LA-UR-16-26609
Approved for public release; distribution is unlimited.

Title:	Parallel File System I/O Performance Testing On LANL Clusters
Author(s):	Wiens, Isaac Christian Green, Jennifer Kathleen
Intended for:	Presentation to HPC Division
Issued:	2016-08-30

Parallel File System I/O Performance Testing On LANL Clusters

Isaac Wiens, U.G.S
Jennifer Green, R&D Scientist
HPC-ENV
Programming & Runtime Environments Team
August 18, 2016
LA-UR #######

UNCLASSIFIED

Introduction

- I/O is a known bottleneck for HPC applications
- Performance optimization of I/O is often required
- This Summer project entailed integrating IOR under Pavilion and automating the results analysis

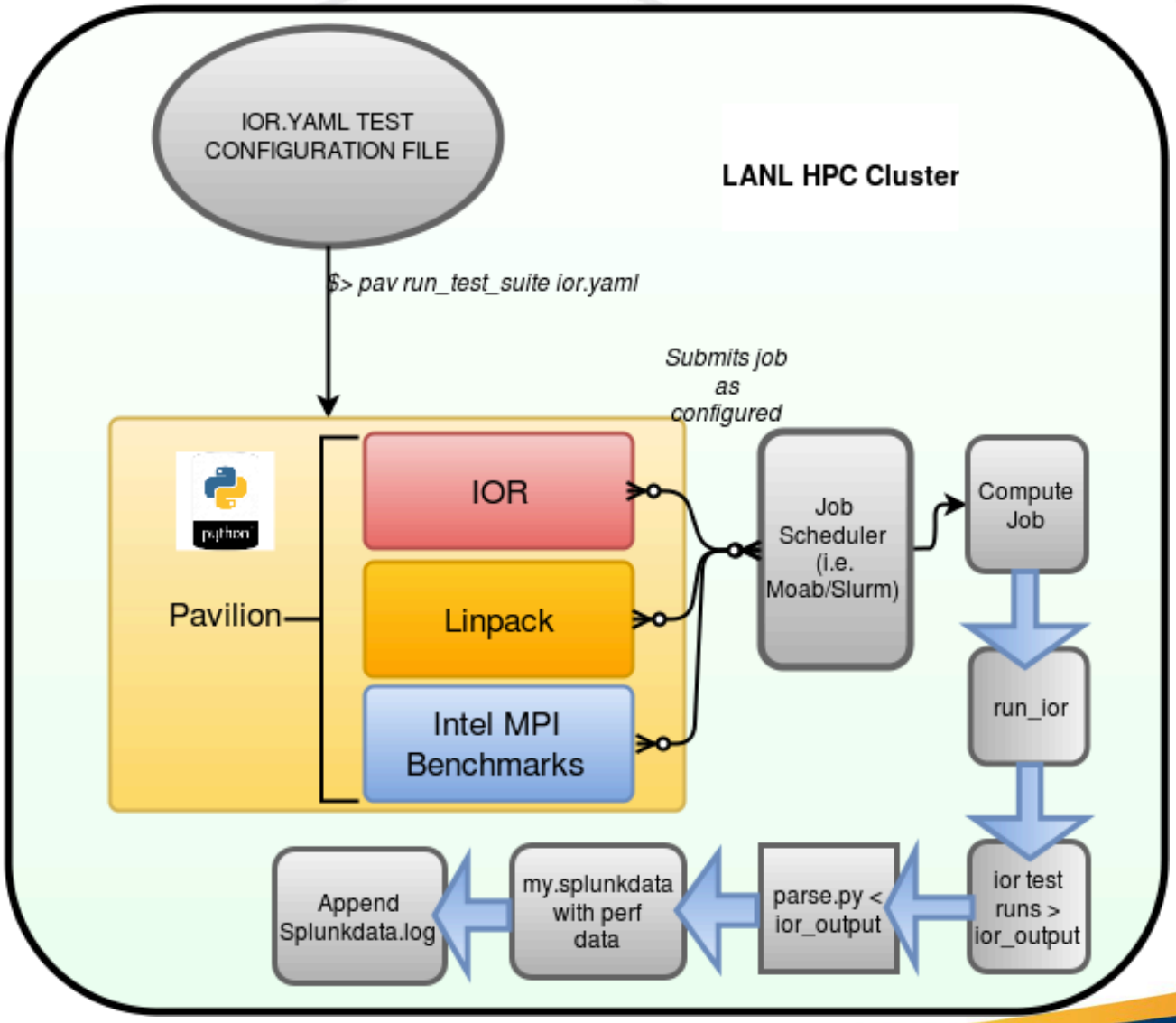
Scope of Work

- Automate Build of IOR File-system Performance Benchmark
- Write a runscript to parse Pavilion test arguments and pass to IOR test at runtime
- Write a parser for the results → Splunk
- Graph results of tests in Splunk

