STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

Can SPDK Deliver High Performance NVMe on Windows?

Nick Connolly, Chief Scientist, MayaData / DataCore Software

Agenda

- Background
- Getting Started
- Windows Platform Development Kit
- Upstream Changes
- Current Status
- Lessons Learnt
- Getting Involved

Background

A Quick Overview

NVMe (Non-volatile Memory Express)

- Low-overhead storage protocol
- Replacement for SATA/SCSI
- Limited command set for efficiency
- Command and Response Queues
- Multiple independent I/O queue pairs
- Enables lock free parallelism
- Supports large queue depths

NVMe over Fabrics (NVMe-oF)

- Provides a connection to remote NVMe device
- Using a Transport Protocol
  - NVMe/FC – NVMe over Fibre Channel
  - NVMe/IB – NVMe over InfiniBand
  - RoCE – NVMe using RDMA
  - NVMe/TCP – NVMe over TCP
- Aim is less than 10 microseconds of additional latency

NVMe on Windows

- Windows has built in support for NVMe disks
  - Using StorNVMe.sys
- Individual vendors may offer their own driver
  - e.g. Intel Optane
- Support for directly connected disks
  - NVMe drives on the PCIe bus
- Kernel drivers
  - I/O involves crossing a protection barrier

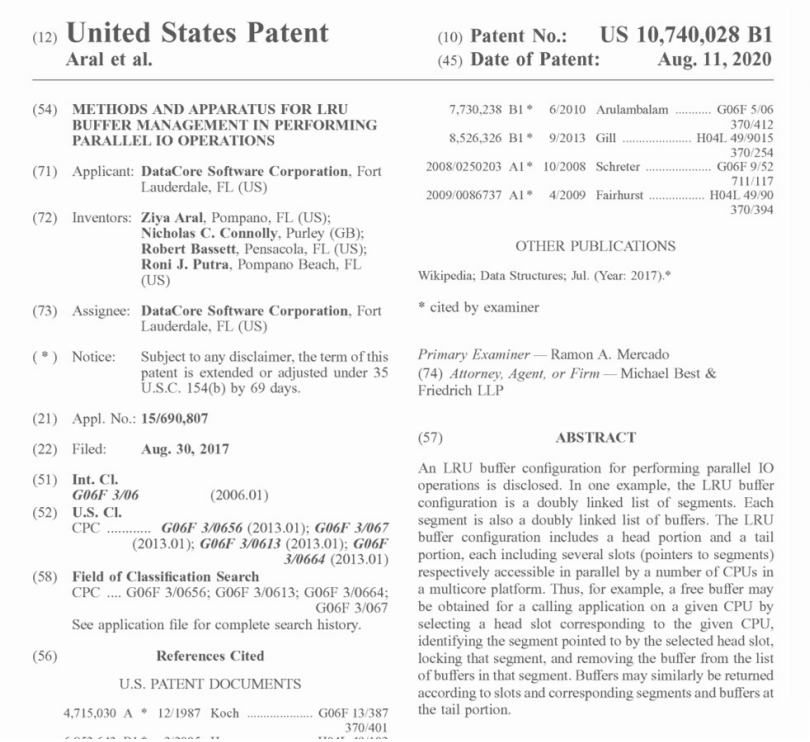
NVMe-oF on Windows

- No native support for NVMe-oF in Windows
- Broadcom supports NVMe/FC
  - Emulex Fibre Channel adaptors
- Marvell supports NVMe/FC
  - QLogic Fibre Channel adaptors
- NVIDIA do not support NVMe-oF on Windows
  - No driver for Mellanox adaptors
- StarWind support NVMe/TCP and RoCE v2
  - 100% software NVMe initiator

DATACORE™

https://www.datacore.com

- High-performance, low-latency storage stack
- Innovative multi-threading techniques, 100+ cores



(12) **United States Patent**
Aral et al.

(10) **Patent No.: US 10,740,028 B1**
(45) **Date of Patent: Aug. 11, 2020**

(54) **METHODS AND APPARATUS FOR LRU BUFFER MANAGEMENT IN PERFORMING PARALLEL IO OPERATIONS**

(71) Applicant: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(72) Inventors: **Ziya Aral**, Pompano, FL (US); **Nicholas C. Connolly**, Purley (GB); **Robert Bassett**, Pensacola, FL (US); **Roni J. Putra**, Pompano Beach, FL (US)

(73) Assignee: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 69 days.

