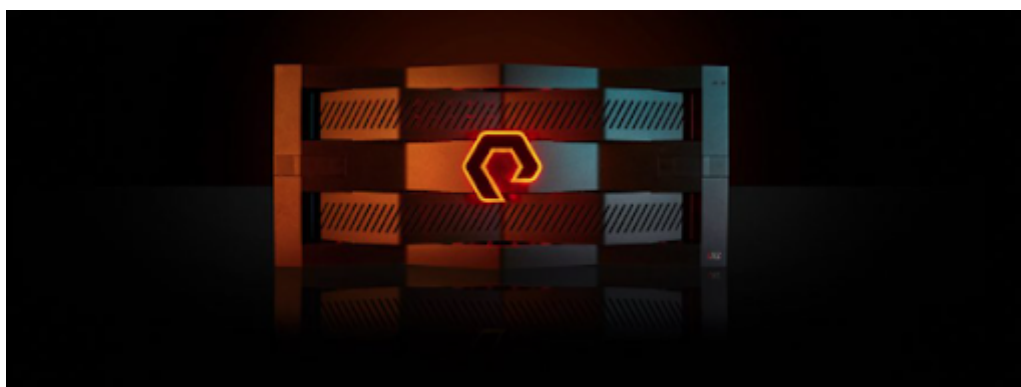


How to Consolidate Oracle Workloads with FlashArray//XL

As enterprises and governments accelerate their [digital transformation journeys](#), modern business applications place ever-increasing demands on if you consolidate [Oracle databases](#). In the face of this, it's physically impossible for infrastructure to remain as performant, scalable, and resilient without workload consolidation. The only way to stay ahead of the game is to do more with less. Enter Pure Storage®, which has enabled users to thread this needle through the [introduction](#) of our flagship [FlashArray//XL™](#).

The crown jewel of the FlashArray™ lineup, FlashArray//XL provides the [unmatched performance and scale](#) required to not only run massive Oracle databases but also consolidate Oracle workloads on fewer arrays.



But *Why* Should I Consolidate Oracle Workloads?

Organizations are increasingly choosing to consolidate their [Oracle](#) databases for a number of important reasons. The primary business driver is the reduction in the total cost of ownership (TCO) resulting from the optimization of Oracle licenses in terms of hardware, as well as human capital. In addition, consolidation streamlines management, increases security, and provides higher availability and better all-around performance.

Oracle database consolidation can be done in both the software and hardware realms:

- **Software consolidation:** Large organizations can have hundreds or thousands of applications each connecting to its own dedicated database on the back end. Using Oracle's multi-tenant architecture, multiple non-

CDB databases can be consolidated into a multitenant container database.

- **Hardware consolidation:** The challenge of hardware consolidation for Oracle databases mostly revolves around two areas—server consolidation and storage consolidation. Server consolidation is commonly done by virtualizing multiple database servers on fewer physical hosts. Similarly, storage consolidation can be achieved by moving an Oracle database's volumes to a bigger and faster FlashArray. By setting QoS limits on the bandwidth at the volume level, we can ensure that heavy IO load on the database doesn't negatively impact other databases running on the same array. FlashArray//XL is an ideal choice for consolidating the storage for multiple Oracle databases.

Why This FlashArray//XL Earns the Designation 'XL'

While architected on top of the foundation of the time-tested FlashArray platform, FlashArray//XL takes things to a whole new level with:

- **Increased performance**, with a larger 5U form factor to provide the necessary power and cooling required for today's higher-powered multi-core CPUs, as well as throughput up to 36 and latency as low as 150 µs.
- **Increased storage density**, to the tune of 20% more per rack unit thanks to 40 DirectFlash® modules with integrated NVRAM, which means you hit the trifecta of lower space requirements, power consumption, and cooling costs.
- **Increased I/O bandwidth**, with double the number of host I/O ports in each controller as compared to [FlashArray//X](#), and more expansion slots.
- **Increased resiliency**, with four power supplies operating in an N+2 redundancy configuration that can survive the failure of up to two power supplies.