Tools

- **Pavilion** – HPC python test harness
- **IOR** – NERSC parallel file-system benchmark
- ***build_ior** – script to automate environment setup, build and linking of IOR on LANL high performance computers
- ***run_ior** – script that extracts YAML file input parameters from Pavilion and populates the arguments passed to IOR
- ***parse.py** – output parser to construct an event in splunk data key-value pair logfile
- **Splunk Dashboard** – data graphing tool interfaced with the test results data

IOR-Pavilion Test Workflow



Build Script

Pavilion has configuration options to permit the "build" code step for every test instance, or manually, one can generate the executable and use it for all tests.

Sources Modulefiles init

Clean installation from Source

Error Checking

Load Env. Modulefile for MPI

Make files handle compile + linking

Change access for shared resource

```
1  #!/bin/tcsh
2
3  source /usr/share/Modules/init/tcsh
4  #source a module command initialization script
5  if ( -d IOR ) then
6     rm -Rf IOR
7  else if ( -e IOR-2.10.3.tgz ) then
8     tar xzf IOR-2.10.3.tgz
9  else
10    echo 'file not found'
11    exit 2
12 endif
13
14
15 cd IOR
16
17 module load openmpi
18
19 gmake clean
20 gmake mpiio
21
22 cd ../
23 chgrp -Rf hpctools IOR #modify later for hpcsoft
24 chmod -Rf g+rwX IOR
25 chmod -f g+x IOR/src/C/IOR
26
```

IOR Parameters

Flag	Key	YAML key=value Usage	Argument Description
-N	testNum	"deprecated"	test number for reference in some output
-a S	api	api=MPIIO	API for I/O [POSIX|MPIIO|HDF5|NCMPI]
-b N	blockSize	blockSize=1g	contiguous bytes to write per task (e.g.: 8, 4k, 2m, 1g)
-B	useO_DIRECT	useO_DIRECT=True	uses O_DIRECT for POSIX, bypassing I/O buffers
-c	collective	collective=True	collective I/O
-C	reorderTasks	reorderTasks=True	changes task ordering to n+1 ordering for readback
-Q N	taskPerNodeOffset	taskPerNodeOffset=4	for read tests use with -C & -Z options (-C constant N, -Z at least N)
-Z	reorderTasksRandom	reorderTasksRandom=True	changes task ordering to random ordering for readback
-X N	reorderTasksRandomSeed	reorderTasksRandomSeed=1234	random seed for -Z option
-d N	interTestDelay	interTestDelay=14	delay between reps in seconds
-D N	deadlineForStonewalling	deadlineForStonewalling=60	seconds before stopping write or read phase
-Y	fsyncPerWrite	fsyncPerWrite=True	perform fsync after each POSIX write
-e	fsync	fsync=True	perform fsync upon POSIX write close
-E	useExistingTestFile	useExistingTestFile=True	do not remove test file before write access
-f S	scriptFile	scriptFile=script_name	test script name
-F	filePerProc	filePerProc=True	file-per-process
-g	intraTestBarriers	intraTestBarriers=True	use barriers between open, write/read, and close
-G N	intraTestBarriers	setTimeStampSignature=1	set value for time stamp signature
-H	showHints	showHints=True	show hints
-i N	repetitions	repetitions=100	number of repetitions of test
-I	individualDataSets	individualDataSets=True	datasets not shared by all procs [not working]
-j N	outlierThreshold	outlierThreshold=35	warn on outlier N seconds from mean
-J N	setAlignment	setAlignment=4k	HDF5 alignment in bytes (e.g.: 8, 4k, 2m, 1g)
-k	keepFile	keepFile=True	don't remove the test file(s) on program exit
-K	keepFileWithError	keepFileWithError=True	keep error-filled file(s) after data-checking
-l	storeFileOffset	storeFileOffset=True	use file offset as stored signature
-m	multiFile	multiFile=True	use number of reps (-i) for multiple file count
-n	noFill	noFill=True	no fill in HDF5 file creation
-N N	numTasks	numTasks=32	number of tasks that should participate in the test
-o S	testFile	testFile=myiortestfile.out	full name for test
-O S	IORdirectives	IORDirectives="checkRead=1"	e.g. -O checkRead=1,lustreStripeCount=32
-p	preallocate	preallocate=True	preallocate file size
-P	useSharedFilePointer	useSharedFilePointer=True	use shared file pointer [not working]
-q	quitOnError	quitOnError=True	during file error-checking, abort on error
-r	readFile	readFile=True	read existing file
-R	checkRead	checkRead=True	check read after read
-s N	segmentCount	segmentCount=32	number of segments
-S	useStridedDatatype	useStridedDatatype=True	put strided access into datatype [not working]
-t N	transferSize	transferSize=8k	size of transfer in bytes (e.g.: 8, 4k, 2m, 1g)
-T N	maxTimeDuration	maxTimeDuration=30	max time in minutes to run tests
-u	uniqueDir	uniqueDir=True	use unique directory name for each file-per-process
-U S	HintsFileName	hintsFileName=myhints.txt	full name for hints file
-v	verbose	verbose=True	output information (repeating flag increases level)
-V	useFileView	useFileView=True	use MPI_File_set_view
-w	writeFile	writeFile=True	write file
-W	checkWrite	checkWrite=True	check read after write
-x	singleXferAttempt	singleXferAttempt=True	do not retry transfer if incomplete
-z	randomOffset	randomOffset=True	access is to random, not sequential, offsets within a file