(21) Appl. No.: **15/690,807**

(22) Filed: **Aug. 30, 2017**

(51) **Int. Cl.**
G06F 3/06 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 3/0656*** (2013.01); ***G06F 3/067*** (2013.01); ***G06F 3/0613*** (2013.01); ***G06F 3/0664*** (2013.01)

(58) **Field of Classification Search**
CPC G06F 3/0656; G06F 3/0613; G06F 3/0664; G06F 3/067
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,715,030 A * 12/1987 Koch G06F 13/387 370/401
7,730,238 B1 * 6/2010 Arulambalam G06F 5/06 370/412
8,526,326 B1 * 9/2013 Gill H04L 49/9015 370/254
2008/0250203 A1 * 10/2008 Schreter G06F 9/52 711/117
2009/0086737 A1 * 4/2009 Fairhurst H04L 49/90 370/394

OTHER PUBLICATIONS

Wikipedia; Data Structures; Jul. (Year: 2017).*

* cited by examiner

Primary Examiner — Ramon A. Mercado
(74) *Attorney, Agent, or Firm* — Michael Best & Friedrich LLP

(57) **ABSTRACT**

An LRU buffer configuration for performing parallel IO operations is disclosed. In one example, the LRU buffer configuration is a doubly linked list of segments. Each segment is also a doubly linked list of buffers. The LRU buffer configuration includes a head portion and a tail portion, each including several slots (pointers to segments) respectively accessible in parallel by a number of CPUs in a multicore platform. Thus, for example, a free buffer may be obtained for a calling application on a given CPU by selecting a head slot corresponding to the given CPU, identifying the segment pointed to by the selected head slot, locking that segment, and removing the buffer from the list of buffers in that segment. Buffers may similarly be returned according to slots and corresponding segments and buffers at the tail portion.

(12) **United States Patent**
Connolly et al.

(10) **Patent No.: US 10,013,283 B1**
(45) **Date of Patent: Jul. 3, 2018**

(54) **METHODS AND APPARATUS FOR DATA REQUEST SCHEDULING IN PERFORMING PARALLEL IO OPERATIONS**

(71) Applicant: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(72) Inventors: **Nicholas C. Connolly**, Purley (GB); **Robert Bassett**, Pensacola, FL (US); **Ziya Aral**, Pompano Beach, FL (US); **Roni J. Putra**, Pompano Beach, FL (US)

(73) Assignee: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 31 days.

(21) Appl. No.: **15/236,902**

(22) Filed: **Aug. 15, 2016**

(51) **Int. Cl.**
G06F 12/00 (2006.01)
G06F 9/48 (2006.01)
G06F 12/0888 (2016.01)
G06F 3/06 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 9/4881*** (2013.01); ***G06F 3/0619*** (2013.01); ***G06F 3/0665*** (2013.01); ***G06F 3/0689*** (2013.01); ***G06F 12/0888*** (2013.01);

(58) **Field of Classification Search**
CPC G06F 9/4881; G06F 3/0619; G06F 3/0665; G06F 3/0689; G06F 12/0888; G06F 2212/6046
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2006/0282689 A1 * 12/2006 Tipley G06F 1/3203 713/300

* cited by examiner

Primary Examiner — Shawn X Gu
(74) *Attorney, Agent, or Firm* — Michael Best & Friedrich LLP

(57) **ABSTRACT**

Methods and apparatus for data request scheduling in performing parallel IO operations are disclosed. In one example, IO requests directed to an operating system having an IO scheduling component are processed. There, an IO request directed from an application to the operating system is intercepted. A determination is made whether the IO request is subject to immediate processing using available parallel processing resources. When it is determined that the IO request is subject to immediate processing using the available parallel processing resources, the IO scheduling component of the operating system is bypassed. The IO request is directly and immediately processed and passed back to the application using the available parallel processing resources.

(12) **United States Patent**
Aral et al.

