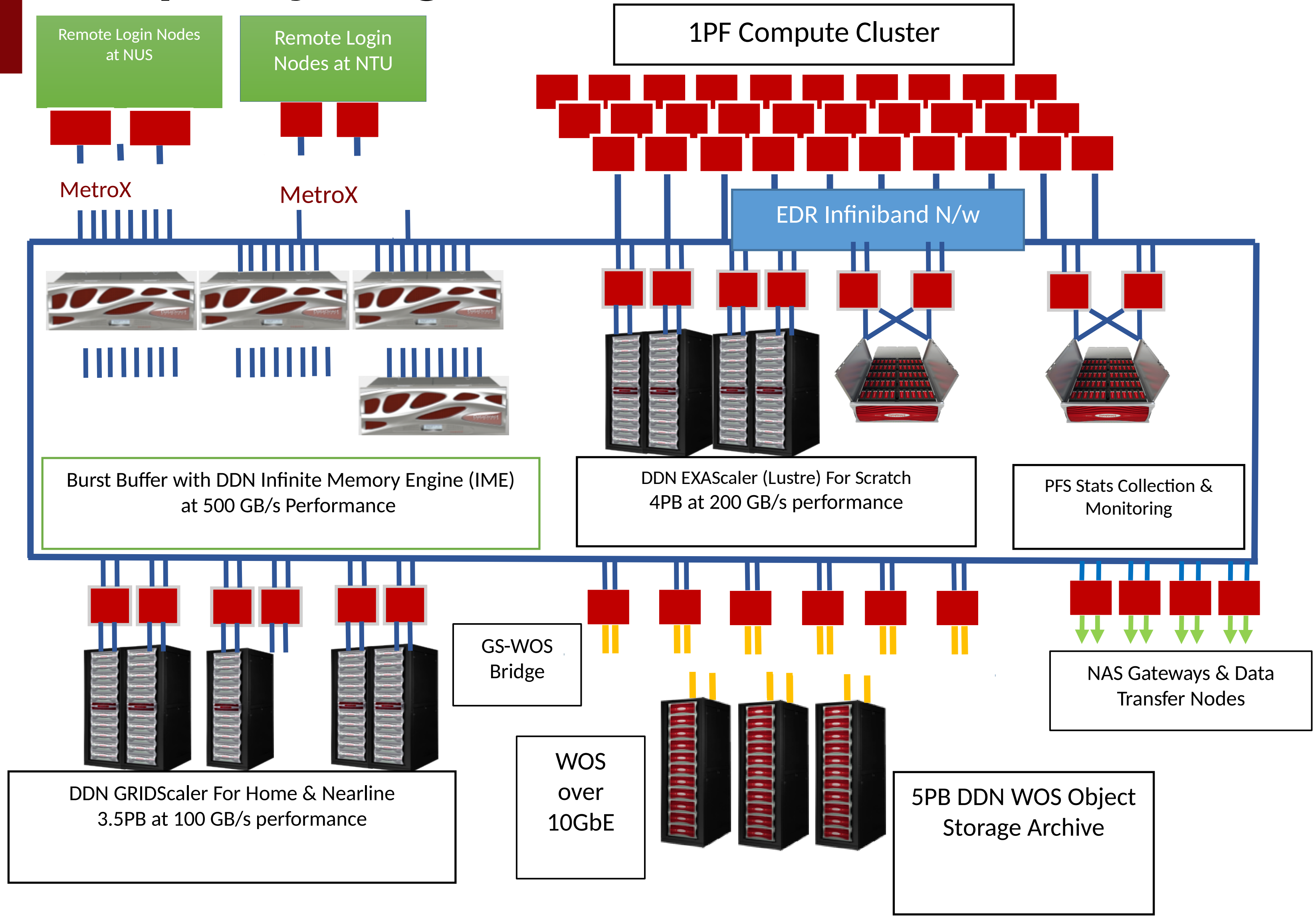
Bridging the complexity gap: Tracing and Replaying I/O

