ddn

THE AI DATA COMPANY

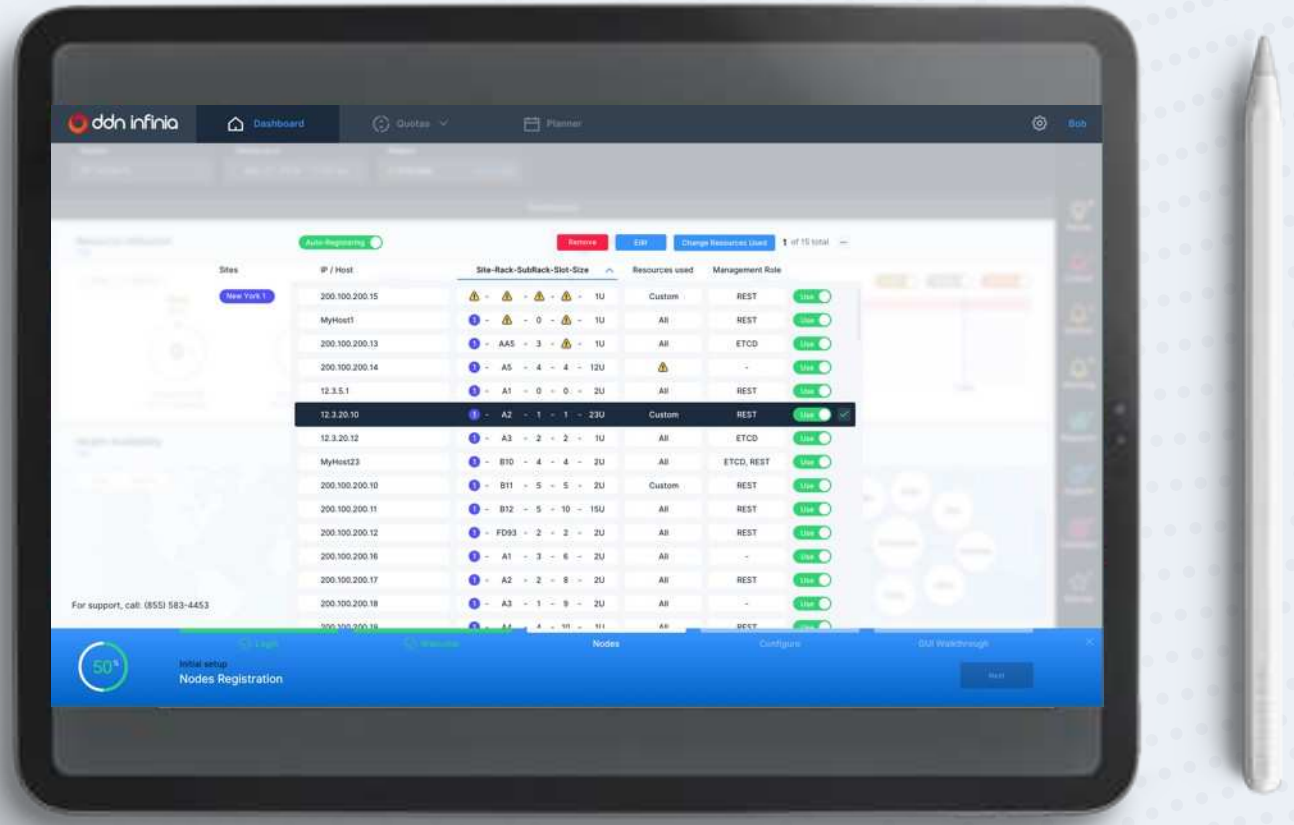
DDN Infinia

Scalable Data Solution for Any Scale AI

Scale Out in the Data Center and in Cloud Up to 100's of PBs

Auto-Installation, **10 minutes to Deploy**

Deploy new services and tenants in just 4 minutes

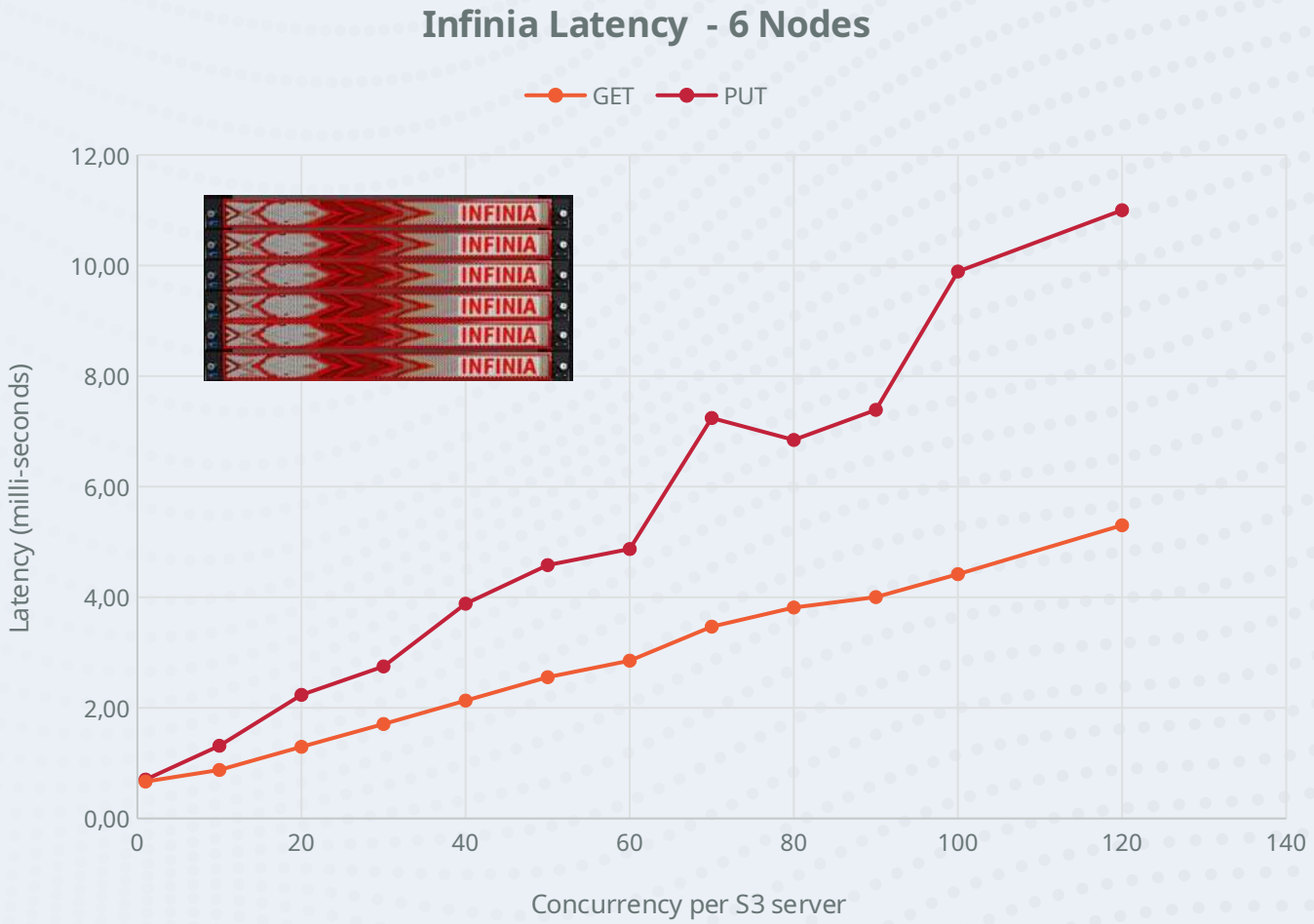


ddn

Accelerate AI Workloads with 100x Lower Latencies

Infinia is the first Object Storage to deliver **sub-millisecond latencies** for PUTs and GETs

- >300K GET Operations per second
- Improve website load times, database response times, queries etc improving end user satisfaction
- Accelerate Enterprise Analytics workloads like Apache Spark, Starburst Presto/Trino, Clickhouse



Multi-Cloud Ready Kubernetes Storage

The complete multi-cloud ready Kubernetes storage platform with elastic scalability, unmatched availability, and self-service access to any storage.