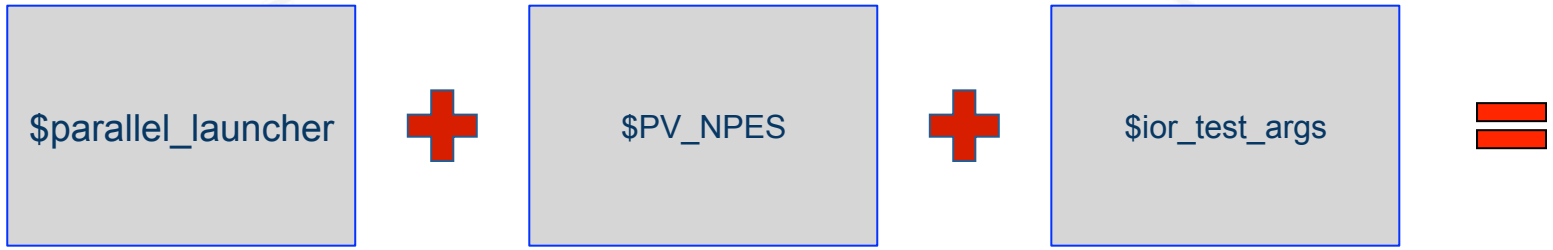
How Parameters Are Passed To IOR

```
ior1:
  name: ior
  source_location: '/usr/projects/hpctools/PAV_PRETEAM/test_src/ior'
  run:
    cmd: 'new_run_ior'
    scheduler: moab
    test_args: ["api=MPIIO segmentCount=4 testFile=/net/scratch1/jgreen/mynewfile.out transferSize=2M blockSize=80M repetitions=10",
                "api=MPIIO segmentCount=4 testFile=/net/scratch2/jgreen/mynewfile.out transferSize=2M blockSize=80M repetitions=10"]

  moab:
    num_nodes:  [ 2 ]
    procs_per_node: 12
    time_limit: "01:00:00"
    account: "hpctest"
  splunk:
    global_data_file: '/usr/projects/splunk/results/iwiens/lightshow/sp
```

```
if ($?PV_TEST_ARGS) then
    set ior_test_args=()
    set i=1
    @ length = $#PV_TEST_ARGS + 1
    while ( $i < $length )

        echo "Setting ${PV_TEST_ARGS[$i]} iteration $i"
        switch ( ${PV_TEST_ARGS[$i]} )
        ##  -a S  api --  API for I/O [POSIX|MPIIO|HDF5|NCMPI]
        case api=*:
            set api=`echo ${PV_TEST_ARGS[$i]} | sed 's/api=//g'`
            echo $api
            set apiArg=" -a $api"
            set ior_test_args=( $ior_test_args $apiArg )
        breaksw
        ##  -B    useO_DIRECT -- uses O_DIRECT for POSIX, bypassing I/O buffers
        case useO_DIRECT=True:
            set useO_DIRECTArg=" -B"
            set ior_test_args=( $ior_test_args $useO_DIRECTArg )
        breaksw
        ##  -b N  blockSize -- contiguous bytes to write per task  (e.g.: 8, 4k, 2m, 1g)
        case blockSize=*:
            set blockSize=`echo ${PV_TEST_ARGS[$i]} | sed 's/blockSize=//g'`
            echo $blockSize
            set blockSizeArg=" -b $blockSize"
            set ior_test_args=( $ior_test_args $blockSizeArg )
        breaksw
        ##  -c    collective -- collective I/O
        case collective=True:
            set collectiveArg=" -c"
            set ior_test_args=( $ior_test_args $collectiveArg )
        breaksw
        ##  -C    reorderTasks -- changes task ordering to n+1 ordering for readback
        case reorderTasks=True:
            set reorderTasksArg=" -C"
```