UIOP 2017, Hamburg , Mar. 22^{nd}

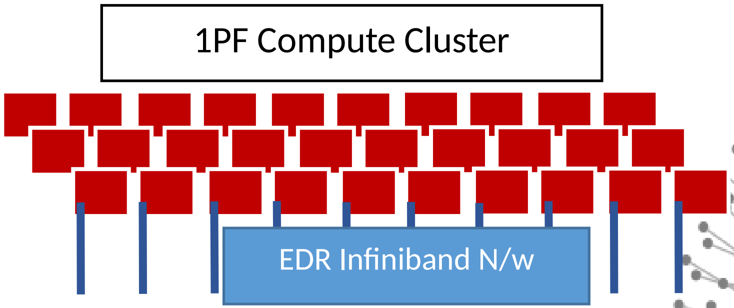
Jean-Thomas Acquaviva, DDN Storage

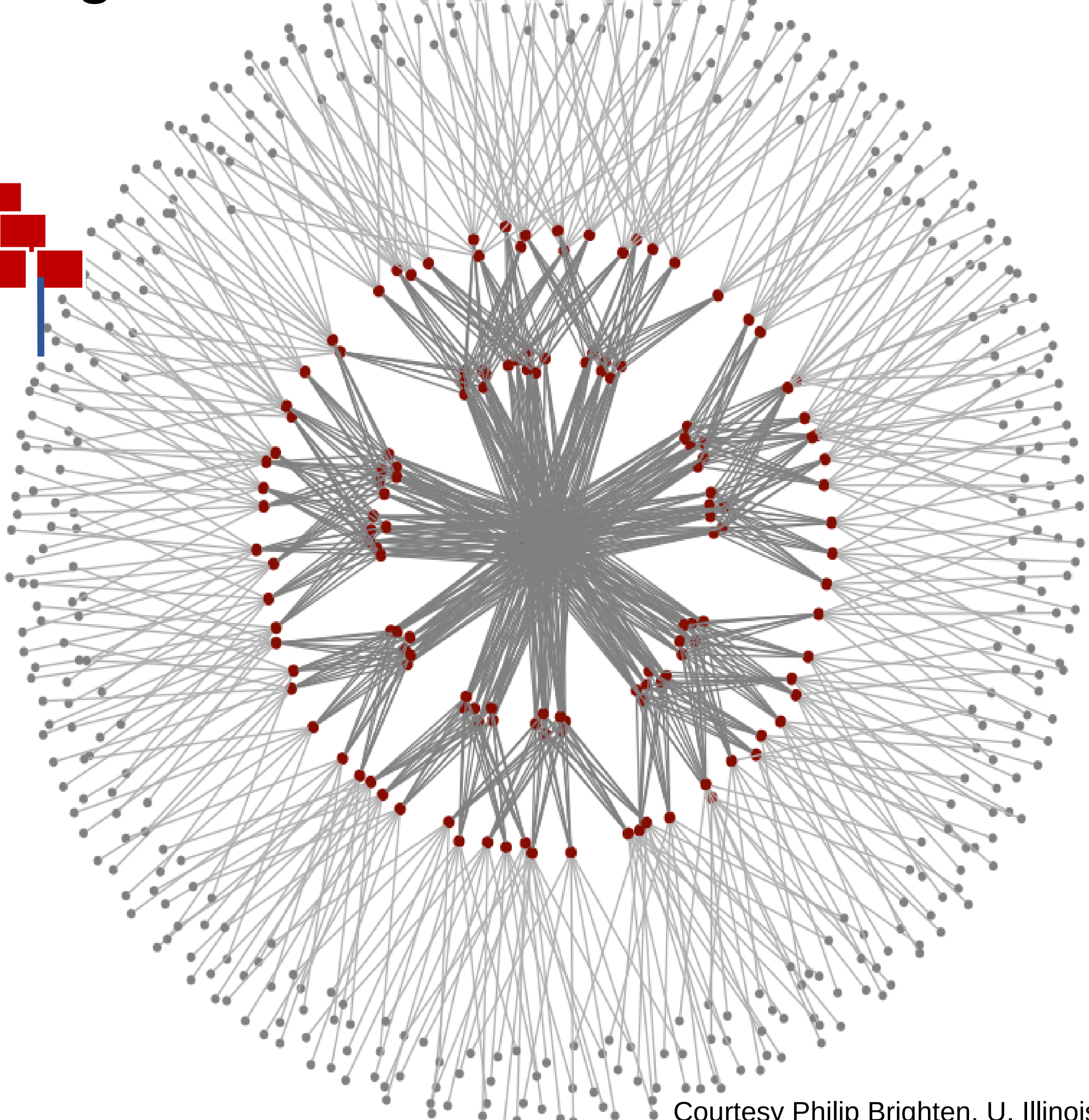
Complexiy: E.g NSCC / A*STAR



Courtesy Philip Brighten, U. Illinois

Complexity: E.g 800 nodes Fat Tree

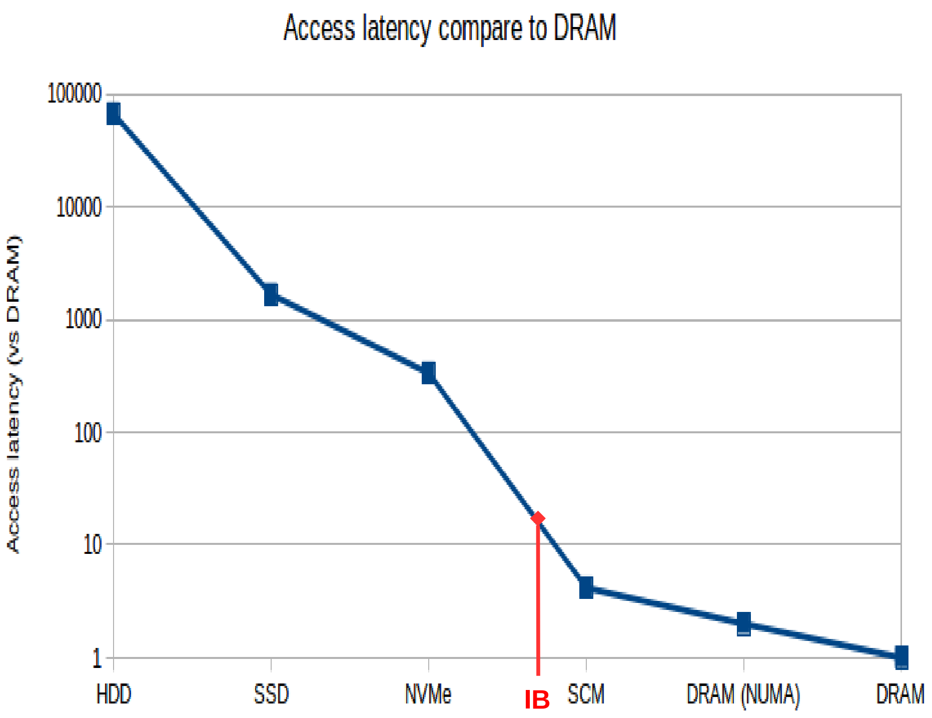


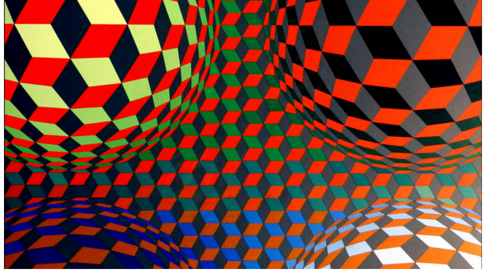


Courtesy Philip Brighten, U. Illinois

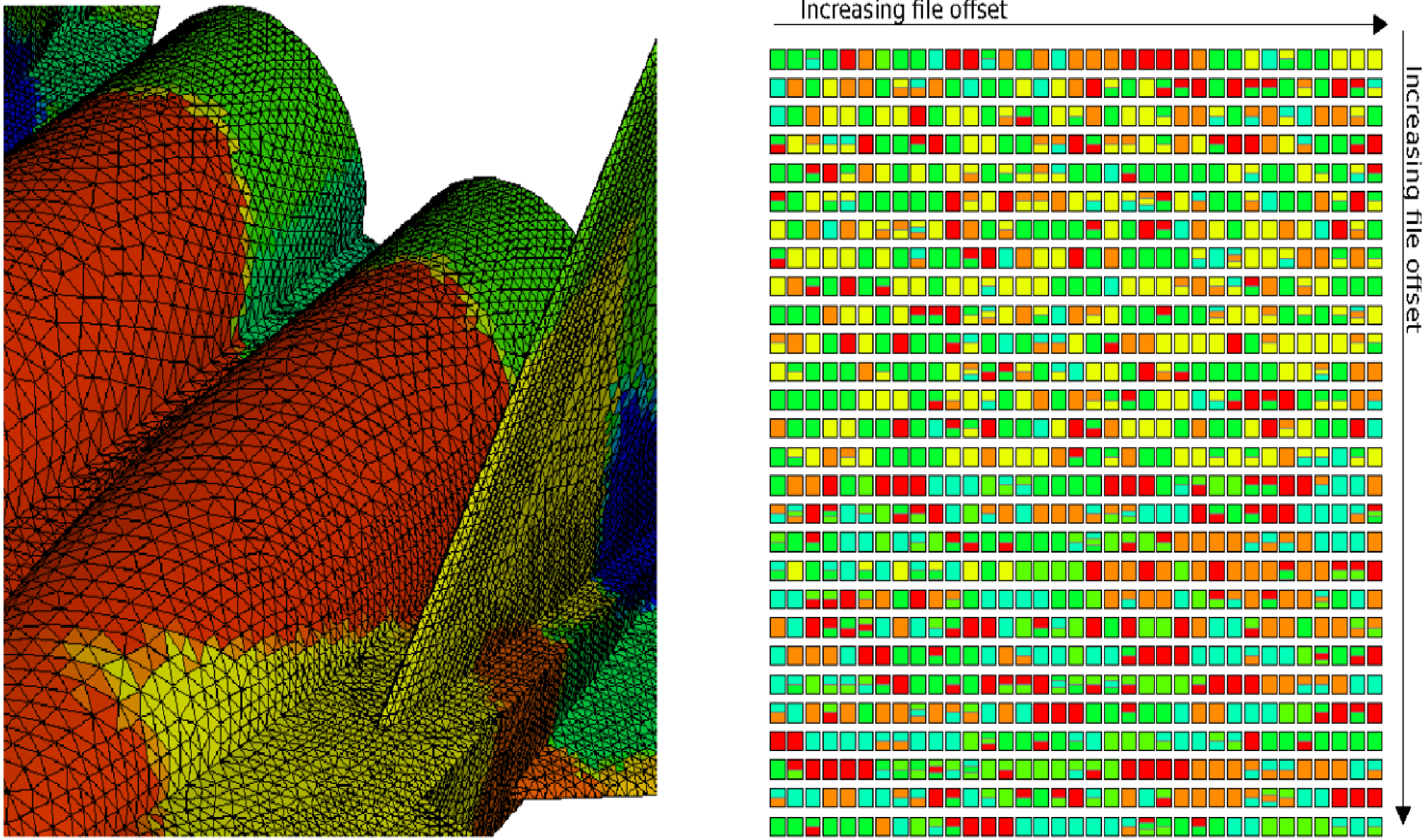
4 Complexity Trend: Deeper Storage Hierarchy

Hierarchy mechanically increases predominance of patterns in the performance equation





Spatial Locality Patterns

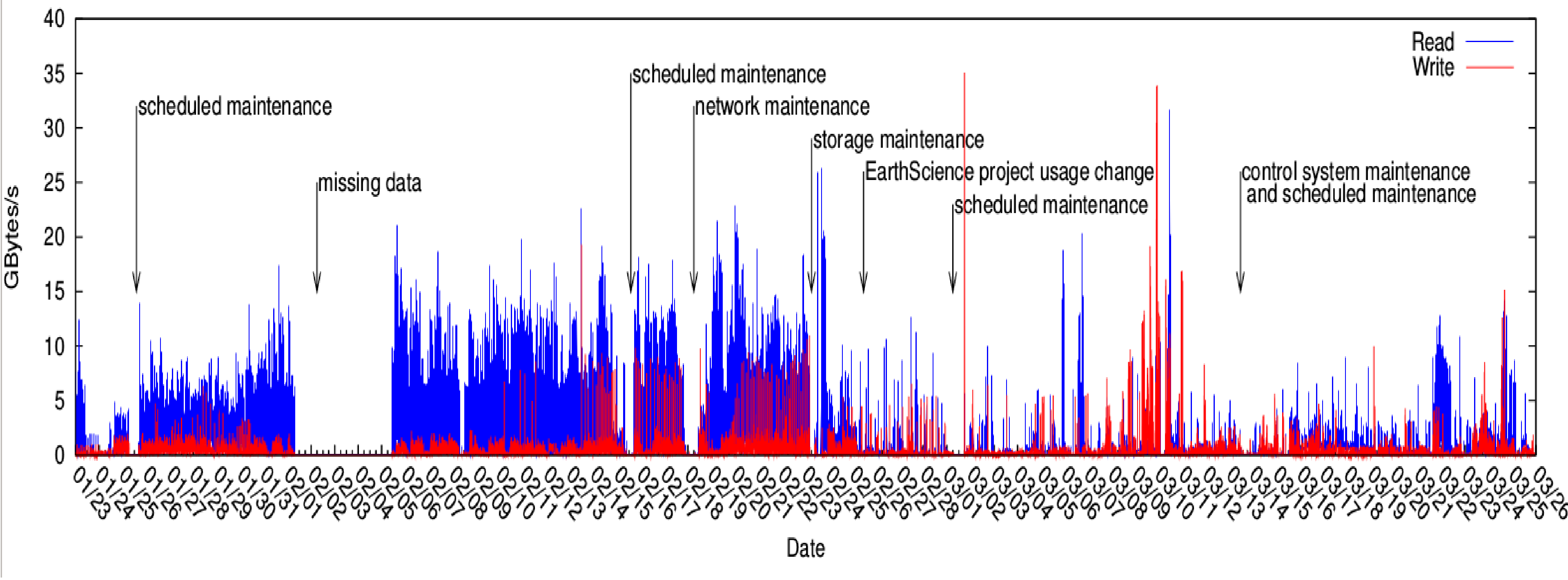


(a) Decomposed mesh (b) File mapping

Source: Storage Models: Past, Present, and Future. Dres Kimpe et Robert Ross, Argonne National Laboratory