(10) **Patent No.: US 10,599,477 B1**
(45) **Date of Patent: *Mar. 24, 2020**

(54) **METHODS AND APPARATUS FOR COMMAND LIST PROCESSING IN PERFORMING PARALLEL IO OPERATIONS**

(71) Applicant: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(72) Inventors: **Ziya Aral**, Fort Lauderdale, FL (US); **Nicholas C. Connolly**, Purley (GB); **Robert Bassett**, Pensacola, FL (US); **Roni J. Putra**, Pompano Beach, FL (US)

(73) Assignee: **DataCore Software Corporation**, Fort Lauderdale, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/395,638**

(22) Filed: **Apr. 26, 2019**

Related U.S. Application Data

(63) Continuation of application No. 15/601,319, filed on May 22, 2017, now Pat. No. 10,318,354.

(51) **Int. Cl.**
G06F 9/48 (2006.01)
G06F 3/06 (2006.01)
G06F 9/50 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 9/5038*** (2013.01); ***G06F 9/5005***

(58) **Field of Classification Search**
CPC G06F 9/48; G06F 3/0601
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,465,335 A 11/1995 Anderson
10,318,354 B1 * 6/2019 Aral G06F 9/5038
2006/0179274 A1 8/2006 Jones et al.
2011/0072211 A1 3/2011 Duluk, Jr. et al.
2012/0180068 A1 7/2012 Wein et al.
2013/0179486 A1 7/2013 Lee et al.
2014/0123146 A1 5/2014 Barrow-Williams et al.

* cited by examiner

Primary Examiner — David E Martinez
(74) *Attorney, Agent, or Firm* — Michael Best & Friedrich LLP

(57) **ABSTRACT**

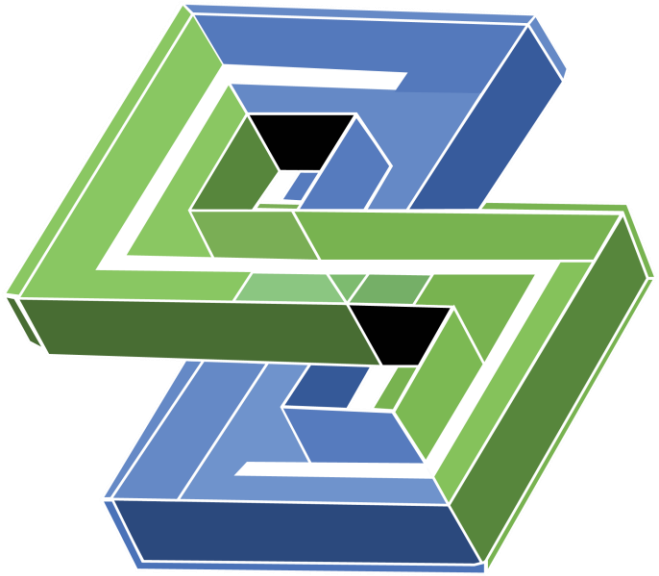
Command list processing in performing parallel IO operations is disclosed. In one example, handling IO requests directed to an operating system having an IO scheduling component entails allocating a command to a thread in association with an IO request. The command is allocated from one of a plurality of command lists accessible in parallel, and the command is also linked to one of a plurality of active command lists that are accessible in parallel. The command lists can be arranged as per-CPU command lists, with each per-CPU command list corresponding to one of a plurality of CPUs on a multi-core processing platform on which the IO requests are processed. Similarly, each of the active command lists can respectively correspond to one of the plurality of CPUs on the multi-core processing platform. Per-volume queues can also be implemented for respective volumes presented to applications.

SPDK

https://www.spdk.io

Storage Performance Development Kit

- Tools and libraries for writing:
  - High performance, scalable
  - User-mode storage applications
- Cutting Edge
  - Leverage the latest NVMe features
  - Poll-mode and event-loop for maximum performance
  - Lockless, thread-per-core design
- Production Ready
- Open Source (BSD 3-Clause)



MayaData

https://www.mayadata.io

- Leads the development of OpenEBS
  - CNCF Sandbox Project
  - Most popular open source storage for Kubernetes
- Based on SPDK
  - New generation high-performance storage stack
  - Designed from the ground up to be Cloud Native
  - MayaData Enterprise Edition
- But what about SPDK for Windows?

OpenEBS

Getting Started

Where do we begin?

