Mistral Supercomputer & I/O

Julian M. Kunkel, Carsten Beyer

kunkel@dkrz.de

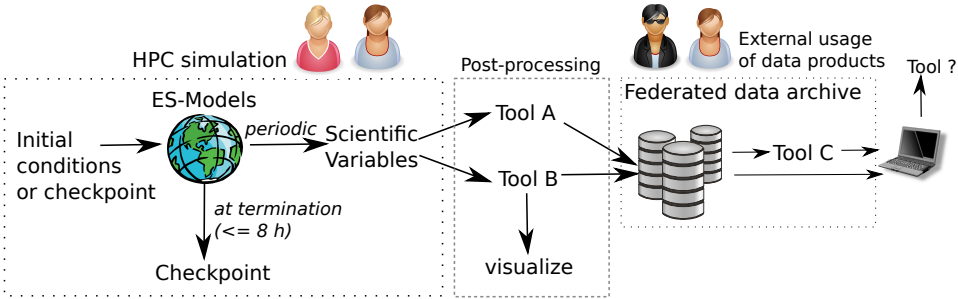
German Climate Computing Center (DKRZ)

30-07-2015

Outline

Scientific Workflow

A typical workflow



Technical background

- Application/domain-specific I/O servers for HPC-IO
- Different post-processing tools
- Involved libraries/formats: NetCDF4 (HDF5), NetCDF3, GRIB, ...

Tape Library with HPSS

- 6 Oracle/StorageTek SL8500 libaries (+ a smaller one)
  - More than 67,000 slots
- One SL8500 library at Garching for backups/disaster recovery
- Variety of tape cartriges/drives
- On Blizzard: 500 TB disk cache
- Update on Mistral: 3 PB disk cache

Mistral Supercomputer

- Phase 1 system, installed Q2/15, Phase 2 system in 2016
- Vendor: Atos (Bull)
- Nodes: 1500 with 2 Intel E5-2680 Haswell@2.5 GHz
  - 24 cores/node
  - 2 Nodes/blade, 9 blades/Chassis, 4 Chassis/Rack
- HPL-performance: 1.1 Petaflop/s, #56 of Top500
- Storage capacity: 20 Petabyte
- Network: FatTree with FDR-14 Infiniband
  - 3 Mellanox SX6536 core 648-port switches
  - 1:2:2 blocking factor
  - 1:1 within chassis (18 nodes)
  - 1:2 9 uplinks per chassis, to 3 linecards on each core switch
  - 1:2 between linecards and spinecards
- Power consumption (HPL): 700 kW

ClusterStor Servers

Phase 1: I/O Architecture

- Lustre 2.5 (+ Seagate patches: some back ports)
- 29 ClusterStor 9000 with 29 Extensions (JBODs)
  - 58 OSS with 116 OST
- ClusterStor 9000 SSUs
  - GridRaid: 41 HDDs, PD-RAID with 8+2(+2 spare blocks)/RAID6, 1 SSD for Log
  - 6 TByte disks
  - SSU: Active/Active failover server pair
  - ClusterStor Manager
  - 1 FDR uplink/server
- Peak performance
  - Infiniband FDR-14: 6 GiB/s $\Rightarrow$ 348 GiB/s
  - CPU/6 GBit SAS: 5.4 GiB/s $\Rightarrow$ 313 GiB/s
- Multiple metadata servers
  - Root MDS + 4 DNE MDS
  - Active/Active failover (DNEs, Root MDS with Mgmt)
  - DNE phase 1: Assign responsible MDS per directory

Performance Results from Acceptance Tests

- Throughput measured with IOR
  - Buffer size 2000000 (unaligned)
  - 84 OSTs (Peak: 227 GiB/s)
  - 168 client nodes, 6 procs per node

Type	Read	Write	Write rel. to peak[2]
POSIX, independent[1]	160 GB/s	157 GB/s	70%
MPI-IO, shared[2]	52 GB/s	41 GB/s	18%
PNetCDF, shared	81 GB/s	38 GB/s	17%
HDF5, shared	23 GB/s	24 GB/s	10%
POSIX, single stream	1.1 GB/s	1.05 GB/s	0.5%

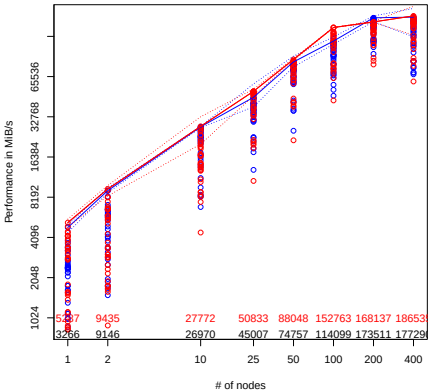
- A few slow servers significantly reduce IOR performance
  - Also: Congestion on IB routes degrade performance
- Metadata measured with a load using Parabench: 80 kOPs/s

[1] 1 stripe per file

[2] 84 stripes per file on 21 SSUs

First Performance Results

- Goal: Identify good settings for I/O
- IOR, independent files & 10 MiB blocks
  - Measured on the production system
  - Slowest client stalls others
  - Proc per node: 1,2,4,6,8,12,16
  - Stripes: 1,2,4,16,116



Best settings for read (excerpt)