DDN STORAGE

Temporal pattern



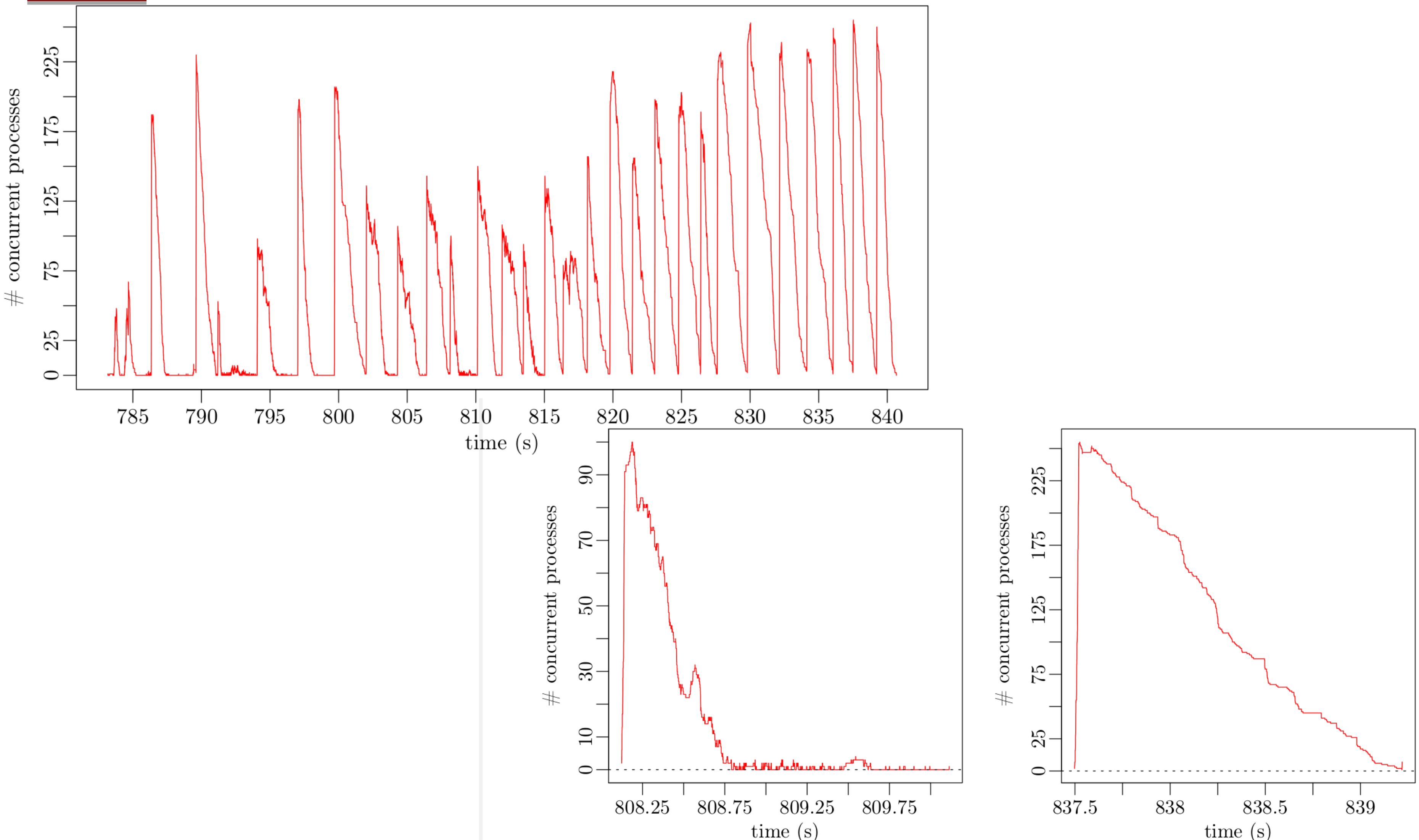
99% of the time the IO sub-system is stressed bellow 30% of its bandwidth

70% of the time the system is stressed under 5% of its peak bandwidth

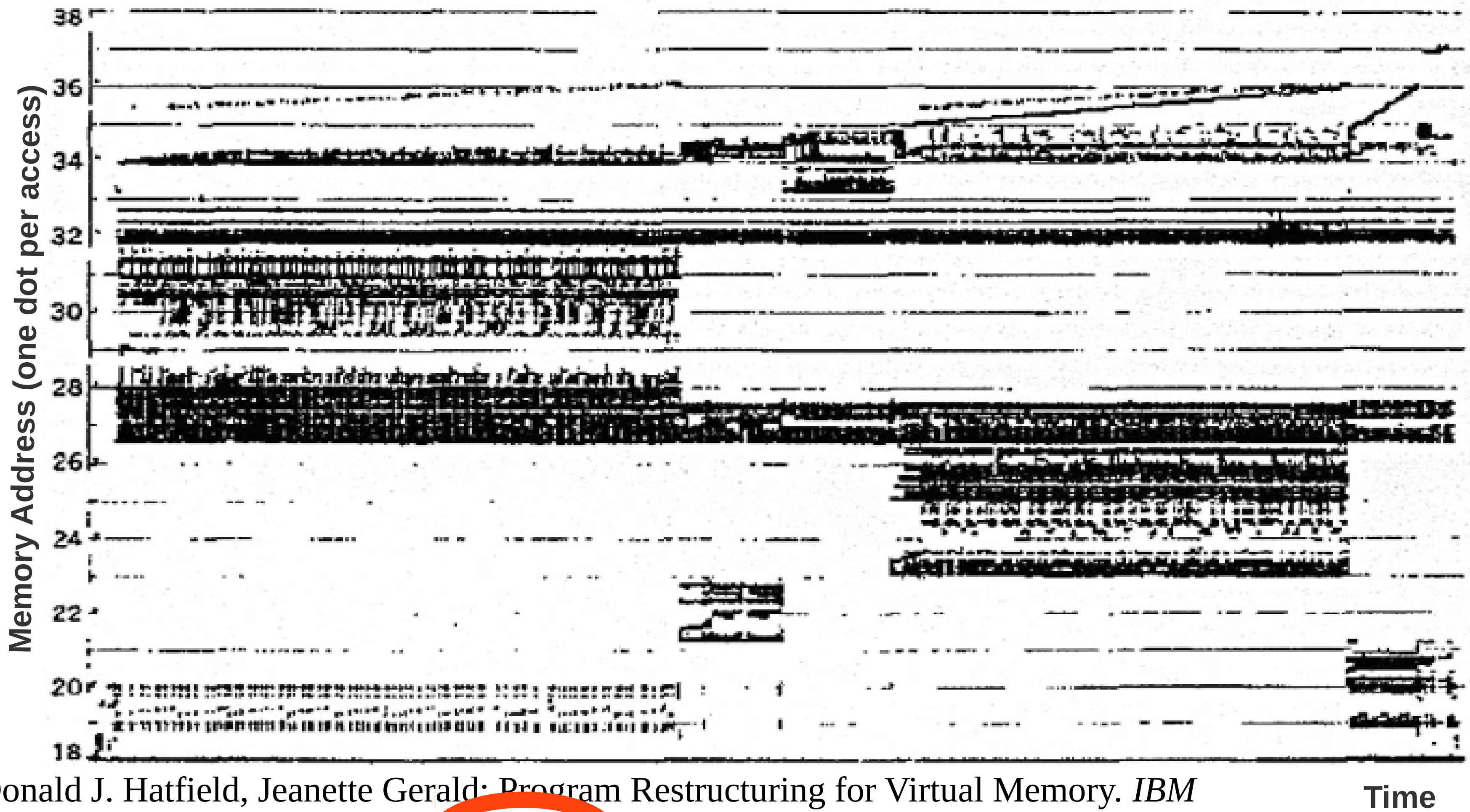
Argone lab.

P. Carns, K. Harms et al., ***Understanding and Improving Computational Science Storage Access through Continuous Characterization***

File contention is temporal

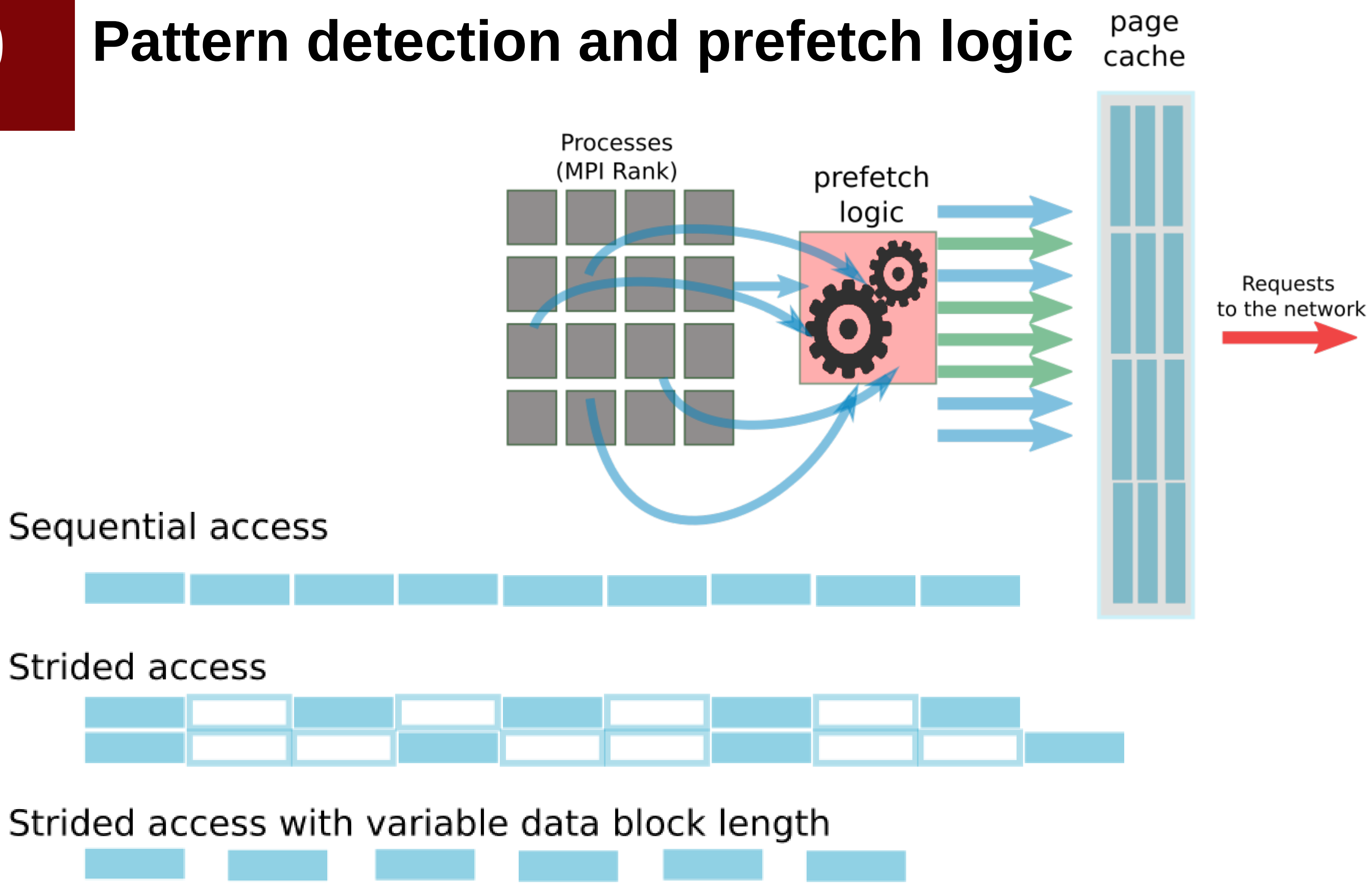


Spatial and Temporal Patterns



Donald J. Hatfield, Jeanette Gerald: Program Restructuring for Virtual Memory. *IBM Systems Journal*, 10 (3): 168-192 (1971)