Getting Started

More than just a few missing pieces!

- SPDK assumes a POSIX platform
- Builds require make and a shell environment
- Platform specific functionality
- Dependency projects with own build processes
- Environment library not sufficient for Windows

- Data Plane Development Kit
- Libraries to accelerate packet processing workloads
- Runs mostly in user space on a variety of CPU architectures

SPDK uses it for:

- Memory Management
- Access to PCIe devices

DPDK on Windows

- Established community working on Windows
  - Contributions from Microsoft, Intel, NVIDIA and others
- DPDK v19.05 added initial limited support

Friendly and approachable:

- Bi-weekly community meetings
- Excellent DPDK Summit 2020
- Discussion with a key Windows maintainer

Build Environment

- One of the biggest challenges
  - Toolset (e.g., GCC, Clang, Visual Studio)
  - Shell and packages
- Initially
  - GCC (mingw-w64) and Clang
  - MSYS2
- Current recommendation
  - Windows Subsystem for Linux (type 1)
  - Cross compilation with GCC (mingw-w64)

Design Goals

- Not a fork
  - Upstream the changes into SPDK
- Work with SPDK community
- Minimal changes to SPDK code
  - The smaller and more localized the better

SPDK Community

- Very positive interactions with SPDK community
  - First exchanges with Ben Walker and Jim Harris
  - Invited to talk to community meeting about the project
  - Code reviews
  - Ongoing exchanges with the core maintainers
- Supportive, inclusive, friendly and responsive
- Effective community meetings, active Slack channel
- If you have the opportunity – get involved!

Windows Platform Development Kit

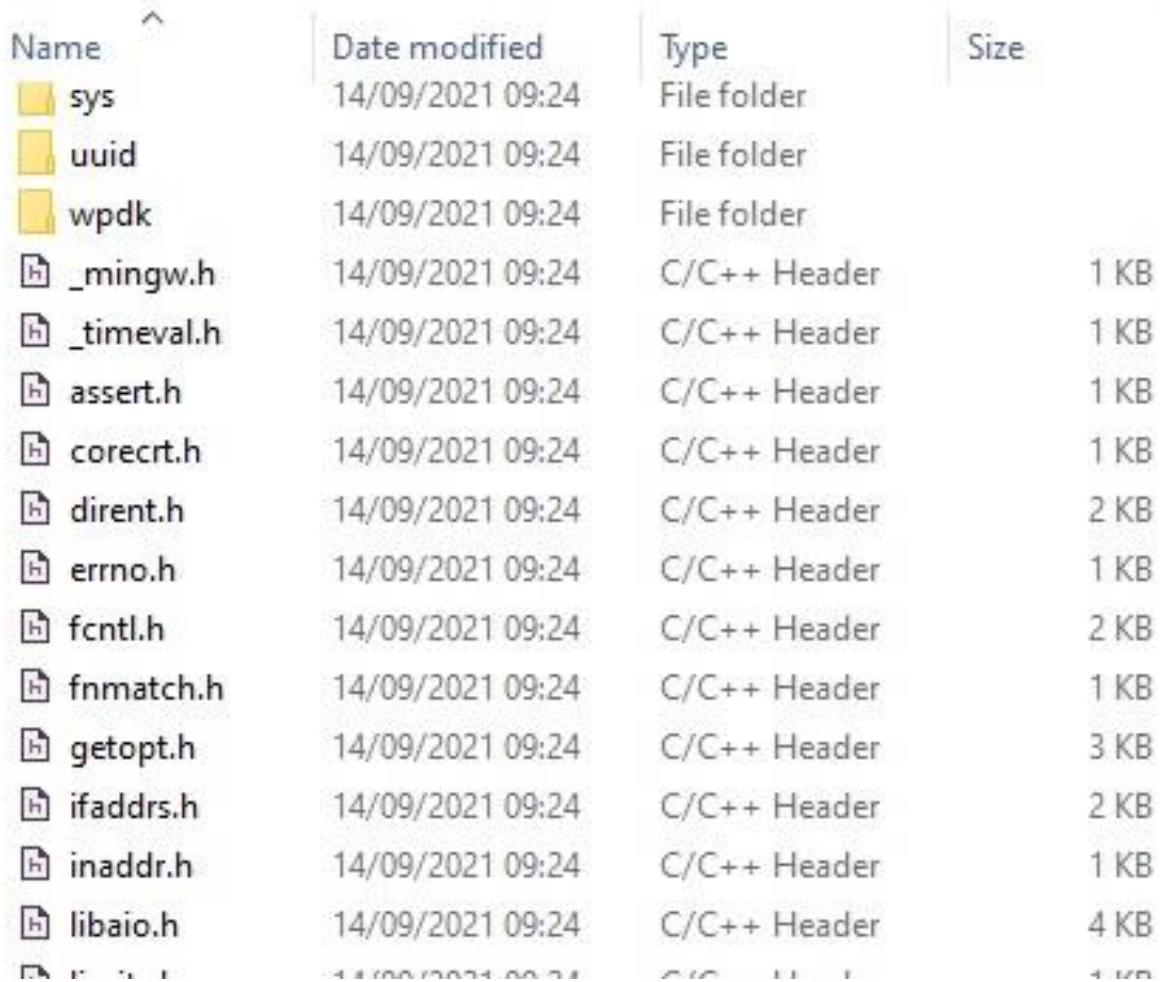
Making it a reality!