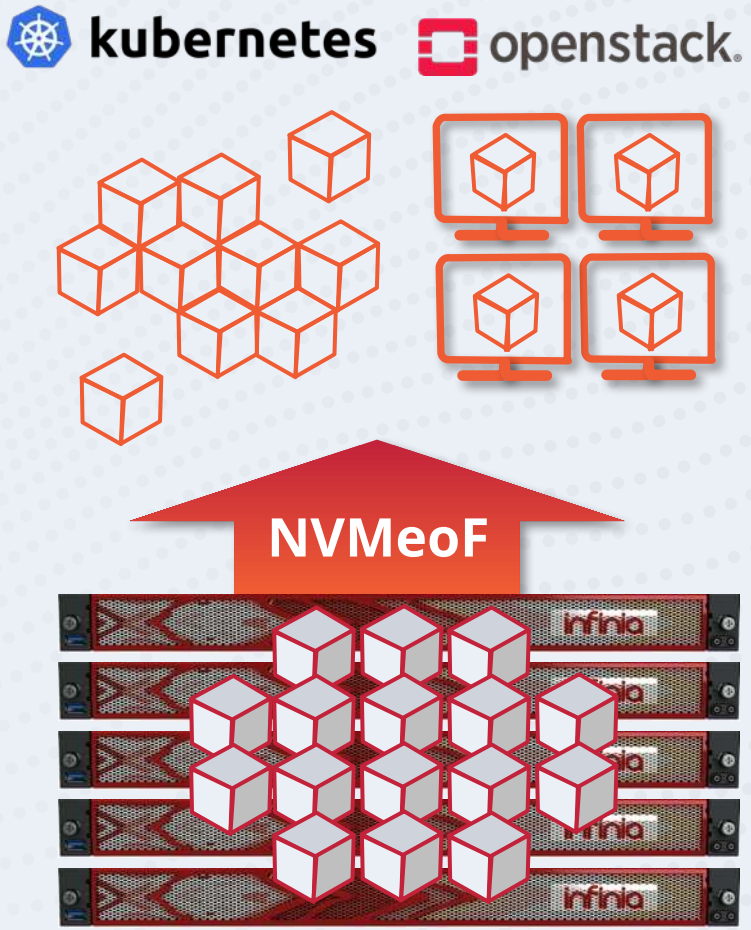
- Scalable persistent storage
- Multi-cloud Data Mobility
- Zero RPO Disaster Recovery

DDN Infinia provides Low Latency, High Bandwidth Block Exports via NVMeoF TCP(1.0) / RoCE(1.2) for Kubernetes and Openstack

CSI Drivers 1.0

Openstack Cinder Drivers 1.1

Single NVMe-oF Client Read over 15 GB/s



ddn

S3 Compatibility details ~80% passed

testsuite	s3tests_boto3		
category	(Multiple Items)		
Count of result_0711	**Column Labels**		
Row Labels	**FAILED**	**PASSED**	**Grand Total**
acl	15	5	20
bucket_ops	3	28	31
bucket_policy	17		17
bucket_policy_status	7	2	9
ceph_utils		1	1
copy_ops		4	4
general	14	32	46
getput	2	26	28
headers	12	36	48
list bucket	28	55	83
mpu	10	14	24
objectcopy	2	15	17
post	19	16	35
presigned	7	12	19
versioning	6	16	22
Grand Total	**142**	**262**	**404**

7/10/2024

testsuite	s3tests_boto3		
category	(Multiple Items)		
Count of testcase	**Column Labels**		
Row Labels	**FAILED**	**PASSED**	**Grand Total**
acl	15	5	20
bucket_ops	3	28	31
bucket_policy	4	13	17
bucket_policy_status	7	2	9
ceph_utils		1	1
copy_ops		4	4
general	11	36	47
getput	3	25	28
headers	12	35	47
list bucket	7	76	83
mpu	8	16	24
objectcopy	2	15	17
post	1	34	35
presigned	6	13	19
versioning	4	18	22
Grand Total	**83**	**321**	**404**

8/8/2024

New libfuse maintainers since spring 2024

Nikolaus Rath busy with other projects

- New maintainers
  - Bernd Schubert
  - Ashley Pittman
  - Antonio SJ Musumeci

My FUSE work

- Thread creation/destruction until libfuse-3.12
- Libc ABI
- Atomic open (will be taken over by a colleague)
- Fuse-over-iouring
- Libfuse maintainer since March 2024
- Forking through posix_spawn (completing the work from Matthias Goergens)
- DLM integration – notifications into both directions
  - TBD

Libfuse-3.12 max_threads API change

Performance issue with max_idle_threads (>= libfuse-3.2) – expensive thread creation and destruction

- max_idle_thread deprecated, new max_threads parameter should be used
- Initial struct fuse_loop_config hard to extend, no reserved fields → API change required
- Entire new API with FUSE_USE_VERSION >= 312 - struct fuse_loop_config opaque/private, with getters and setters

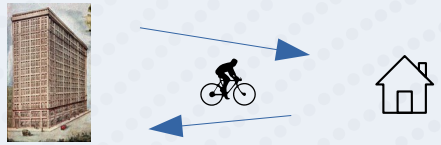
```
/* creates config with libfuse defaults */
struct fue_loop_config *loop_config = fuse_loop_cfg_create();

fuse_loop_cfg_set_max_threads(config, get_nprocs_conf());
fuse_loop_cfg_set_clone_fd(loop_config, 1);
int ret = fuse_session_loop_mt(se, loop_config);
```