As organizations [increasingly emphasize sustainability initiatives](#) to reduce the environmental impact of their data centers, Pure's ultra-dense, super-efficient flash reduces not only your power bill but also your greenhouse gas emissions. As detailed in our [inaugural ESG report](#), Pure is no Johnny-come-lately to the sustainability game—we've been efficient from the jump due to our seamless intertwining of software and hardware.

The FlashArray//XL family is made up of the FlashArray//XL130 and the FlashArray//XL170. With two DirectFlash expansion shelves, they scale to

maximum effective capacities of 3.53PB and 5.5PB in a dense 11U of rack space. For more details on the technical specifications of each model, please refer to the [FlashArray//XL data sheet](#).

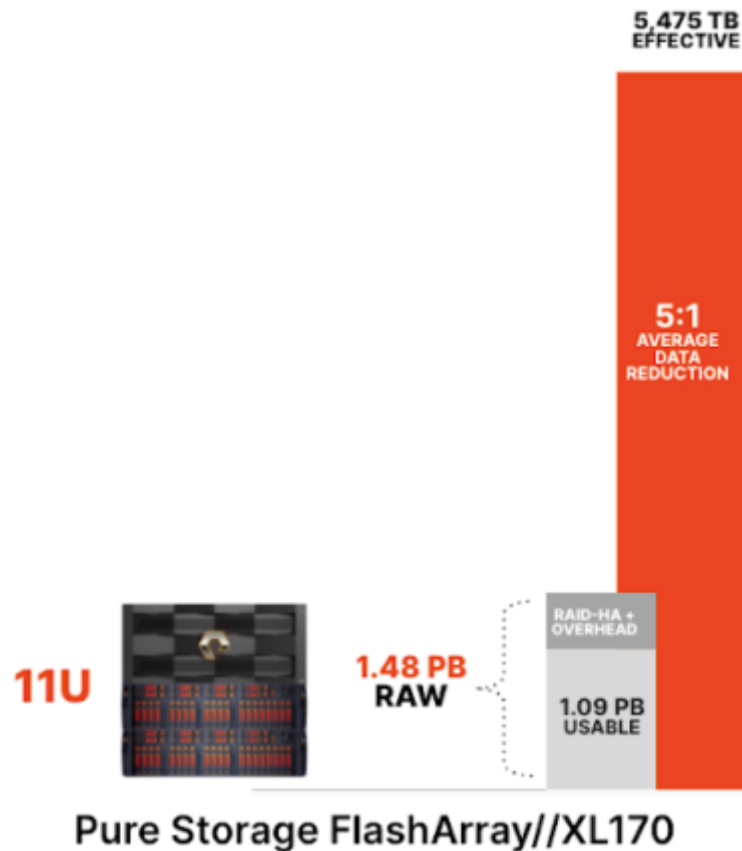


Figure 1:

Why is FlashArray//XL the Ideal Choice for Demanding Oracle Workloads?

Let's look at what makes FlashArray//XL ideal for Oracle workloads and applications.

Performance

Many of the world's largest organizations run their mission-critical production Oracle applications on FlashArray//X because of the performance, availability, efficiency, and simplicity it provides. Some of these customers have massive Oracle databases that need to support thousands of concurrent users. In such situations, the IO requirements of the database can exceed the capacity provided by a single FlashArray//X90.

The solution to this is to spread the volumes of an Oracle database across two or more FlashArray devices. There are some downsides to this approach, though. The

most obvious negative of this approach is that it will increase the complexity of management since database volumes now reside on multiple arrays. Automation scripts that were designed assuming all volumes of a database reside on a single array will no longer work. There's a bigger challenge, though. How does one decide which database volumes go on which array so that the IO remains evenly balanced and all arrays are optimally utilized? Analyzing data from the AWR repository will certainly help make this decision, but IO hotspots and patterns usually change over time.