Windows Platform Development Kit (WPDK)

- Simple POSIX emulation layer for SPDK's needs
- Production quality
- Native Windows executables
- No surprises
- Independently testable
- Includes dependencies
- Permissive license (BSD 3-Clause)
- https://wpdk.github.io

WPDK - Include Files

- Missing headers



Name	Date modified	Type	Size
sys	14/09/2021 09:24	File folder	
uuid	14/09/2021 09:24	File folder	
wpdk	14/09/2021 09:24	File folder	
_mingw.h	14/09/2021 09:24	C/C++ Header	1 KB
_timeval.h	14/09/2021 09:24	C/C++ Header	1 KB
assert.h	14/09/2021 09:24	C/C++ Header	1 KB
corecrt.h	14/09/2021 09:24	C/C++ Header	1 KB
dirent.h	14/09/2021 09:24	C/C++ Header	2 KB
errno.h	14/09/2021 09:24	C/C++ Header	1 KB
fcntl.h	14/09/2021 09:24	C/C++ Header	2 KB
fnmatch.h	14/09/2021 09:24	C/C++ Header	1 KB
getopt.h	14/09/2021 09:24	C/C++ Header	3 KB
ifaddrs.h	14/09/2021 09:24	C/C++ Header	2 KB
inaddr.h	14/09/2021 09:24	C/C++ Header	1 KB
libaio.h	14/09/2021 09:24	C/C++ Header	4 KB

WPDK - Include Files

- Missing headers
- Missing definitions

```
#ifndef _WPDK_LIMITS_H_
#define _WPDK_LIMITS_H_

#ifndef PATH_MAX
#define PATH_MAX 260
#endif

#ifndef NAME_MAX
#define NAME_MAX 256
#endif

#ifndef SSIZE_MAX
#define SSIZE_MAX _I64_MAX
#endif
```


WPDK - Include Files

- Missing headers
- Missing definitions
- Compiler differences

```
#if !defined(__MINGW32__) || !defined(_INC_STRING_S)
#if defined(__USE_GNU)
#define strerror_s(buf,len,err) wpdk_strerror_r_gnu(err,bu
#else
#define strerror_s(buf,len,err) wpdk_strerror_r(err,buf,le
#endif
#endif
```


WPDK - Include Files

- Missing headers
- Missing definitions
- Compiler differences
- Function wrapping

```
int wpdk_getifaddrs(struct ifaddrs **ifap);
void wpdk_freeifaddrs(struct ifaddrs *ifa);

#ifndef _WPDK_BUILD_LIB_
#define getifaddrs(ifap) wpdk_getifaddrs(ifap)
#define freeifaddrs(ifa) wpdk_freeifaddrs(ifa)
#endif
```


WPDK – Error Handling

- Invalid parameter handling
- Error code mapping

```
case ERROR_ACCESS_DENIED:
        /* Access is denied */
        return EACCES;

case ERROR_ADAP_HDW_ERR:
        /* A network adapter hardware error occurred */
        return EIO;

case ERROR_ALERTED:
        /* Alerted */
        return EINTR;
```