- While loop iterates $i while $i < $#PV_TEST_ARGS
- Case switches in c-shell test each element of the list against cases that expect a key=*
- A match will set the argument and will strip the key= from the element's value
- $ior_test_args is appended
- Switch is broken and the next element is processed

*run_ior: Functionality




```
iwiens@lightshow:TURQUOISE -> mpirun -n $NPES ./IOR.x -a POSIX -s 4 -o
\ /net/scratch1/iwiens/mynewfile.out_18456 -t 2M -b 8M -i 1
```

Python IOR-Output Parser

Gathers Test Environment from Pavilion

Calls HPCSoft utilities for system values

Creates a header string

Iterates over the lines of ior_output

Reformats Time Stamps

Regular Expressions parse each line

Pass / Fail Criteria based on "Operation" existing in the output file

```
42 npes = os.environ.get('PV_NPES')
43 nnodes = os.environ.get('PV_NNODES')
44 jobid = os.environ.get('PV_JOBID')
45 nodes = os.environ.get('PV_NODES')
46 pespernode = os.environ.get('PV_PESPERNODE')
47 testname = os.environ.get('PV_TESTNAME')
48 username = os.environ.get('USER')
49 machine = hpcsoft("/usr/projects/hpcsoft/utilities/bin/sys_name")
50 os = hpcsoft("/usr/projects/hpcsoft/utilities/bin/sys_os")
51 kv_header = "Machine=" + machine + " Os=" + os + " TestName=" + testname + " NNodes=" + nnodes + " NPES=" + npes + " PESPERNODE=
   " + pespernode + " JobID=" + jobid + " NodeList=" + nodes
52 #print kv_header
53
54 for line in open("ior_output", 'r'):
55    if re.search("Run began:", line):
56       key = 'start'
57       line = line.strip()
58       value = datetime.strptime(line.split("Run began: ")[1].rstrip().lstrip(), "%c").strftime('%Y-%m-%d'" "'%H:%M:%S')
59       kv_header = ( kv_header + " " + key + "=" + value + " " )
60    if re.search("Run finished:", line):
61       key = 'end'
62       line = line.strip()
63       value = datetime.strptime(line.split("Run finished: ")[1].rstrip().lstrip(), "%c").strftime('%Y-%m-%d'" "'%H:%M:%S')
64       kv_header = ( kv_header + " " + key + "=" + value + " " )
65    if re.search("=", line):
66       key = line.split("=")[0]
67       value = line.replace(key, "").lstrip().rstrip().replace(" ", "_").lstrip("=_")
68       key = key.rstrip().lstrip().replace(" ", "_")
69
70       if re.search("[\d+]_[a-zA-Z]+", value):
71          value1 = value.split("_")[0]
72          value2 = value.split("_")[1:]
73          value3 = ''
74          for x in value2:
75            value3= value3 + x.strip("()")
76       if value1:
77          value = value1
78          metric = value3
79          kv_constructor = ( key + "=" + value + " " + key + "metric" + "=" + metric + " " + kv_constructor )
80       else:
81          kv_constructor = ( key + "=" + value + " " + kv_constructor )
82
83    if re.search("Operation", line):
84       passfail = "PASSED"
85       line = line.strip()
86       header_list = line.replace('  ',",").split(",")
87
```