FUSE-over-IO-URING

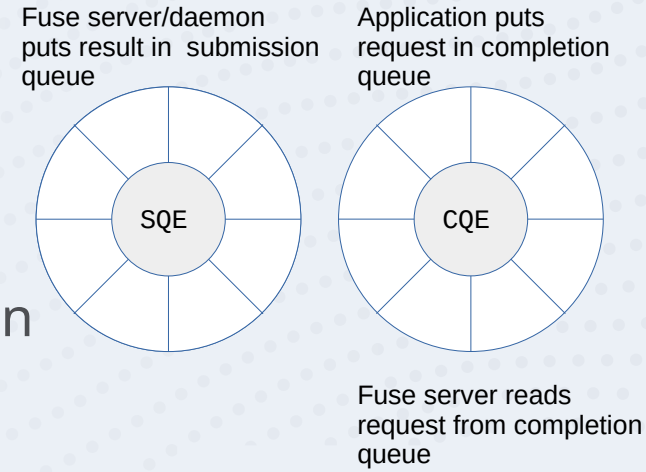
Goal: Performance!

- Reduction of kernel/user-space transitions
- NUMA awareness and core affinity
- No or very limited changes for FUSE-server



- Use of IORING_OP_URING_CMD
  - Commit result and fetch next in one kernel/userspace transition
  - Fuse over /dev/fuse:
    - Fetch request with read()
    - Submit result with write()
- Async requests
  - Full power of io-uring – multiple requests without transition

“Reversed” IO-URING



Libfuse-3.14.1 ABI breakage

Spotting by code review in March this year by Ashley, one year after the release of that version. Issue is in *struct fuse_file_info*

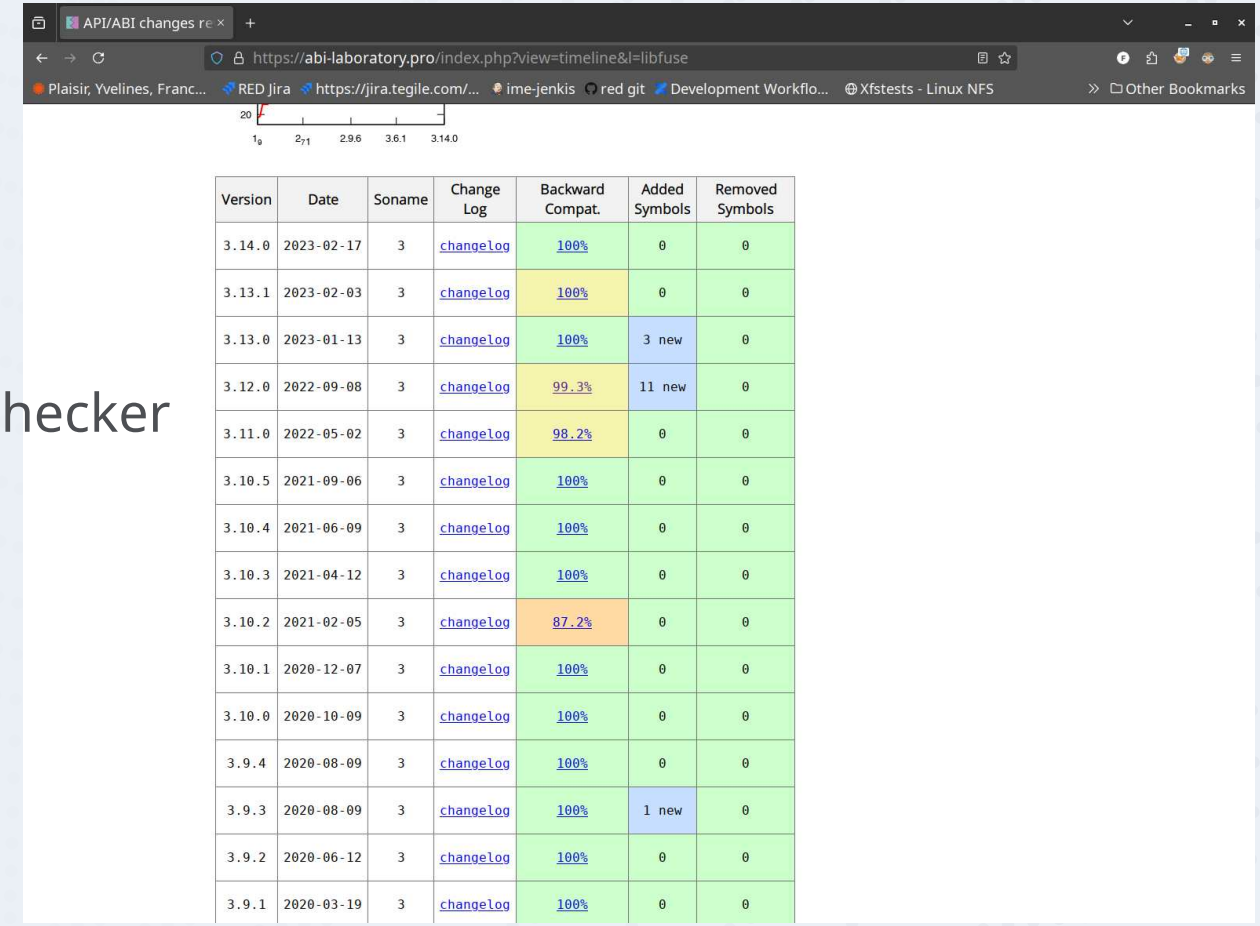
```
struct fuse_file_info {
…
unsigned int keep_cache : 1;
unsigned int parallel_direct_writes : 1; ====> new field
unsigned int flush : 1;
unsigned int nonseekable : 1;
unsigned int flock_release : 1;
unsigned int cache_readdir : 1;
unsigned int noflush : 1;

=====> should have been here


unsigned int padding : 23;
```

