this aligns with the idea of creating shared, flexible resource pools popularized (for good reason) by the march of public cloud services.

The growing interest in self-service capabilities can also be attributed to the impact of cloud-related experiences on mindsets and expectations, though adoption of DevOps is another driving force here. A common objective in the software and application delivery context is allowing developers and application teams to self-provision resources, especially at the upstream dev/test/staging points in the application lifecycle.

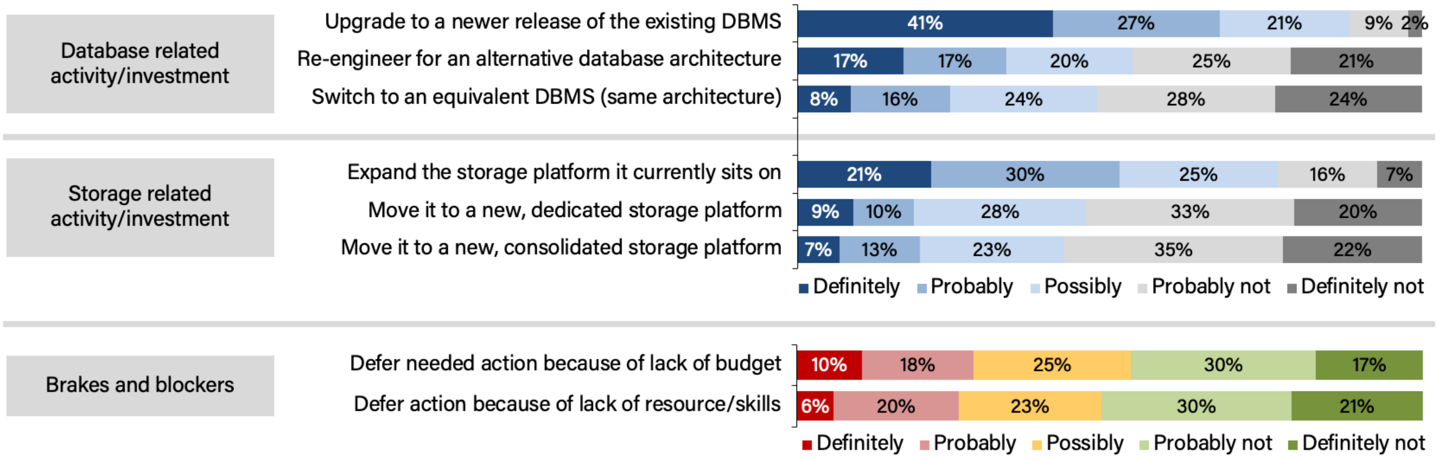
Turning to another key constituency, self-service also appeals to DBAs, who benefit in general from storage monitoring and management functionality being surfaced within the familiar tools they use on a day-to-day basis (all subject to appropriate storage admin controls, of course). Another DBA consideration is then the avoidance of clashes between the way they manage placement of data in the database environment, and any self-management capabilities implemented at a storage level, such as automated tiering.

The last couple of capabilities we will mention are remote snapshotting/replication and hot-migration. These are in recognition of the increasing need for 24x7 uptime. In many cases, you want to avoid taking the database down in order to backup, recover or relocate a business-critical data set. Capabilities here have arguably been playing catchup with the live-migration and workload management features we have seen for a while in relation to the compute part of the equation.

Pulling it all together

Looking at current intentions, quite a few study participants have definite plans to upgrade at the RDBMS software level, e.g. moving to the latest release of Oracle, SQL Server or MySQL. This is always a good point to review storage requirements. Other possible triggers for storage system reappraisal include application re-engineering exercises and switching database software suppliers. With the exception of simply adding more capacity, intentions around storage layer upgrades are noticeably softer.

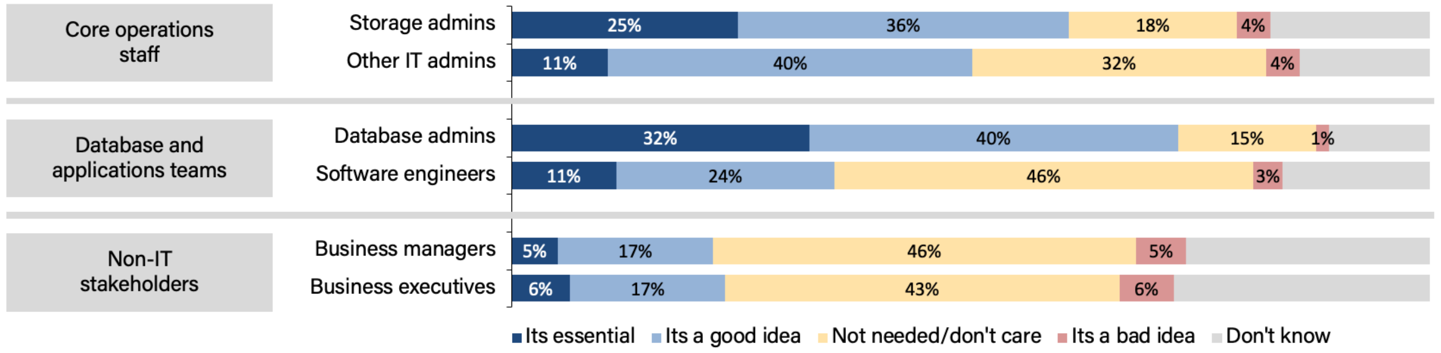
In the coming 12 months, how likely are you to do the following with one or more key applications?



Looking at the brakes and blockers noted on the above chart, the challenge around skills and resources is worth considering further in the context of storage. At first sight, the latest feature-rich all-flash arrays, for example, can raise fears to do with unfamiliarity and perceived complexity. The reality, however, is quite the opposite. Compared to old storage environments that need a lot of specialist knowledge and ongoing nurturing to keep running optimally, modern flash-based systems are much more 'hands-off' from an operational perspective as a lot of the automation smarts are designed to stay under the covers.

On the question of budget, the price differential compared to traditional disk-based systems has also come down considerably. Furthermore, any additional Capex requirements are typically more than offset by Opex reductions. Together with consumption-based pricing and evergreen-style subscriptions, achieving that step-change in efficiency, performance and user experience shouldn't be commercially prohibitive. The research data does suggest, however, that you might need to educate some stakeholders in order to secure their buy-in.

How do the following regard the idea of matching database technology with the right storage options?



With this in mind, we hope the insights presented in this report have helped you crystallize your thoughts around the database storage modernization opportunity, and how to make the case to relevant stakeholders.

About Freeform Dynamics

Freeform Dynamics is an IT industry analyst firm. Through our research and insights, we aim to help busy IT and business professionals get up to speed on the latest technology developments, and make better-informed investment decisions.

For more information, and access to our library of insights, commentary and free research, visit www.freeformdynamics.com.

About Pure Storage

Pure Storage empowers innovators to build a better world with data by delivering a simple, evergreen platform that enables organizations of all kinds to turn data into intelligence and advantage.

For more information, visit www.purestorage.com.

Terms of Use