Nodes	PPN	Stripe	W1	W2	W3	R1	R2	R3	Avg. Write	Avg. Read	WNode	RNode	RPPN	WPPN
1	6	1	3636	3685	1034	4448	5106	5016	2785	4857	2785	4857	809	464
2	6	1	6988	4055	6807	8864	9077	9585	5950	9175	2975	4587	764	495
10	16	2	16135	24697	17372	27717	27804	27181	19401	27567	1940	2756	172	121

Defaults: fixed stripe & PPN for running on all numbers of nodes

Stripes	PPN	RNode	WNode	R arithmetic mean	W arithmetic mean	RHarmonic	WHarmonic
1	1	788	780	34	32	52	53
1	2	1214	1155	53	47	75	71
1	4	1352	1518	59	62	81	79
1	6	2179	1835	95	75	91	88
1	8	1943	2235	84	92	86	93
1	12	1974	1931	86	79	92	84
1	16	1890	1953	82	80	84	72
2	1	734	763	32	31	51	51
2	2	1165	1182	50	48	72	72
2	4	1814	1745	79	71	87	85
2	6	1935	1693	84	69	88	83
2	8	1726	2039	75	84	88	89
2	12	1780	2224	77	91	90	92
2	16	1806	1752	79	72	79	75
4	1	726	761	31	31	49	51
4	2	1237	1185	54	48	70	66
4	4	1737	1744	75	71	85	84
4	6	1719	1888	75	77	85	86
4	8	1751	1931	76	79	87	90
4	12	1841	1972	80	81	87	89
4	16	1745	2064	76	85	72	74
16	1	743	726	32	29	48	49
16	2	1109	1216	48	50	66	71
16	4	1412	1554	61	64	75	81
16	6	1489	1812	65	74	72	85
16	8	1564	1841	68	75	79	90
16	12	1597	1939	69	79	71	78
16	16	1626	1900	71	78	64	68
116	1	588	432	25	17	34	31
116	2	871	773	38	31	44	52
116	4	1270	1258	55	51	53	69
116	6	1352	978	59	40	52	51
116	8	1397	901	61	37	56	47
116	12	1470	1020	64	42	55	46
116	16	1503	1147	65	47	55	42

Monitoring Tools

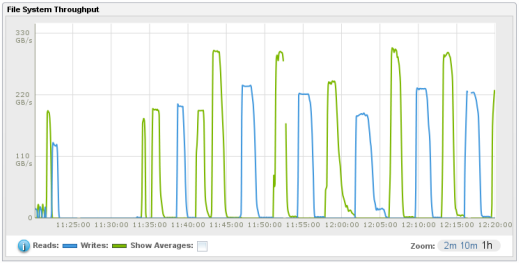
On Mistral

- For compute
  - Nagios (status & performance)
  - Planned: XDMoD (for utilization)
  - Slurm statistics (accounting)
- Seagate's Data Collection System (DCS)
  - Metadata and data rates
  - CPU and MEM utilization
  - Node Health
- ltop
- `cscli lustre_perf`
- ClusterStor Manager
- Planned: SIOX

On Blizzard

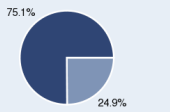
- Nagios
- llview (for Load-Leveler)
- Ganglia
- ibview

Monitoring I/O Performance with ClusterStor



Top System Statistics

	Metric		
File System	Peak Read	274.34	GB/s
	Current Read	45.6	KB/s
	Peak Write	308.37	GB/s
	Current Write	222.54	GB/s
Metadata	Current Operations	0	Op/s
Storage	Number of OSTs in use	116	
	Number of Disks in use	4,882	
	Capacity in use	4.78	PB
	Capacity available	14.41	PB
Power	Current cluster usage	70.17	KW



Obstacles

Lack of knowledge

- Usage of file formats and middleware libraries is limited
  - Analysis of file extensions does not suffice
  - Library usage could theoretically be monitored, but ...
- The workflows of users is sometimes diffuse
- The cause of innefficient operations is unknown

Shared nature of storage

- With 1/60th of nodes one can drain 1/7th of I/O performance
  - ⇒ 10% of nodes drain all performance
  - Since applications are not doing I/O all the time this seems fine
- But: interaction of I/O may degrade performance
  - I/O intense benchmark increased application runtime by 100%
- Metadata workloads are worse, problematic with broken scripts

Obstacles

Difficulties in the analysis

- Performance is sensitive to I/O patterns, concurrent activitity
- Infiniband oversubscription
- Application-specific I/O servers increase complexity
- Capturing a run's actual I/O costs
- Lustre's (performance) behavior

Others

- Outdated (and inefficient) file formats are still dominant
- Performance of RobinHood may be too slow (2000 ops/s)
- Capability increase from Blizzard to Mistral[3]
  - Compute performance by 20x
  - Storage performance by 25-30x
  - Storage capacity by 7x

Relevant R&D

- Monitoring and analysis of I/O on system AND application level
- Monitoring user workflows
- Optimized data layouts for HDF/NetCDF
- Accounting of I/O in Slurm but also with additional granularity
- Reduce interference of concurrent I/O (interactive vs. large scale runs)
- Evaluation of alternative storage backends (DDN's IME, object storage)
- Compression of data (lossless 1:2.5, lossy 1:10)

! At best without changes on user apps