Libfuse-3.14.1 ABI breakage - continued

- Reason for the delay of 3.17
- Heuristics patch – needs testing
- Libfuse issue 1029
- New versions will be tested by https://github.com/lvc/abi-compliance-checker
- struct fuse_config also affected, actually also already in 3.10.2
- Report for 3.12 false positive?
- 3.17 supposed to be out by next week



Version	Date	Soname	Change Log	Backward Compat.	Added Symbols	Removed Symbols
3.14.0	2023-02-17	3	changelog	100%	0	0
3.13.1	2023-02-03	3	changelog	100%	0	0
3.13.0	2023-01-13	3	changelog	100%	3 new	0
3.12.0	2022-09-08	3	changelog	99.3%	11 new	0
3.11.0	2022-05-02	3	changelog	98.2%	0	0
3.10.5	2021-09-06	3	changelog	100%	0	0
3.10.4	2021-06-09	3	changelog	100%	0	0
3.10.3	2021-04-12	3	changelog	100%	0	0
3.10.2	2021-02-05	3	changelog	87.2%	0	0
3.10.1	2020-12-07	3	changelog	100%	0	0
3.10.0	2020-10-09	3	changelog	100%	0	0
3.9.4	2020-08-09	3	changelog	100%	0	0
3.9.3	2020-08-09	3	changelog	100%	1 new	0
3.9.2	2020-06-12	3	changelog	100%	0	0
3.9.1	2020-03-19	3	changelog	100%	0	0

Post 3.17 attribute timeout API change?

Kernel:

```
struct fuse_attr_out {
	uint64_t attr_valid; /* Cache timeout for the attributes */
	uint32_t attr_valid_nsec;
	uint32_t dummy;
	struct fuse_attr attr;
};
```

libfuse: double in *struct fuse_entry_param*

- Draft:
  struct fuse_entry_param2
  fuse_reply_entry2()
  fuse_reply_create2()
  fuse_reply_attr2()
  fuse_add_direntry_plus2()

  https://github.com/libfuse/libfuse/pull/1032

Data alignment

- Buffer layout for op FUSE_WRITE

```
struct fuse_in_header;
struct fuse_write_in;
uint8_t payload[];
```

  - Note: No issue with splice instead of read()/write() of /dev/fuse)

- Buffer layout for FUSE_SETXATTR

```
struct fuse_in_header;
char attribute_name[];
uint8_t attribute_payload[];
```

(Also see <libfuse>/doc/libfuse-operations.txt>)

open+getattr

sshfs report:
https://github.com/libfuse/libfuse/issues/945
https://github.com/libfuse/sshfs/issues/302

- Attribute cache used and cache IO, i.e. not FOPEN_DIRECT_IO
  - Reads only to the known file size
- Patch from Joanne Koong to invalidate attributes on open
  - Two calls – transition between kernel and userspace and hard to handle by server in one request – needs caching within the server
- Patch from me to allow open+getattr in one request
  - Update need to switch to NFS4 like compounds

Upcoming DLM notifications

- For sync (metadata and direct-io) mostly in fuse server
  - But page invalidation for remote clients
- Cached reads
  - Page invalidation
- Cached writes
  - Invalidation of remote clients before data have reached fuse server
  - Mmap!


ddn
THE AI DATA COMPANY
Thank you
www.ddn.com