Pattern detection and prefetch logic



10 Storage devices are sensitive to pattern → increased life expectancy

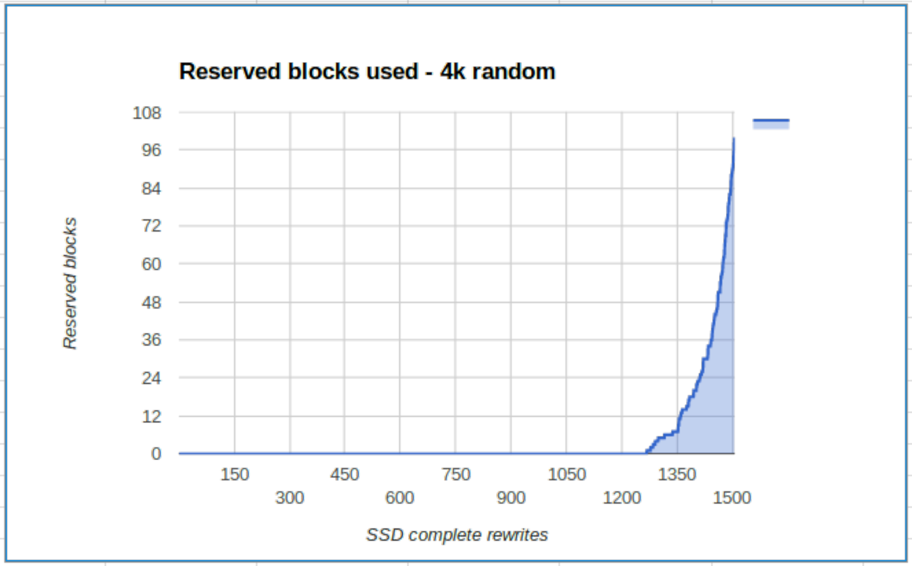
1) Vendors estimate life time on 4K random write pattern
2) IME limits write amplification
3) Combo: better performance + longer life

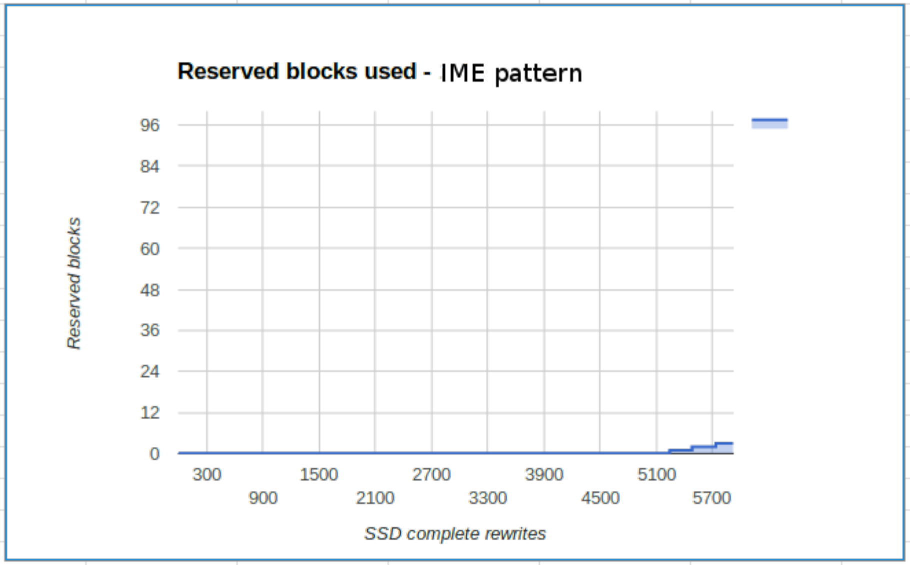
Testbed:
Burning SSD with different patterns + monitoring SMART counters

ID#	ATTRIBUTE_NAME	FLAG	VALUE	WORST	THRESH	TYPE	UPDATED	WHEN_FAILED	RAW_VALUE
5	Reallocated_Sector_Ct	0x0033	093	093	010	Pre-fail	Always	-	62
9	Power_On_Hours	0x0032	099	099	000	Old_age	Always	-	1986
12	Power_Cycle_Count	0x0032	099	099	000	Old_age	Always	-	7
177	**Wear_Leveling_Count**	**0x0013**	**001**	**001**	**000**	**Pre-fail**	**Always**	**-**	**6242**
179	**Used_Rsvd_Blk_Cnt_Tot**	**0x0013**	**093**	**093**	**010**	**Pre-fail**	**Always**	**-**	**62**
181	Program_Fail_Cnt_Total	0x0032	100	100	010	Old_age	Always	-	0
182	Erase_Fail_Count_Total	0x0032	100	100	010	Old_age	Always	-	0
183	**Runtime_Bad_Block**	**0x0013**	**093**	**093**	**010**	**Pre-fail**	**Always**	**-**	**62**
187	Reported_Uncorrect	0x0032	099	099	000	Old_age	Always	-	21
190	Airflow_Temperature_Cel	0x0032	059	048	000	Old_age	Always	-	41
195	Hardware_ECC_Recovered	0x001a	199	199	000	Old_age	Always	-	21
199	UDMA_CRC_Error_Count	0x003e	100	100	000	Old_age	Always	-	0
235	Unknown_Attribute	0x0012	099	099	000	Old_age	Always	-	2
241	Total_LBAs_Written	0x0032	099	099	000	Old_age	Always	-	346288758224

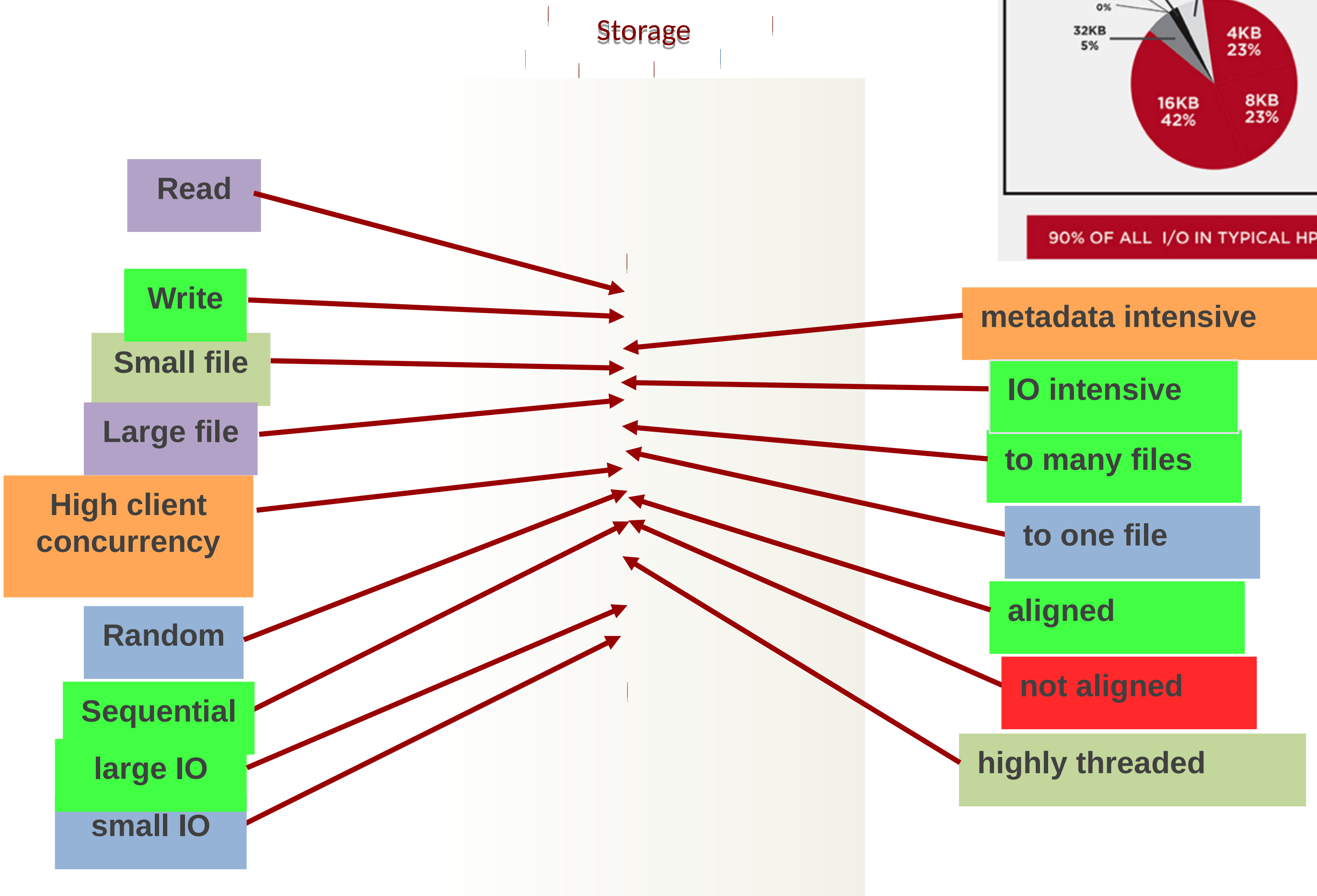
11 Limited Write amplification → increased life expectancy