Python IOR-Output Parser (continued)

```
Reported: 2 (out of 2) daemons - 24 (out of 24) procs
IOR-2.10.3: MPI Coordinated Test of Parallel I/O

Run began: Sat Jul 23 20:32:28 2016
Command line used: ./IOR.x -a POSIX -s 4 -o /net/scratch1/jgreen/mynewfile.out_18456 -t 2M -b 8M -i 1
Machine: Linux ls019.localdomain

Summary:
        api               = POSIX
        test filename     = /net/scratch1/jgreen/mynewfile.out_18456
        access            = single-shared-file
        ordering in a file = sequential offsets
        ordering inter file= no tasks offsets
        clients           = 24 (12 per node)
        repetitions       = 1
        xfersize          = 2 MiB
        blocksize         = 8 MiB
        aggregate filesize = 768 MiB

Operation  Max (MiB)  Min (MiB)  Mean (MiB)   Std Dev  Max (OPs)  Min (OPs)  Mean (OPs)   Std Dev  Mean (s)
---------  ---------  ---------  ----------   -------  ---------  ---------  ----------   -------  --------
write         58.27      58.27       58.27      0.00       7.28       7.28        7.28      0.00  13.18071   EXCEL
read        1808.16    1808.16     1808.16      0.00     226.02     226.02      226.02      0.00   0.42474   EXCEL

Max Write: 58.27 MiB/sec (61.10 MB/sec)
Max Read:  1808.16 MiB/sec (1895.99 MB/sec)

Run finished: Sat Jul 23 20:32:42 2016
```

```
if re.search("=", line):
    key = line.split("=")[0]
    value = line.replace(key, "").lstrip().rstrip().replace(" ", "_").lstrip("=_")
    key = key.rstrip().lstrip().replace(" ", "_").replace("(", "_").replace(")", "")
    regexScratch = re.compile(r"\/\w+\/\w+(\d)\/.*")
```

The regular expression captures the "Test Summary" portion of the ior_output file, and includes this information in the kv_header string

```
123   if re.search("Operation", line):
124       passfail = "PASSED"
125       line = line.strip()
126       header_list = line.replace('   ',",").split(",")
```

The performance metrics summary in the ior_output file is iterated over (more detail on the next slide), hinging on the column header values detected for every run. This will make the test results' parsing mechanism work for any number of columns that may be present in ior_output files.

Python IOR-Output Parser (continued)

operations = [“read”, “write”]

```
128  for ops in operations:
129     if re.search(ops, line):
130        resultslist = line.strip()
131        resultslist = resultslist.split()
132        i = 0
133        while ( i < len(header_list) ):
134           results = ops + header_list[i].replace(" ","").replace("(","_").replace(")","")+"="+resultslist[i].replace("(","_").replace(")","")+" "
135           i += 1
136           if kv_constructor:
137              kv_constructor = ( results + kv_constructor )
138           else:
139              kv_constructor = ( results )
```

Python IOR-Output Parser (continued)

```
Reported: 2 (out of 2) daemons - 24 (out of 24) procs

IOR-2.10.3: MPI Coordinated Test of Parallel I/O

Run began: Sat Jul 23 20:32:28 2016
Command line used: ./IOR.x -a POSIX -s 4 -o /net/scratch1/jgreen/mynewfile.out_18456 -t 2M -b 8M -i 1
Machine: Linux ls019.localdomain

Summary:
        api                = POSIX
        test filename      = /net/scratch1/jgreen/mynewfile.out_18456
        access             = single-shared-file
        ordering in a file = sequential offsets
        ordering inter file= no tasks offsets
        clients            = 24 (24 per node)
        repetitions        = 1
        xfersize           = 2 MiB
        blocksize          = 8 MiB
        aggregate filesize = 768 MiB

Operation  Max (MiB)  Min (MiB)  Mean (MiB)   Std Dev  Max (OPs)  Min (OPs)  Mean (OPs)   Std Dev  Mean (s)
---------  ---------  ---------  ----------   -------  ---------  ---------  ----------   -------  --------
write         58.27      58.27       58.27      0.00       7.28       7.28        7.28      0.00  13.18071   EXCEL
read        1808.16    1808.16     1808.16      0.00     226.02     226.02      226.02      0.00   0.42474   EXCEL

Max Write: 58.27 MiB/sec (61.10 MB/sec)
Max Read:  1808.16 MiB/sec (1895.99 MB/sec)

Run finished: Sat Jul 23 20:32:42 2016
```

```
for ops in operations:
    if re.search(ops, line):
        resultslist = line.strip()
        resultslist = resultslist.split()
        i = 0
        while ( i < len(header_list) ):
            results = ops + header_list[i].replace(" ","").replace("(","_").replace(")","")+"="+resultslist[i].replace("(","_").replace(")","")+" "
            i += 1
            if kv_constructor:
                kv_constructor = ( results + kv_constructor )
            else:
                kv_constructor = ( results )
```