Then, there's another challenge. One of the most important features of FlashArray is its ability to take near-instantaneous snapshots of its volumes. It's recommended to place all volumes of an Oracle database into a protection group so that all the volumes can be snapshotted as of a single point in time so that the database backup is crash consistent. If the database is spread across multiple arrays, it's not possible to take a crash-consistent backup because a protection group can't span multiple arrays. Two individual protection group snapshots could be defined (one per array), but the database would need to be put in hot backup mode and require media recovery to recover it.

FlashArray//XL takes performance to the next level and supports a higher number of users. It also provides higher IOPS. This enables it to run massive Oracle databases on a single array that previously would have needed to be distributed over two arrays. Depending on the application, a FlashArray//XL130 can provide up to 35% more IOPs than a FlashArray//X90, and a FlashArray//XL170 can provide up to 80% more IOPs than a FlashArray//X90.

FlashArray//XL Performance Tests

FlashArray//XL has replaced the FlashArray//X90 as the top-tier highest-performing FlashArray. We wanted to see FlashArray//XL in action, so we decided to run a series of performance tests on both the FlashArray//XL models—the FlashArray//XL130 and FlashArray//XL170—as well as on the latest generation of the FlashArray//X90—the FlashArray//X90R3. The objective of this exercise was to compare the performance of the FlashArray//XL170 and FlashArray//XL130 with that of the FlashArray//X90R3. For generating and simulating an Oracle workload, we decided to use two of the most popular load testing tools available today—the Silly Little Oracle Benchmark (SLOB) and the TPC-H toolkit.

Performance gain depends on many factors, including application design and the current bottlenecks in your system. I conducted these tests using the compute and networking resources that were available to me in my lab. Needless to say, your

mileage may vary.

Test Environment

I installed Oracle Database 19c Enterprise edition on eight servers running Red Hat 7.7 in non-CDB configuration. The servers were commodity x86 servers with two Intel Xeon CPU E5-2697 v2 at 2.70GHz with 12 cores each (totaling 24 cores) and 512GB of RAM.

The servers were connected to the Fibre Channel switch Cisco MDS 9148T using Emulex Corporation LPe32000 Fibre Channel Host Adapter. The servers were also connected to the Arista 7050SX switch using 25G network interface cards. Each path was redundant and multipathing was configured on each server.

Each server was zoned to the FlashArray//X90R3, FlashArray//XL130, and FlashArray//XL170 devices.

Oracle Performance Test with SLOB

SLOB is a widely used Oracle I/O workload generator and load testing tool. It's an excellent tool for testing storage I/O performance for Oracle databases even though it doesn't attempt to mimic a real-life application. It's free of all application-level contention and very effective in testing the I/O subsystem with random single block physical reads, as well as testing random single block writes.

Setting Up the Test

I installed Oracle Database 19c EE and then created three databases on each server. Each of the three databases was created using volumes from each of the three arrays. The parameter setting on each database was identical, which ensured that the exact same test was run against each of the three databases.

To maximize the I/O hitting the physical storage, the SGA was constrained to 5G and the PGA was set to 1G.

I wanted to have a mix of database storage options so I had configured five of the eight database servers to use ASM Filter Driver, and the remaining three to use the XFS file system.

SLOB configuration on each server:

- 128 schemas

- Each schema size was 16G
- Total size: 2T
- Test run time: 30 minutes

Test Execution

The load test was performed with each of the eight SLOB databases running 8,16, 32, and 128 users. The SLOB think time was set to 0, which means that each user session was continuously active on the database.

For each user count, tests were performed with the following read-write ratios:

- 100% reads
- 90% reads, 10% updates
- 80% reads, 20% updates
- 70% reads, 30% updates

Test Results

The test results were in line with our performance expectations. As a read/write ratio of 80/20 is a close estimation of OLTP workload, we'll use that test scenario to present the results. The following charts show IOPS improvement on the FlashArray//XL130 and FlashArray//XL170 over a FlashArray//X90R3 for read/write ratio of 80/20. The total database IOPS were calculated by aggregating the "physical read total IO requests" and "physical write total IO requests" values respectively from the "Instance Activity" section of the AWR report from all eight databases.

