1. **Compact NUMA-aware Locks, Dave Dice, Alex Kogan, Oracle labs, 2019**
   https://arxiv.org/abs/1810.05600
   https://arxiv.org/pdf/1810.05600.pdf
   https://linuxplumbersconf.org/event/11/contributions/966/

2. **EUROSYS 2021**
   https://2021.eurosys.org/papers.html#papers

   Przykładowe propozycje:
   - On-Demand-Fork: A Microsecond Fork for Memory-Intensive and Latency-Sensitive Applications
   - Memory-Mapped I/O on Steroids

3. **What is New in the Linux kernel**

   Przykładowe propozycje:
   - https://lwn.net/Articles/868070/ - More IOPS with BIO caching
   - https://lwn.net/Articles/838339/ - Scheduling for asymmetric Arm systems
     Core scheduler support for scheduling on asymmetric systems has been merged. There is another piece to make use of this functionality on Arm processors that is presumably coming later in the merge window.
     https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=702f43872665
   - https://lwn.net/Articles/862707/ - NUMA policy and memory types
   - https://lwn.net/Articles/862018/ - Rust for Linux redux
   - https://lwn.net/Articles/861251/ - Core scheduling lands in 5.14
   - https://lwn.net/Articles/857133/ - Top-tier memory management
   - https://lwn.net/Articles/856931/ - Multi-generational LRU: the next generation
   - https://lwn.net/Articles/855226/ -a pair of memory-allocation improvements in 5.13

4. **Concurrency bugs should fear the big bad data-race detector**
   https://lwn.net/Articles/793253/
   https://lwn.net/Articles/802128/
   https://lwn.net/Articles/816850/
   https://lwn.net/Articles/816854/
   When compiling Linux-kernel code that does a plain C-language load or store the C standard grants the compiler the right to assume that the affected variables are neither accessed nor modified by any other thread at the time of that load or store. The compiler is therefore permitted to carry out a large number of transformations. However, our increasingly aggressive modern compilers produce increasingly surprising code optimizations. Some of these optimizations might be especially surprising to developers who assume that each plain C-language load or store will always result in an assembly-language load or store.
   Kernel Concurrency Sanitizer (KCSAN) can locate data races across the entire Linux kernel. This wide scalability does not come for free: KCSAN relies on compiler instrumentation and performs its analysis at runtime, which slows down the kernel considerably. In addition, it can only report data races that actually happen or almost happen during code execution. Nevertheless, KCSAN has already pointed out numerous problems, many of which have now been fixed.

4. **RCU**
   https://www.youtube.com/watch?v=hZX1aokdNiY
   RCU's First-Ever CVE, and How I Lived to Tell the Tale, Paul McKanney, 2019, Linux Conf (**CVE**, short for Common Vulnerabilities and Exposures, is a list of publicly disclosed computer security flaws)

5. **Storage Developers Conference**
   https://storagedeveloper.org/events/sdc-2021 (wkrótce podlinkuję stronę z prezentacjami z tegorocznej konferencji, September 28, 2021)
   https://www.snia.org/events/storage-developer/presentations2020
   https://www.snia.org/events/storage-developer/presentations19

   Przykładowe propozycje:
   - https://www.snia.org/sites/default/files/SDC/2019/presentations/NVMe/Hellwig_Christoph_Linux_NVMe_and_Block_Layer_Status_Update.pdf
   - https://www.snia.org/sites/default/files/SDC/2019/presentations/General_Session/Bouzeid_Gilbert_eBay_Object_Storage_at_Scale.pdf

6. **FAST'2021, Proceedings of the 19th USENIX Conference on File and Storage Technologies, February 2021**
   https://www.usenix.org/conference/fast21/technical-sessions
   (sporo prezentacji na YouTube)

   Przykładowe propozycje:
   - https://www.youtube.com/watch?v=aN2KysHRsZA, Facebook's Tectonic Filesystem: Efficiency from Exascale

7. **Linux Plumbers Conference 2021 (20/24 September)**
   https://www.linuxplumbersconf.org/event/11/
   https://linuxplumbersconf.org/event/11/timetable/?view=lpc