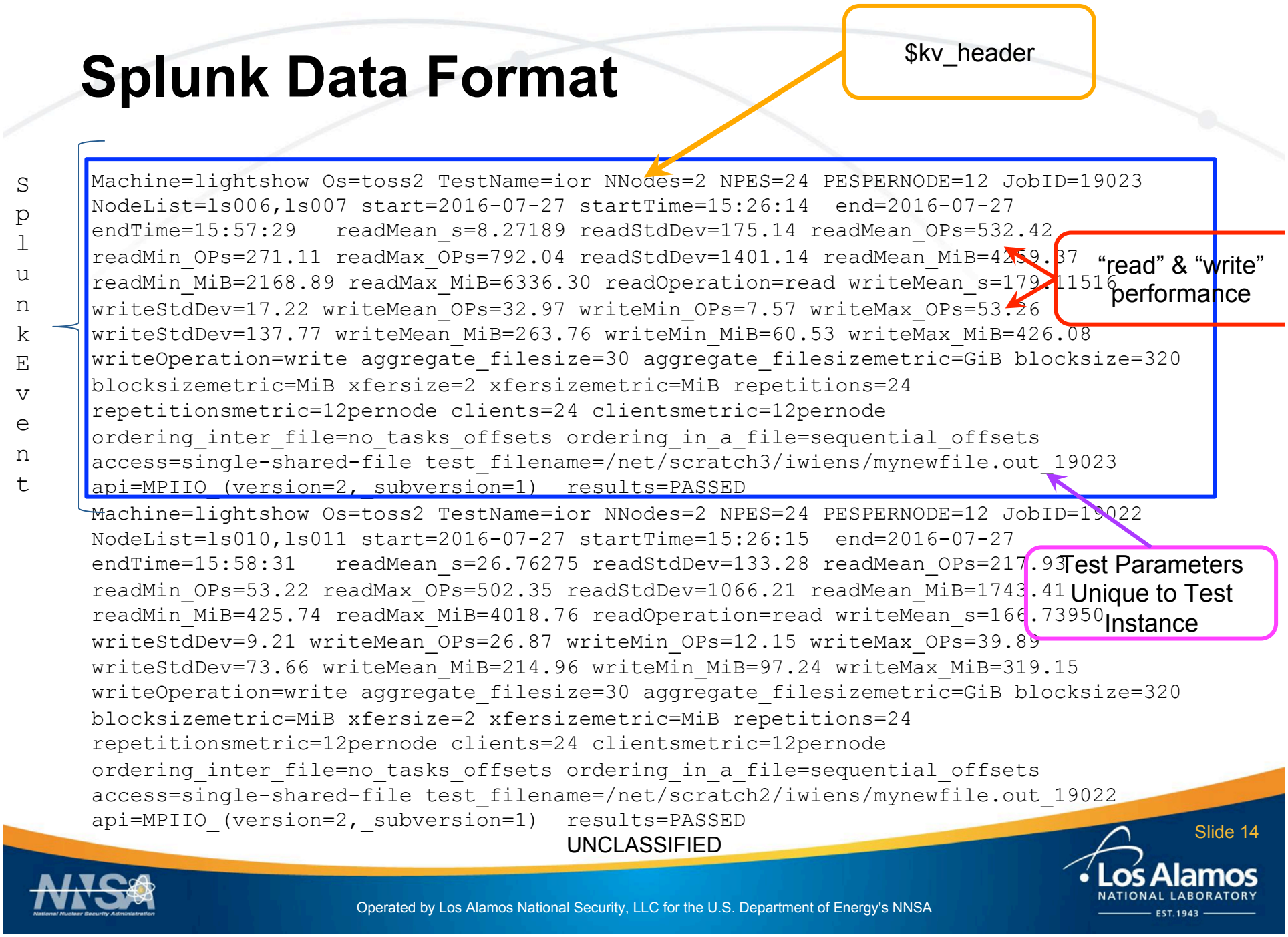
results = “write” + ”Max_MiB” + “=” + “58.27”

ops + header_list[$i]