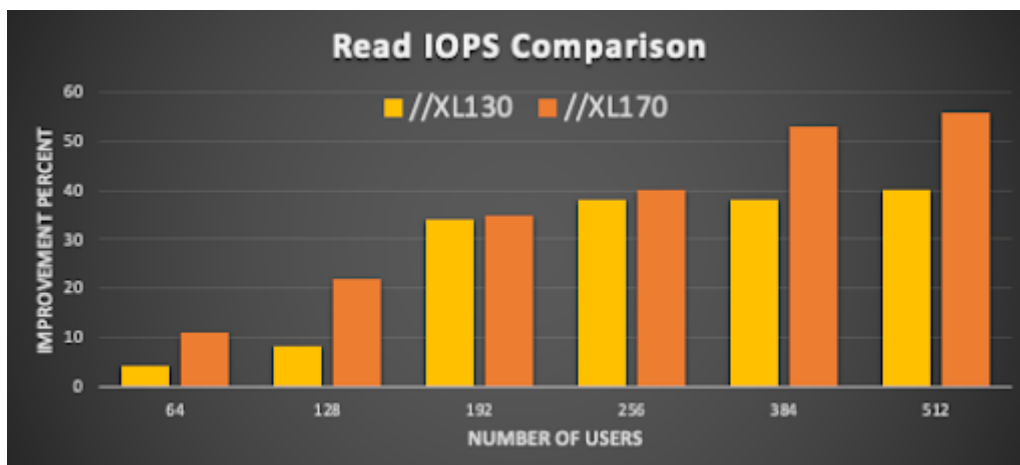


Figure 2: Read IOPS improvement over FlashArray//X90R3.

The next chart similarly shows the improvement in write IOPs:

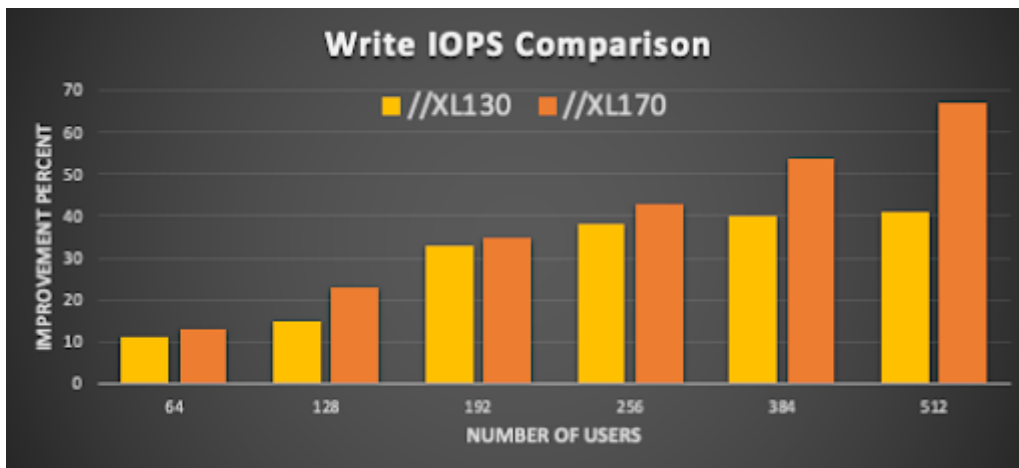


Figure 3: Write IOPS improvement over FlashArray//X90R3.

FlashArray//XL: Data Warehouse Workload Performance Tests

To compare the throughput performance of the three arrays, I used the TPC-H toolkit version 3.0.0 available from the [TPC website](#).

Setting Up the Test

For this test, I created a new database on each of the five database servers that I had previously set up with ASM FD. The **dbgen** utility from the toolkit was used to generate 1TB of data. The generated data is in the form of flat files. These flat files were then uploaded into the TPC-H schema in each database using the EXTERNAL tables feature.