WPDK - Threads

- No built-in libpthread on Windows
- Some external packages are available
- Uses simple wrapper around Windows primitives
  - Threads
  - Synchronization: Mutex, SpinLock, Barrier, Condition Variable
  - Thread specific values
  - Affinity
  - Thread name

WPDK – Memory Allocation

- posix_memalign – aligned allocations
- Free must work with aligned and unaligned
- Requires WPDK wrapper around malloc / calloc / free

```
#if !defined(__MINGW32__) || !defined(_INC_STDLIB_S)
#define calloc(nelem,elsize) wpdk_calloc(nelem, elsize)
#define free wpdk_free
#define malloc(size) wpdk_malloc(size)
#define realloc(ptr,size) wpdk_realloc(ptr,size)
#endif
#endif
```


WPDK - Sockets

- Superficially close, but significant differences
- Returns SOCKET, not integer file descriptor
- Requires WPDK wrapper around read / write / close
- Missing readv / writev
  - Implemented using WSARecv / WSASend
- Differences in behaviour (socket options, non-blocking mode)
- AF_UNIX is supported between WSL 1 and Windows

WPDK - Signals

- Windows signal handling is limited
  - SIGABRT, SIGFPE, SIGILL, SIGINT, SIGSEGV, SIGTERM
- Signals can’t be sent to other processes
- Killing a process doesn’t invoke SIGTERM handler

- Added event-based signal worker thread
- wpdk_kill to send a signal from the shell
- Allows for graceful shutdown of processes

WPDK – Other Functionality

- Missing POSIX functionality
  - dirent, gettimeofday, mmap, poll, select
- Missing libraries
  - uuid, crypto (MD5)
- Linux functionality
  - getifaddrs, epoll, libaio
- Mocking for unit tests (GCC –wrap)
- Pathname mapping to Windows

Upstream Changes

Into SPDK and DPDK

SPDK Changes

- Integer sizes
  - Linux uses LP64 (32-bit int, 64-bit long and pointer)
  - Windows uses LLP64 (32-bit int and long, 64-bit pointer)
  - A pointer won't fit in a long on Windows (use intptr_t)
  - Must print 64-bit values with PRI[udx]64 for portability
  - Some unit tests made assumptions about arithmetic limits

- Mutex initialization
  - POSIX requires mutex initialization
  - Some unit tests relied on Linux not enforcing it

SPDK Changes

- Bit-field packing
  - Bit field packing when basic types are different
  - include/nvme_spec.h for NVMe definitions
- Fine tuning of some #ifdef's
  - #ifdef __FreeBSD__ → #ifndef __linux__
- Minor adjustments to build scripts
  - Support for building with WPDK

DPDK Changes

- Very few changes required
- Build related changes
  - To fix warnings or settings
- Mapping a PCIe NVMe disk
  - Extend netuio.sys to recognize NVMe disks
  - Add PCIe class detection for NVMe disks

Current Status

Are we nearly there yet?

Current State

- The project is currently at an alpha stage
- NVMe initiator and target run
  - NVMe/TCP target can serve storage
  - Drive I/O to a physical NVMe disk
- Unit tested
  - All of the SPDK tests pass
  - Majority of WPDK functionality is unit tested

Conclusion

Can SPDK Deliver High Performance NVMe on Windows?

Even in alpha, yes it can!

Lessons Learnt

A beginner’s guide to open source

Lessons Learnt

- Work with the community
  - Find out where they are heading and get involved
  - Seek to add value rather than fork projects
- Be honest about your agenda
- Get to know people if you can
  - Attend the community meetings
- Be patient
- Be respectful - remember you are a guest!
  - Don’t dominate discussions to pursue your own agenda

Getting Involved

What, me?

Getting Involved

- Contributions are welcome and needed!
  - Head to the WPDK documentation to get started
  - https://wpdk.github.io
- Please join the SPDK community (https://spdk.io/community)
  - Tell us how you are using SPDK on Windows
  - For real-time discussions, Slack has a #windows channel
- Happy Experimenting!

Thank You!

- Thank you for listening!
- Please ask questions in Slack
- Thank you to MayaData for their support and encouragement of this project

Please take a moment to rate this session.

Your feedback is important to us.