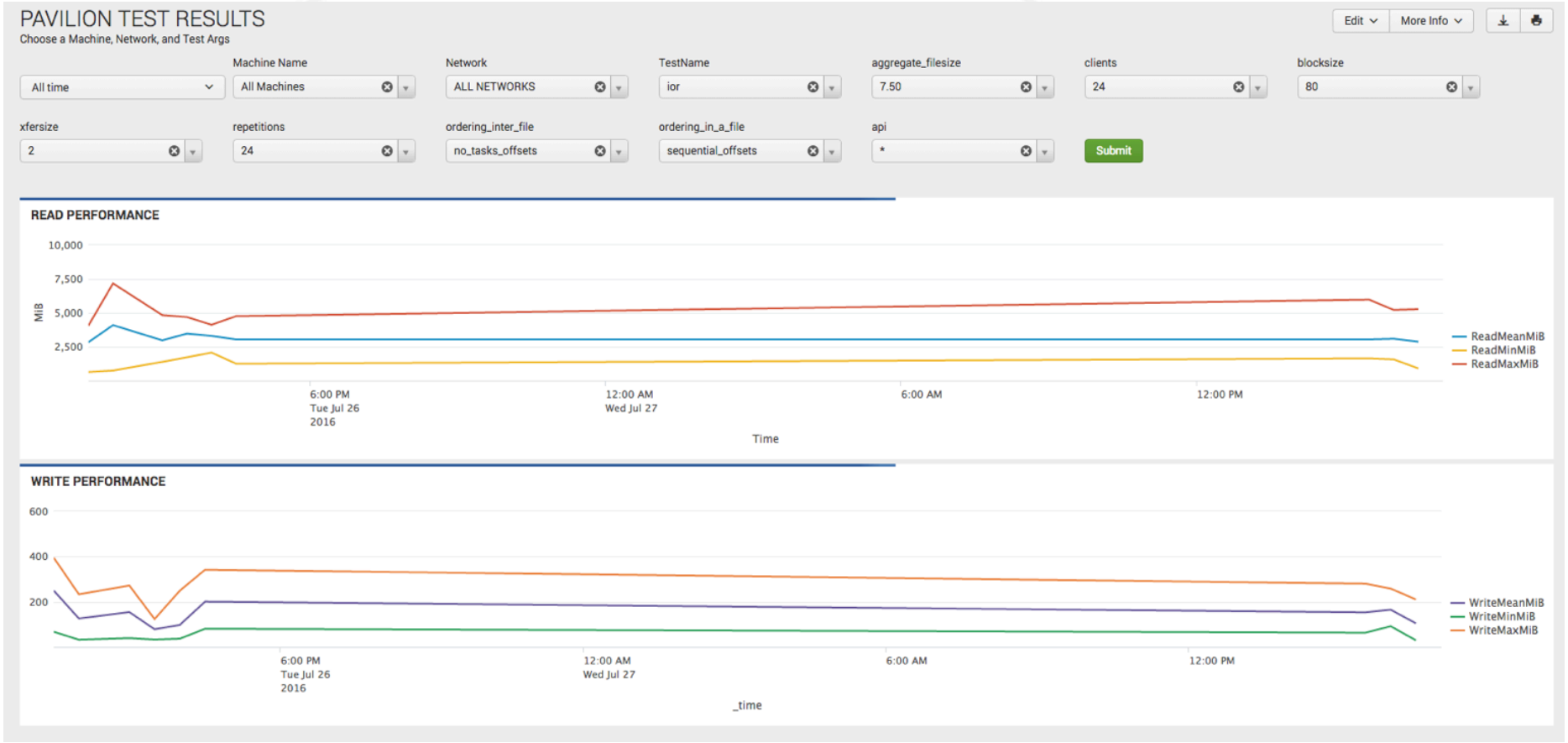
resultslist[$i]

where ops == “write”