Test Execution

The TPC-H toolkit provides a utility called **qgen** to generate data warehouse (DWH) type queries that can be run against the TPC-H database. It generates a set of 22 data warehousing queries that I bundled into a script. To simulate multiple data warehousing jobs running in parallel, I opened five terminal windows on each database host, thereby simulating the parallel execution of 25 DWH jobs. The Oracle parallel query degree of parallelism for each session was set to 16.

Test Results

While the tests were running, the IO was monitored at the OS level, as well as on

the FlashArray//XL dashboard. I observed that the bandwidth on the FlashArray//XL170 was 79% higher than what was achieved on the FlashArray//X90R3. Similarly, on the FlashArray//XL130, the bandwidth was 46% higher than the FlashArray//X90R3.

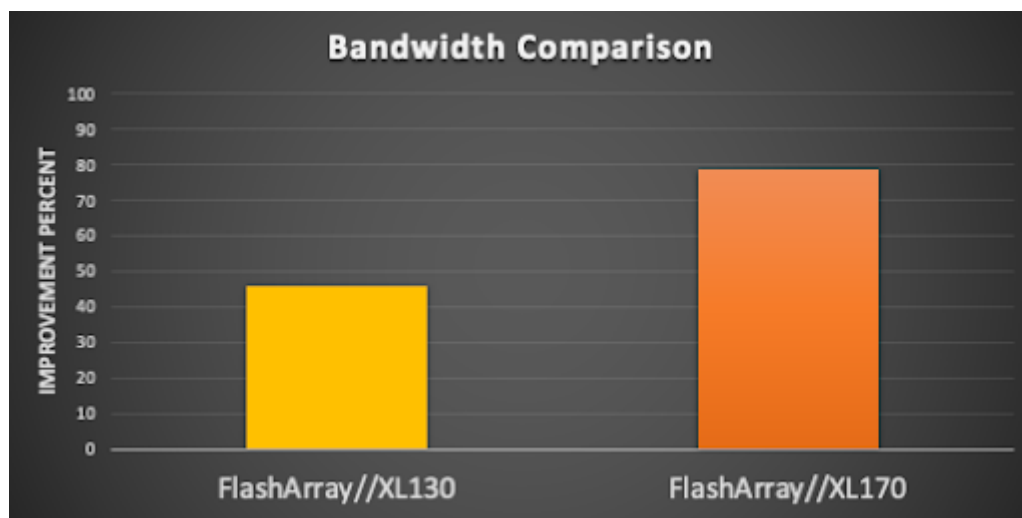


Figure 4: Data warehouse bandwidth improvement over FlashArray//X90R3.

In Conclusion

The newest member of the FlashArray family, FlashArray//XL, has been engineered from the ground up to run the most demanding mission-critical applications at scale. It comes with higher-power processors, increased bandwidth, double the number of host ports, denser storage, higher capacity, and improved resiliency to make it an ideal choice for consolidating Oracle workloads to fewer arrays.

Depending on the application, it's possible to achieve up to 80% improvement in Oracle database IOPS and bandwidth on a FlashArray//XL170 as compared to an FlashArray//X90R3. Similarly, improvements in the range of 45% were observed on the FlashArray//XL130.

Simplify and Scale Your Oracle Workloads with FlashArray//XL

Pure's approach with FlashArray//XL helps you simplify how your Oracle data is stored, mobilized, and protected. Consolidate and optimize your database operations, accelerate ransomware recovery, and gain data flexibility with a true, [as-a-service model](#) that's simple to manage and always evergreen.

More on Oracle:

- [Introducing Purity ActiveDR™ for Oracle Disaster Recovery](#)
- [A Modern Data Experience for Oracle Databases](#)
- [Docker, Oracle 12c and Persistent Storage](#)
- [Online resizing of Oracle volumes on Pure FlashArray](#)
- [Using Oracle's vdbench tool to create a ~5.5:1 reducible dataset on a Pure Storage FlashArray](#)
- [Oracle 12c Online Database Move Datafile](#)
- [ActiveCluster for Oracle RAC on Extended Distance Cluster](#)
- [Cloning Oracle an Database on ASM with Pure FlashArray](#)