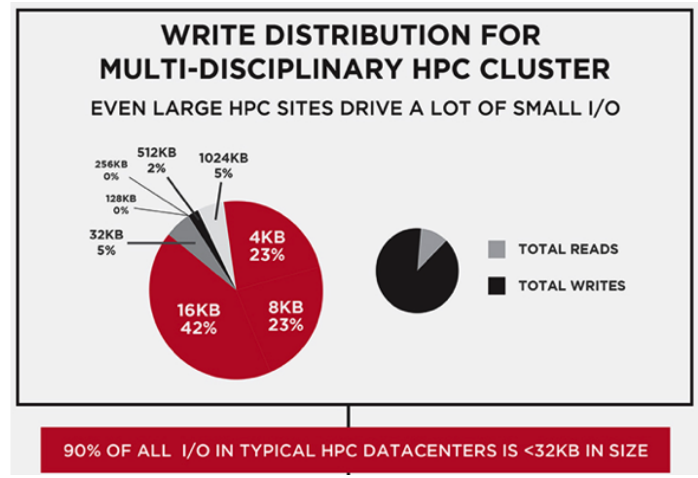
1) Assess devices Life expectancy
2) Understand deprecation rate
→ preventive maintenance





Parallel File System are sensitive to patterns





Community answers

- Real applications
- Synthetic benchmarks
- Analytical Model
- Mini-app

Community answers

- **Real applications**
  - **Best estimation of production run**
    - *Is the application going to run alone?*
  - **Costly: require expertise and topnotch engineering**
- **Synthetic benchmarks**
  - IOR
  - IOZONE
  - MDTEST
    - *Do not capture temporal locality*

Analytical Model: NEural Simulation Tool (NEST)

Courtesy

NEST, computational neuro-science

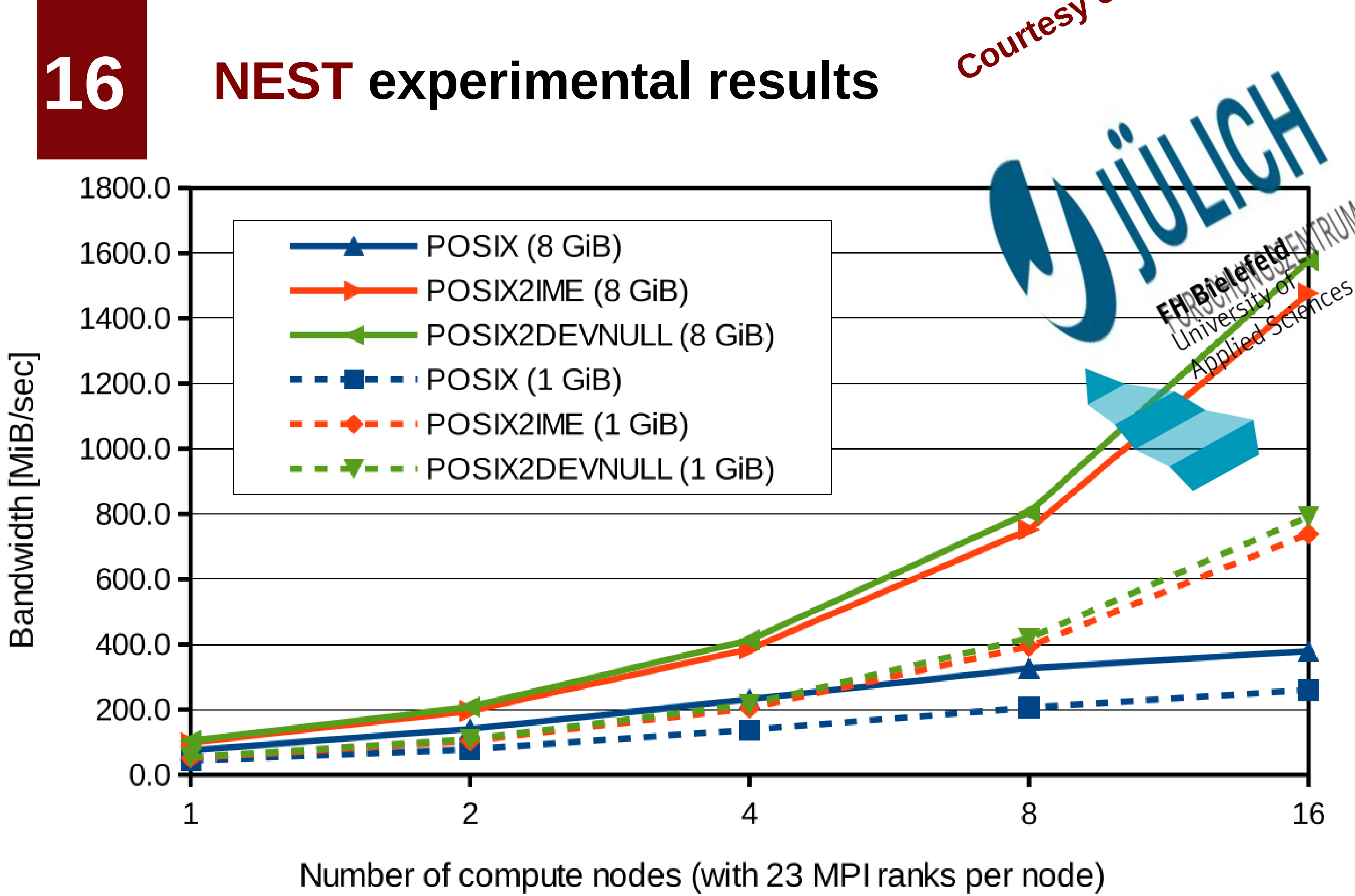
- Dynamics of interactions between nerve cells
- First step of wiring a neural network
- Next step simulate the network of spiking point neurons

- Developed for both local small experiments or deployment on leading super-computer for extreme-scale simulations
  → MPI + OpenMP
- I/O pattern burst of write at the end of every simulation step

Early Evaluation of the "Infinite Memory Engine" Burst Buffer Solution
WOPSSS ’16, 2016, Frankfurt

Wolfram Schenck
Faculty of Engineering and Mathematics Bielefeld University of Applied Sciences Bielefeld, Germany
Salem El Sayed, Maciej Foszczynski, Wilhelm Homberg, Dirk Pleiter
Jülich, Germany Jülich Supercomputing Centre Forschungszentrum Jülich

NEST experimental results



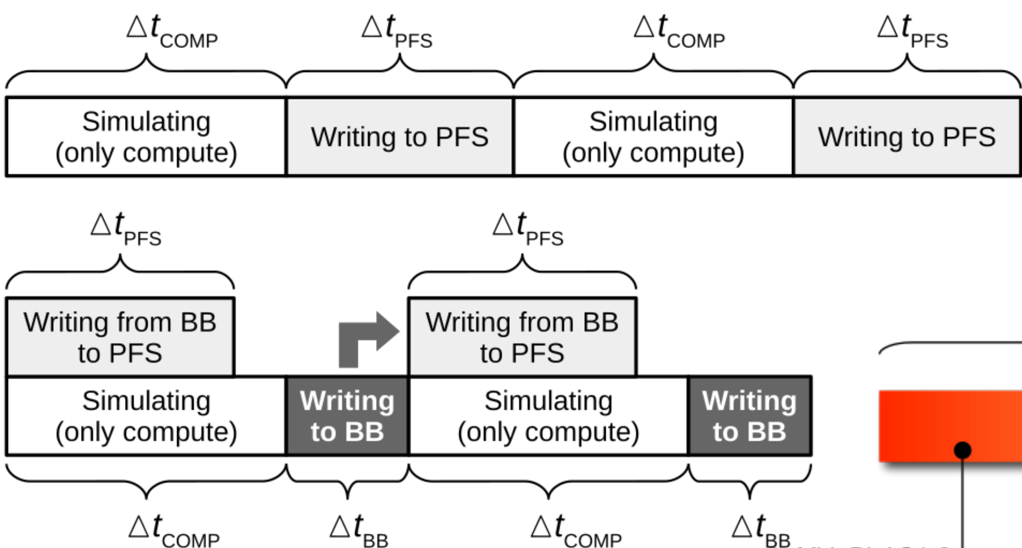
Platform – JUDGE: IBM iDataPlex NodeDual Xeon Westmere – 48 GB RAM – Network IB-QDR

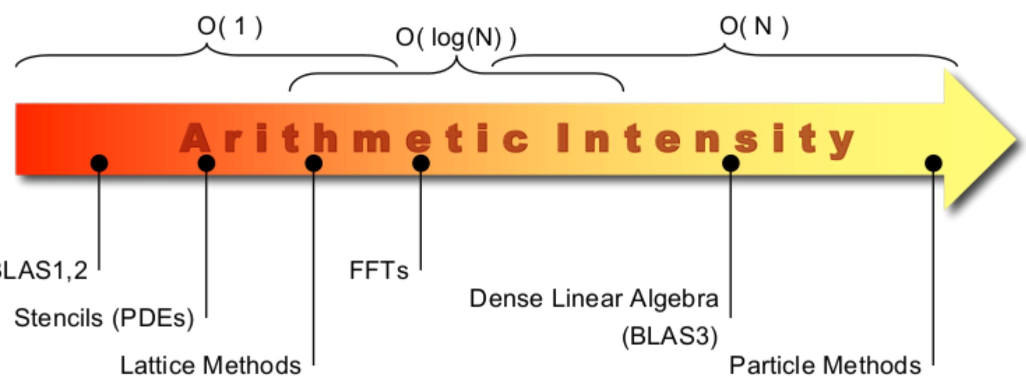
NEST modeling and analysis

Courtesy

→ *hum… POSIX2IME and POSIX2DEVNULL show nearly identical behavior*

If computation fully overlap I/O bandwidth is irrelevant





1 bit of I/O per FLOP is a rule of thumb [see Jim Gray]