   Przykładowe propozycje:
   - https://linuxplumbersconf.org/event/11/contributions/954/ - Use of eBPF in CPU scheduler
   - https://linuxplumbersconf.org/event/11/contributions/1117/ - Challenge of selecting an idle CPU
   - https://linuxplumbersconf.org/event/11/contributions/986/ - Rust for Linux (więcej prezentacji nt Rust), https://linuxplumbersconf.org/event/11/contributions/1034/
   - https://linuxplumbersconf.org/event/11/contributions/976/ - Efficient buffered I/O
   - https://linuxplumbersconf.org/event/11/contributions/896/ - Overview of memory reclaim in the current upstream kernel
   - https://linuxplumbersconf.org/event/11/contributions/941/ - Towards a BPF Memory Model (Paul McKenney)
   - https://linuxplumbersconf.org/event/11/contributions/989/ - Building a fast nvme passthrough

8. **GitHub Copilot**
   https://copilot.github.com/
   https://lwn.net/Articles/862769/ - GitHub in my copilot

AI pair programmer. With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor. GitHub Copilot works with a broad set of frameworks and languages. The technical preview does especially well for Python, JavaScript, TypeScript, Ruby, and Go, but it understands dozens of languages.

9. **Prace z Research at Google, Distributed Systems and Parallel Computing**
   https://research.google/research-areas/distributed-systems-and-parallel-computing/
   Zdecydowana większość prac bardzo dobrze pasuje do tematyki seminarium, oczywiście lepsze są te najnowsze, z ostatniego roku. Przykładowo:

   - https://research.google/pubs/pub50648/
     The Monolith Strikes Back: Why Istio Migrated from Microservices to a Monolithic Architecture
   - https://research.google/pubs/pub49174/
     Autopilot: Workload Autoscaling at Google Scale
   - https://research.google/pubs/pub49065/
     Borg: the Next Generation
   - https://ai.google/research/pubs/pub48000
     Designing and Operating Highly Available Software Systems at Scale
   - https://research.google/pubs/pub48413/
     Code coverage at Google
   - https://research.google/pubs/pub48392/
     The DevOps Phenomenon

Zeszłoroczne propozycje, nadal aktualne

10. **Artykuły z 2020 USENIX Annual Technical Conference (podaję kilka przykładowych, ale w materiałach z konferencji jest dużo innych ciekawych, właściwie każdy się nadaje do prezentacji)**

    https://www.usenix.org/conference/atc20/technical-sessions
    https://2459d6dc103cb5933875-c0245c5c937c5dedcca3f1764ecc9b2f.ssl.cf2.rackcdn.com/atc20_full_proceedings.pdf

    - https://www.usenix.org/conference/atc20/presentation/bittman (best presentation)
      Twizzler: a Data-Centric OS for Non-Volatile Memory
    - https://www.usenix.org/conference/atc20/presentation/keynote-miller
      The Future of the Past: Challenges in Archival Storage
    - https://www.usenix.org/conference/atc20/presentation/gouicern
      Fewer Cores, More Hertz: Leveraging High-Frequency Cores in the OS Scheduler for Improved Application Performance
    - https://www.usenix.org/conference/atc20/presentation/zhu-weixi
      A Comprehensive Analysis of Superpage Management Mechanisms and Policies
    - https://www.usenix.org/conference/atc20/presentation/al-maruf (best paper)
      Effectively Prefetching Remote Memory with Leap

11. **FAST'2020, Proceedings of the 18th USENIX Conference on File and Storage Technologies**

    https://www.usenix.org/conference/fast20/technical-sessions
    https://www.usenix.org/sites/default/files/fast20_full-proceedings.pdf

    Wiele ciekawych prac, przykładowo:
    - https://www.usenix.org/conference/fast20/presentation/zhan
      How to Copy Files
    - https://www.usenix.org/conference/fast20/presentation/he
      Read as Needed: Building WiSER, a Flash-Optimized Search Engine
    - https://www.usenix.org/conference/fast20/presentation/maneas (best paper)
      A Study of SSD Reliability in Large Scale Enterprise Storage Deployments
    - https://www.usenix.org/conference/fast20/presentation/lu
      Making Disk Failure Predictions SMARTer!
    - https://www.usenix.org/conference/fast20/presentation/wang-ao
      InfiniCache: Exploiting Ephemeral Serverless Functions to Build a Cost-Effective Memory Cache
    - https://www.usenix.org/conference/fast20/presentation/kumar
      Quiver: An Informed Storage Cache for Deep Learning
    - https://www.usenix.org/conference/fast20/presentation/ganesan (best paper)
      Strong and Efficient Consistency with Consistency-Aware Durability