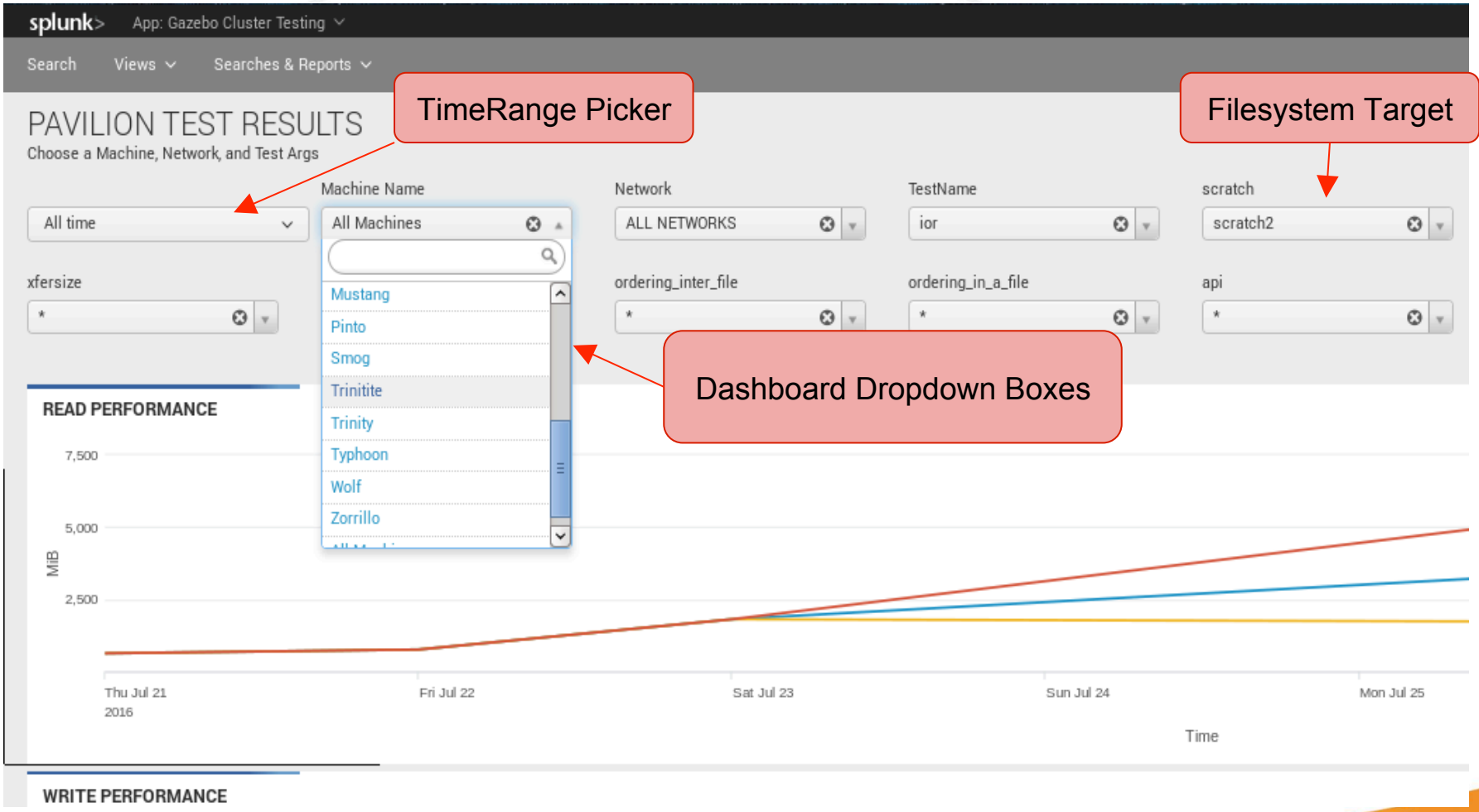
Splunk Data Format



Splunk Dashboard



Splunk Dashboard Features



Future Work

Future work will include

- testing researched methods of improving I/O using IOR as a benchmark tool
- Generating Baselines from Filesystem Team's Recommended Configurations
  - Determine Pass/Fail Criteria based on deviation from Baseline performance
- Improve Splunk Dashboard
  - Make Generic to Accept all Pavilion Tests Run at LANL
- Manipulate Lustre FS as user with Pavilion Test Args to capture performance differences with various stripe settings on targets
- Enable Darhsan in the Pavilion ior.yaml file
  - automate performance collection within a test
  - proof of concept in instrumenting Darshan as a Pavilion Plugin

Questions ??

Thank You