Analytical model

- **Allow to understand**
  - **Bottleneck isolation**
    - Optimize both application and architecture
- **Difficult to scale**
  - **Difficult to extend to an application portfolio**
  - **Difficult to model complex workload**
    - Resource sharing

Mini-apps: Brain Simulation and Neuromapp

Mini-apps have been fashionable since quite a while in HPC it's reaching now I/O

See github.com/PETTT

Brain simulator are large SW

3 decades of development

500 KLOC + DSL + src2src compiler

NeuroMapp

- **Mini-app framework**
- **Each mini-app (1KLOC) represents a single critical neuron scientific algorithm$**
- ***Stay tuned, results to be published***

Mini-apps can be assembled together

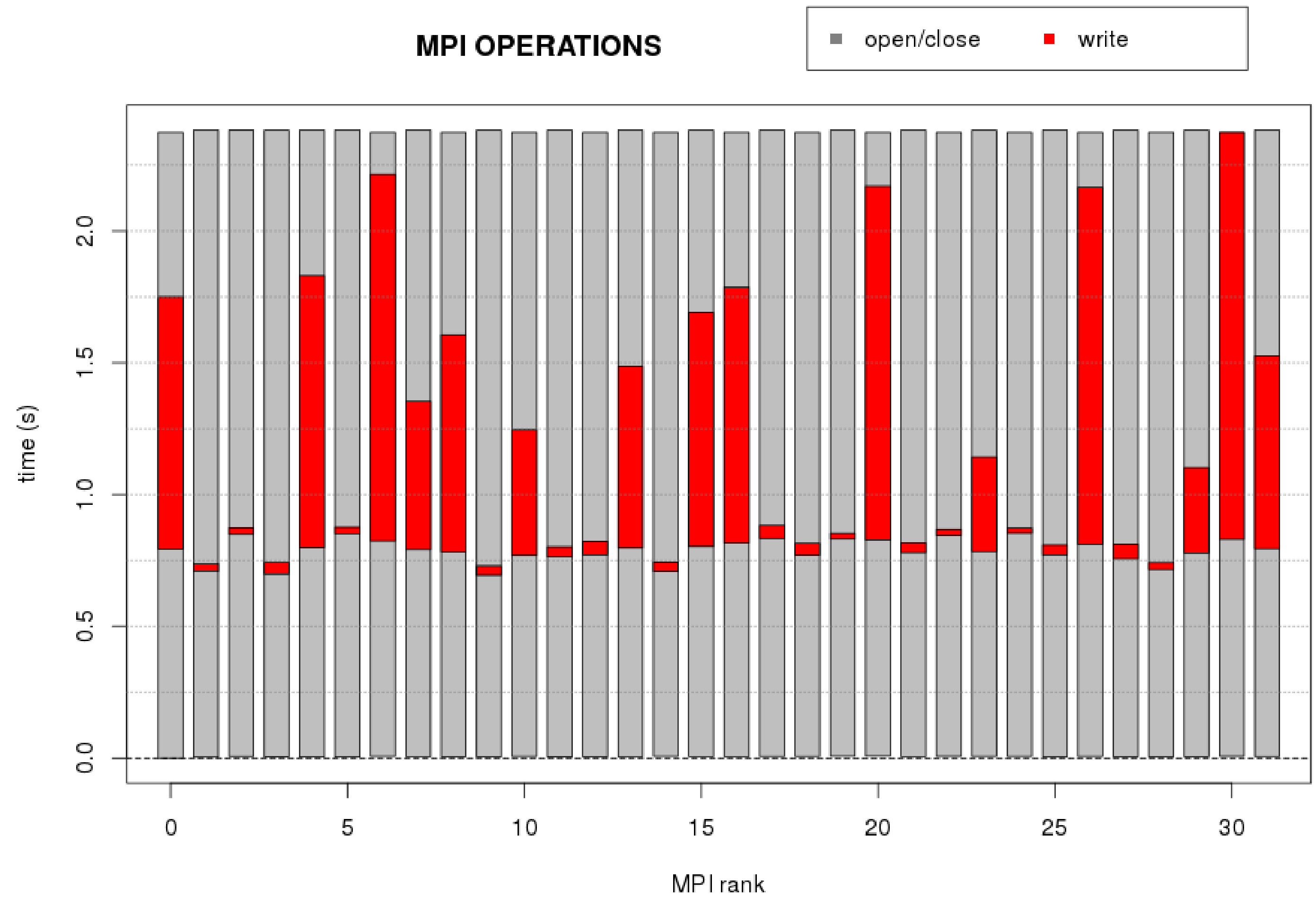
- **Form the skeleton of the initial scientific application**
- **Experiment work-flow optimization opportunities**

github.com/BlueBrain/neuromapp

Mini-App: summary

- Easy to deploy
- Provide ‘reasonable’ estimations
- Provide ‘reasonable’ insight
- Costly to develop
  - Require code maintenance
  - Evolve jointly with the core application

DIO-pro


MPI OPERATIONS
open/close
write
time (s)
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
25
30
MPI rank

I/O Profiling with DIO-pro

- Capture I/O traffic
  - Cope with high I/O loads (overhead <1%)
  - Support Posix and MPI-IO
- Characterize I/O patterns
  - Build a distributed accurate clock
- Evaluate I/O efficiency
- Extrapolate performances
  - Architectural prospection
  - Estimate IME perf. impact

- **DIO-pro is not yet a product still a prototype**

General overview of DIOPRO

Default scenario

Application

write()

I/O library

file

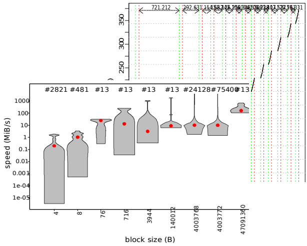
DIOPRO

dlsym trick:

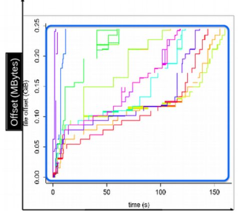
```
#define MAP_FUNCTION(func) \
    if (!(__real_ ## func)) \
          __real_ ## func = dlsym(RTLD_NEXT, #func); \
}
```

Read buffer → **DioPro** → Produces compressed trace file

Trace → Read trace file → **DioPro** → Produces analysis



Read trace file → **DioPro** → Produces performance numbers



DIO-pro sanity check

```
$ mpirun -x LD_PRELOAD=/usr/lib/dio-pro.so -n 8 ./mpi-write-multi -f output/file -l 16
 0: io_time = 0.078403
 1: io_time = 0.093839
 2: io_time = 0.086452
 3: io_time = 0.167057
 4: io_time = 0.065364
 5: io_time = 0.111795
 6: io_time = 0.167707
 7: io_time = 0.150377
longest_io_time      = 0.167707 seconds
total_number_of_bytes = 536870912
transfer rate        = 3052.944138 MB/s
```

Timespans and I/O speed matches

(given a small error)

The source code says this should be MiB/s

```
$ zcat /tmp/dio-pro/iotrace-jobid-*.bin.gz | dio-pro-xml | dio-pro-stat -p | grep 'Process\|write'
Process 1:   ID: 12382 - MPI rank: 0
write   67.108864 MB        1    0.078378 s  816.558 MiB/s  816.558 MiB/s  816.558 MiB/s      0   B/s
Process 2:   ID: 12383 - MPI rank: 1
[...]
Process 8:   ID: 12390 - MPI rank: 7
write   67.108864 MB        1    0.150335 s  425.717 MiB/s  425.717 MiB/s  425.717 MiB/s      0   B/s
```

(bytes*8)/max(time)/1024^2 = [MiB/s]

(67108864*8)/max(0.078378,0.093811,0.086434,0.167011,0.065349,0.111760,0.167680,0.150335)/1024^2 = **3053.435 MiB/s**

What To Do With I/O Traces? Observe!