12. **EUROSYS 2020**
    https://www.eurosys2020.org/program-2/accepted-papers/
    Wiele ciekawych prac, przykładowo

    - Don't shoot down TLB shootdowns! (best paper)
      https://dl.acm.org/doi/abs/10.1145/3342195.3387518

13. **Artykuły z 2019 USENIX Annual Technical Conference (podaję kilka przykładowych, ale w materiałach z konferencji jest dużo innych ciekawych, właściwie każdy się nadaje do prezentacji)**

    https://www.usenix.org/conference/atc19/technical-sessions
    https://www.usenix.org/sites/default/files/atc19-full-proceedings.pdf

    - Everyone Loves File: File Storage Service (FSS) in Oracle Cloud Infrastructure
    - Not So Fast: Analyzing the Performance of WebAssembly vs . Native Code
    - Extension Framework for File Systems in User space
    - FlexGroup Volumes: A Distributed WAFL File System
    - Effective Static Analysis of Concurrency Use-After-Free Bugs in Linux Device Drivers
    - LXDs: Towards Isolation of Kernel Subsystems
    - Multi-Queue Fair Queuing
    - BRAVO – Biased Locking for Reader-Writer Locks
    - Asynchronous I/O Stack: A Low-latency Kernel I/O Stack for Ultra-Low Latency SSDs
    - Data Domain Cloud Tier: Backup here, backup there, deduplicated everywhere!

- GAIA: An OS Page Cache for Heterogeneous Systems
- ElasticBF: Elastic Bloom Filter with Hotness Awareness for Boosting Read Performance in Large Key-Value Stores
- Lessons and Actions:What We Learned from 10K SSD-Related Storage System Failures

14. **Propozycje Andrzeja Jackowskiego (9Livesdata)**
    Wydaje mi się, że najlepiej na seminarium nadaje się publikacja o której nie mówiłem, czyli "Orion: A Distributed File System for Non-Volatile Main Memory and RDMA-Capable Networks" (https://www.usenix.org/conference/fast19/presentation/yang).

    Z pozostałymi publikacjami jest trochę taki problem, że ciężko będzie o nich ciekawie mówić nawet przez 45 minut. Ale jeśli ktoś będzie chciał poszukać podobnych publikacji i opowiedzieć o czymś więcej niż jednym artykule, to dobrymi kandydatami są moim zdaniem:

    "INSTalytics: Cluster Filesystem Co-design for Big-data Analytics" (https://www.usenix.org/conference/fast19/presentation/sivathanu) - tutaj detale są nietrywialne i można ciekawie zahaczyć o istniejące systemy BigData (np. Hadoop, Spark)

    "Automatic, Application-Aware I/O Forwarding Resource Allocation" (https://www.usenix.org/conference/fast19/presentation/ji) - sam artykuł może nie był super ciekawy, ale opowiada o komputerze stojącym na podium listy Top500. Jest to dobra podstawa do fajnej prezentacji o superkomputerach.

    "GearDB: A GC-free Key-Value Store on HM-SMR Drives with Gear Compaction" (https://www.usenix.org/conference/fast19/presentation/yao) - z jednej strony dotyczy SMR dysków, które moim zdaniem są ciekawe koncepcyjnie, z drugiej strony dotyka struktur danych LogStructuredMergeTree-podobnych.

15. **FAST'2019, Proceedings of the 17th USENIX Conference on File and Storage Technologies**
    https://www.usenix.org/conference/fast19/technical-sessions
    https://www.usenix.org/sites/default/files/fast19-full-proceedings.pdf
    Wiele ciekawych prac, przykładowo:

    - Design Tradeoffs for SSD Reliability
    - ScaleCheck: A Single-Machine Approach forDiscovering Scalability Bugs in Large Distributed Systems

16. **Marteen van Steen (University of Twente)**
    https://www.distributed-systems.net/index.php/research/217-2/
    https://www.distributed-systems.net/index.php/research/current-projects/

17. **Andrew Tanenbaum (Vrije Universiteit, retired)**
    http://www.cs.vu.nl/~ast/Publications/index.html#papers

18. **The Redmond Distributed Systems Research Group**
    https://www.microsoft.com/en-us/research/group/systems-research-group-redmond/

19. **Microsoft Research Systems & Networking Publications**
    https://www.microsoft.com/en-us/research/research-area/systems-and-networking/