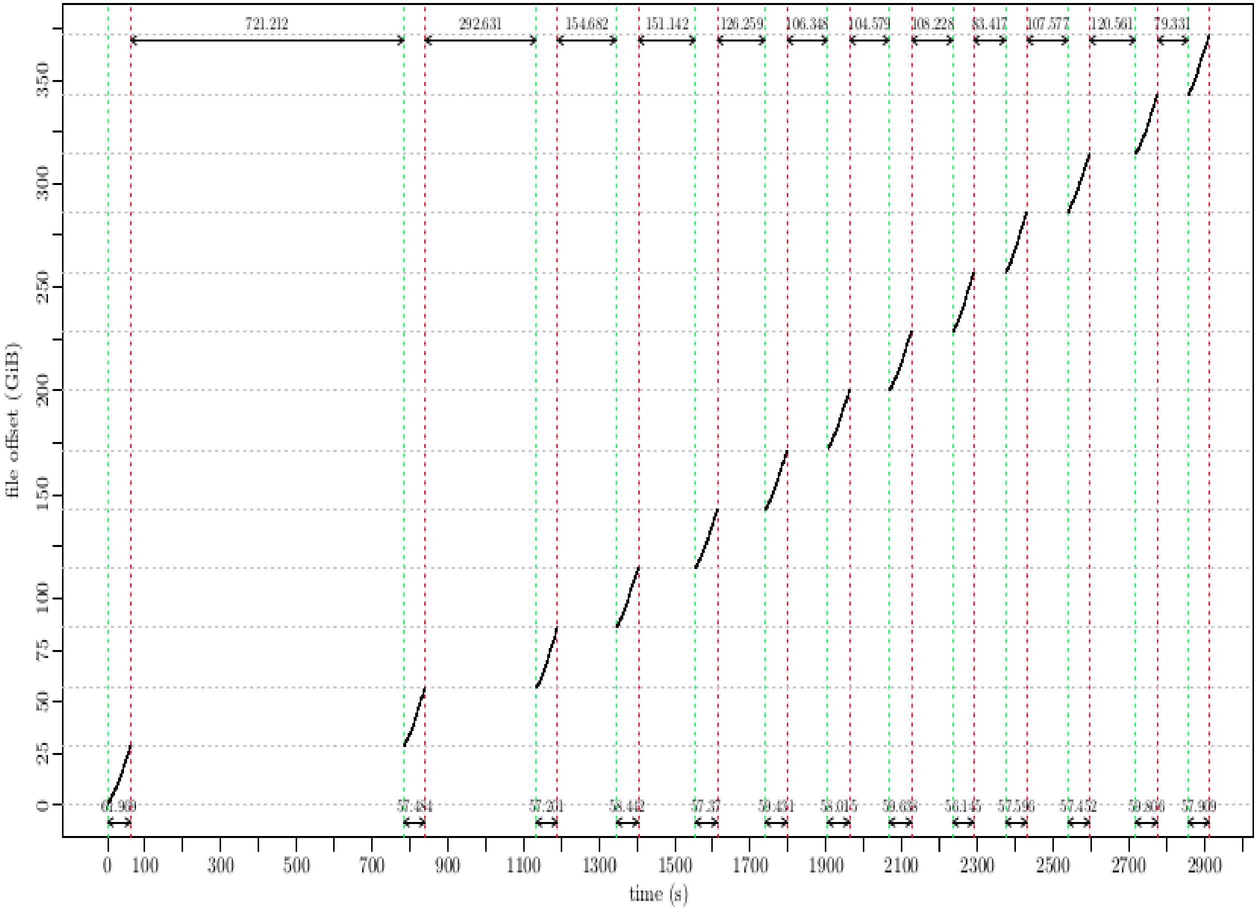


Figure 4: Temporal overview of the `.bigSharedFile` file's write offset. `IO MPI Ranks` ranks write to the `.bigSharedFile` file in 13 *steps*, as shown in the figure. Each step contains 29 *arrays*. If one could zoom into the arrays, `IO MPI Ranks` blocks of around 4MB would be visible (See Figure 7). The time-wise length (bottom arrows) of each I/O step is indicated between the green and red vertical dashed lines. The time between each step is shown (top arrows) between the red and green dashed lines.

DDN STORAGE

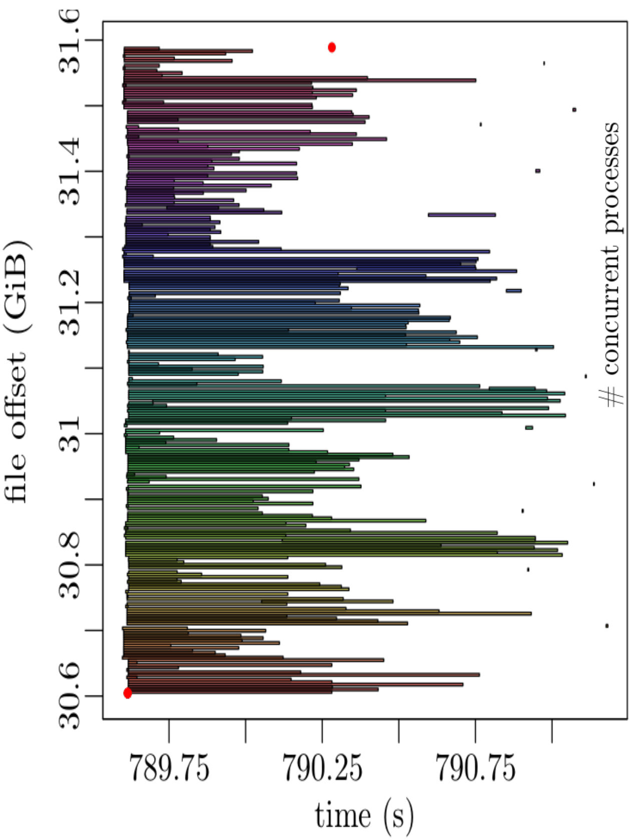
ddn.com

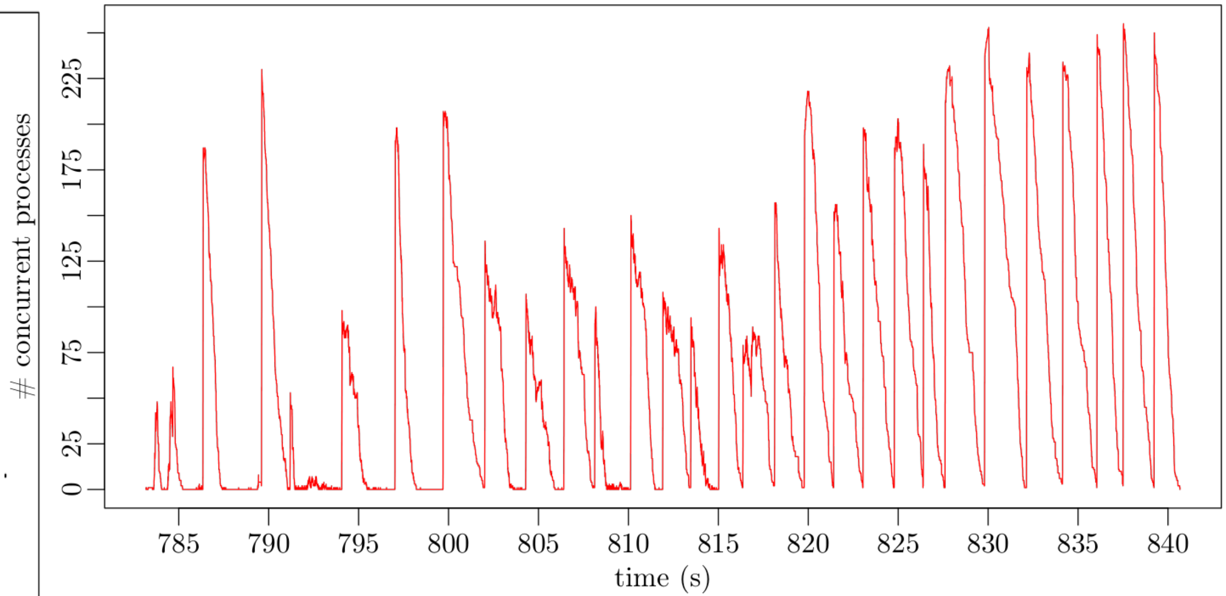
What To Do With I/O Traces? Solve!

Tag	Ranks	Read					Write					Meta		
		%	sec	bytes	MiB/s	#	%	sec	bytes	MiB/s	#	%	sec	#
M1	0:263	0	0	0	0	0	94.11	50565.613	399,101,407,640	7.527	102895	0.84	11.344	100056
M2	0:2499	0	0	0	0	0	1.17	5954.931	9,526,773,296	1.526	6670	4.29	545.382	12500
M3	0:2499	0	0	0	0	0	1.14	5778.100	25,046,129,003	4.134	8711	2.48	315.829	15005
M4	0:2499	0	0	0	0	0	1.01	5143.017	22,747,889,798	4.218	8711	1.43	182.066	15005
M5	0:2499	0	0	0	0	0	0.84	4262.277	28,538,881,332	6.386	7341	3.76	478.320	72500
M6	0:2499	0	0	0	0	0	0.36	1811.795	4,967,209,534	2.615	6672	0.91	115.303	7522
M7	0:2499	0	0	0	0	0	0.17	857.708	1,111,308	0.001	8778	1.72	219.008	5033
M8	0:2499	0	0	0	0	0	0.14	736.059	1,111,308	0.001	8778	1.51	191.449	5033
M9	0:2499	0	0	0	0	0	0.09	461.604	791,404	0.002	6650	1.33	168.908	5025
M10	0:2499	0	0	0	0	0	0.05	231.236	1,056,995,742	4.359	276	1.35	171.514	7501
M11	0:3	0	0	0	0	0	0.53	4.306	245,253,112	54.315	351	0.81	0.165	60
M12	0:3	0	0	0	0	0	0.38	3.065	132,058,908	41.095	68	0.03	0.006	24
M13	0	0	0	0	0	0	0.02	0.034	1,275,584	35.799	56	0.05	0.002	6
M14	0	0	0	0	0	0	0.01	0.013	2,369,796	177.234	312	0.01	0.001	15
M15	0	0	0	0	0	0	0.00	0.003	132,624	36.152	648	0.01	0.001	28
M16	0	0	0	0	0	0	0.00	0.002	150,080	67.359	336	0.01	0.001	16
M17	0	0	0	0	0	0	0.00	0.000	74,580	169.832	56	0.01	0.001	6
M18	0	0	0	0	0	0	0.00	0.000	33,924	116.803	56	0.01	0.001	6
M19	0:2499	52.84	46018.207	16,944,108,416	0.351	37500	0	0	0	0	0	8.85	1124.747	25000
M20	0:2499	5.83	5079.783	1,057,315,552	0.198	17500	0	0	0	0	0	7.06	897.779	17500
M21	0:2499	5.65	4922.981	1,057,315,552	0.205	17500	0	0	0	0	0	8.61	1095.296	17500
M22	0:2499	5.58	4860.482	1,057,315,552	0.207	17500	0	0	0	0	0	10.24	1302.176	17500
M23	0:2499	5.36	4670.732	1,057,315,552	0.216	17500	0	0	0	0	0	9.63	1224.194	17500
M24	0:2499	5.16	4491.221	1,057,315,552	0.225	17500	0	0	0	0	0	7.58	963.684	17500
M24	0:2499	5.14	4476.999	1,057,315,552	0.225	17500	0	0	0	0	0	7.21	916.270	17500
M25	0:2499	5.01	4365.792	1,057,315,552	0.231	17500	0	0	0	0	0	6.84	870.168	17500
M27	0:2499	4.83	4210.032	1,057,315,552	0.240	17500	0	0	0	0	0	6.89	876.195	17500
M28	0:2499	4.58	3986.756	1,057,315,552	0.253	17500	0	0	0	0	0	6.52	828.806	17500
Total		**14.31**	**87083.0**	**26.459 GB**		**195000**	83.60	**75809.76**	**491.370 GB**		**167365**	2.09	**12498.6**	**427841**

Table 10: Per file read/write/meta activity as observed for MPI. Times indicated are the sum of the time spend by all ranks in that particular operation. Rel indicates the perceptual share of that file in its read/write/meta category, time-wise. The orange tags are files also present in the POSIX table.

What To Do With I/O Traces? **Solve!**





- **Hot file show a pathological contention pattern**

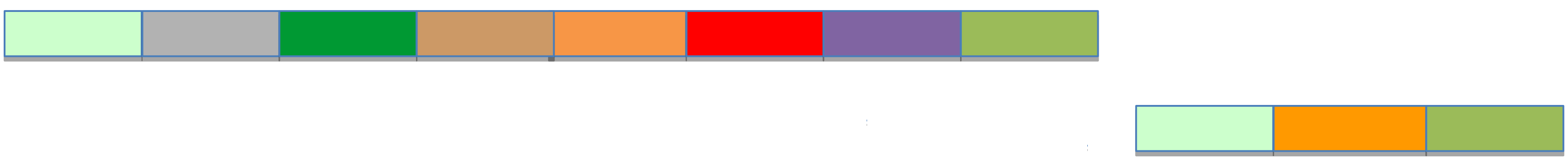
What To Do With I/O Traces? Solve!

- Replication of the I/O pattern in Düsseldorf /Paris lab.
- HW optimization (IME) improves performance by x 64
- SW optimization (MPI-IO rewriting/ tuning) improve performance by x 1.2 (20%)

What To Do With I/O Traces? Replayer

- **Instead of rewriting application I/O kernel replay its trace !**
- **Application characterization on large trace (2000+ ranks)**
  - Isolate critical files
- **Estimate control (metadata + synchronization) vs data cost**
  - Distribution: metadata = 18.89%, barrier = 1.46%, data = 79.64%
- **Search for symmetry**
- **Maintain group weight in downsizing**
- **Performance offset shift to avoid discontinuity artifact**

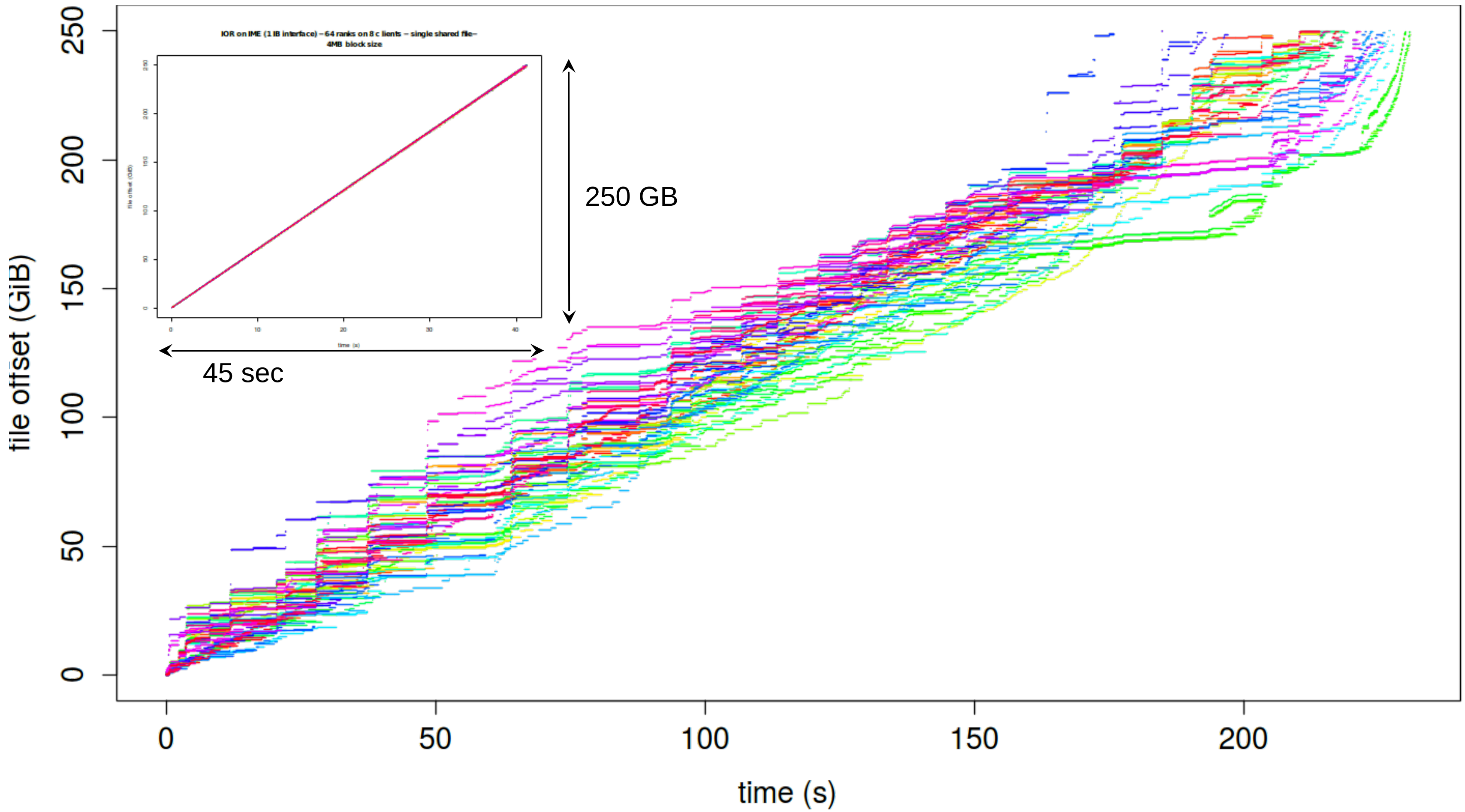
$$W_i = \frac{\#ranks\ in\ group\ i}{total\ \#ranks}$$



Replayer for architectural investigation

IME vs Spectrum Scale

IOR single shared file – 64 ranks on 8 clients 4MB block size



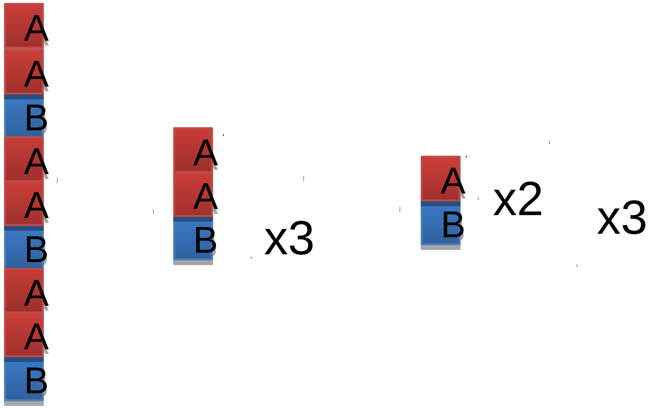
How to generate an I/O signature

► Structured part

- Exploiting the deterministic behavior
- Noise needs to be removed beforehand
- Grammar analysis → identifying structure in sequences
  - Nested Loop Recognition (University Of Strasbourg)
  - Sequitur (Google, University of Waikato)

► Random part

- Quantized via statistical analysis
  - Continuous-time Markov chain modeling
  - Long-range dependency matching
  - Use of robust statistical methods



► Standardized I/O signature formulation

DIO-pro an on-going effort

- Complexity is becoming unmanageable
- New app. challenge *our* (mine) expertise
- Fill a gap in the performance investigation stack
  - Full application
  - → We are here !
  - Mini-app
  - Synthetic kernel
  - Analytical mode

isc-hpc-io.org

ISC Workshops

June 22, 2017 | Frankfurt, Germany

WOPSSS

HPC I/O in the Data Center

Join us for a day of workshops dedicated to I/O at ISC High Performance 2017 on June 22, 2017.

WOPSSS

The Workshop On Performance and Scalability of Storage Systems (WOPSSS) aims to present state-of-the-art research, innovative ideas, and experience that focus on the design and implementation of HPC storage systems in both academic and industrial worlds, with a special interest on their performance analysis.

W·O·P·S·S·S

WORKSHOP ON PERFORMANCE AND SCALABILITY OF STORAGE SYSTEMS

The arrival of new storage technologies and scales unseen in previous practice lead to significant loss of performance predictability. This will leave storage system designers, application developers and the storage community at large in the difficult situation of not being able to precisely detect bottlenecks, evaluate the room for improvement, or estimate the matching of applications with a given storage architecture. WOPSSS intends to encourage discussion of these issues through submissions of researchers or practitioners from both academic and industrial worlds.

All accepted papers will be published in the Proceedings by Springer Extended versions of the best papers will be published in the ACM SIGOPS (http://www.sigops.org/osr.html) journal Papers need to be submitted via Easychair (http://easychair.org/conferences/?conf=wopsss2017).

Submission Deadline: March 31
Workshop: June 22

I/O in the Data Center Workshop

Managing scientific data at large scale is challenging for scientists but also for the host data center. The storage and file systems deployed within a data center are expected to meet users' requirements for data integrity and high performance across heterogeneous and concurrently running applications.

HPC I/O DC

With new storage technologies and layers in the memory hierarchy, the picture

Thank You!

Keep in touch with us

sales@ddn.com

@ddn_limitless

company/datadirect-networks

9351 Deering Avenue
Chatsworth, CA 91311

1.800.837.2298
1.818.700.4000