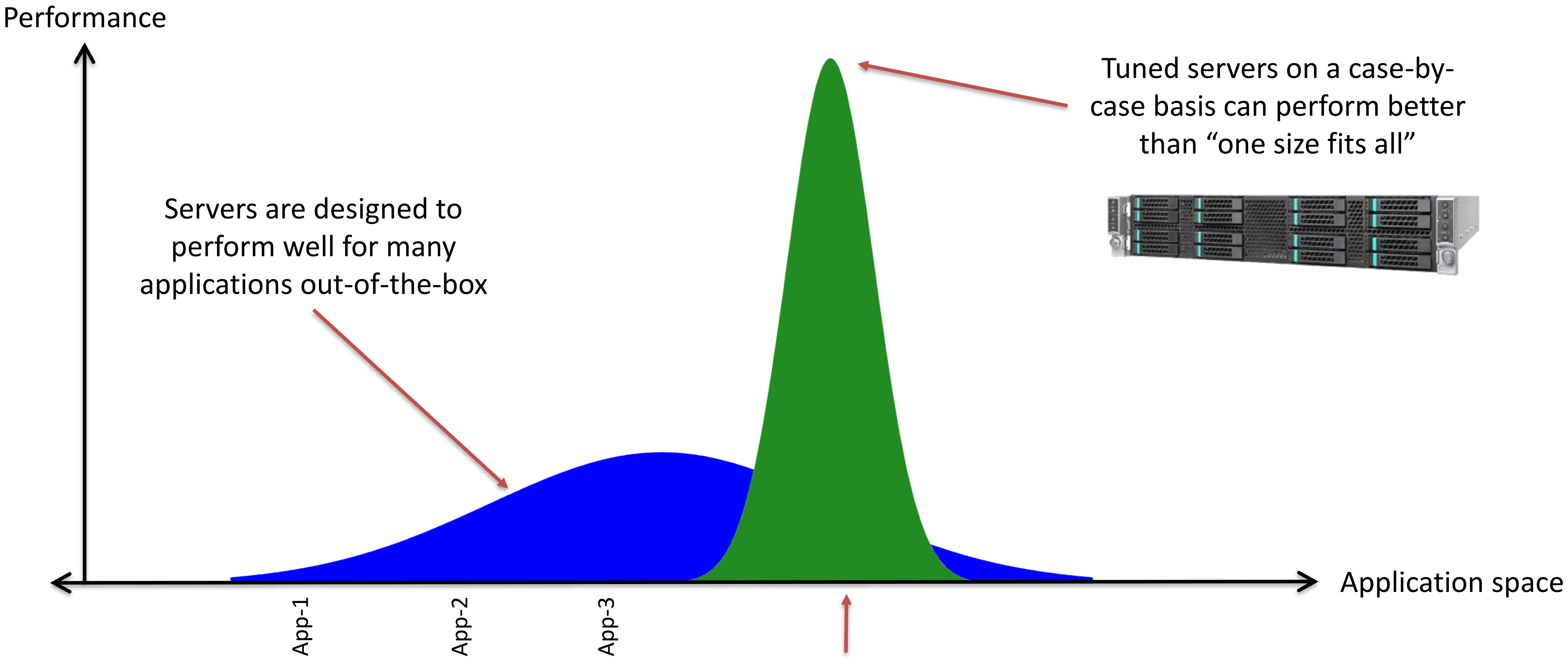
The Era of Self-Tuning Servers

February 7, 2017

DatArcs

Dr. Tomer Morad, CEO
tomer.morad@datarcs.com
www.datarcs.com

Intuition



Servers are designed to perform well for many applications out-of-the-box

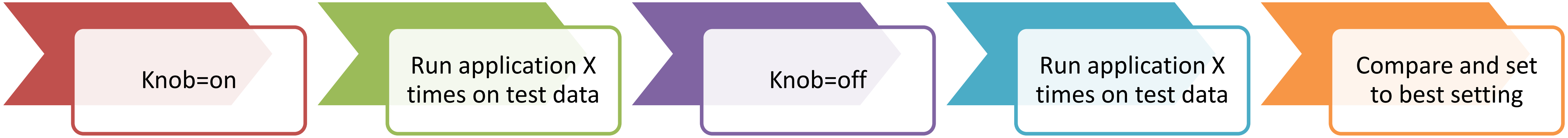
Tuned servers on a case-by-case basis can perform better than “one size fits all”

Only specific applications actually run in production

Introduction to Tuning

- ❑ Knobs (in this talk) represent settings on a server that:
  - ❑ Can be changed in real time
  - ❑ Affects performance / energy efficiency
  - ❑ Retains correctness

- ❑ Tuning
  - ❑ The process of finding the best setting of a knob
  - ❑ Tuning example (one application, one on/off knob):



Knob=on

Run application X times on test data

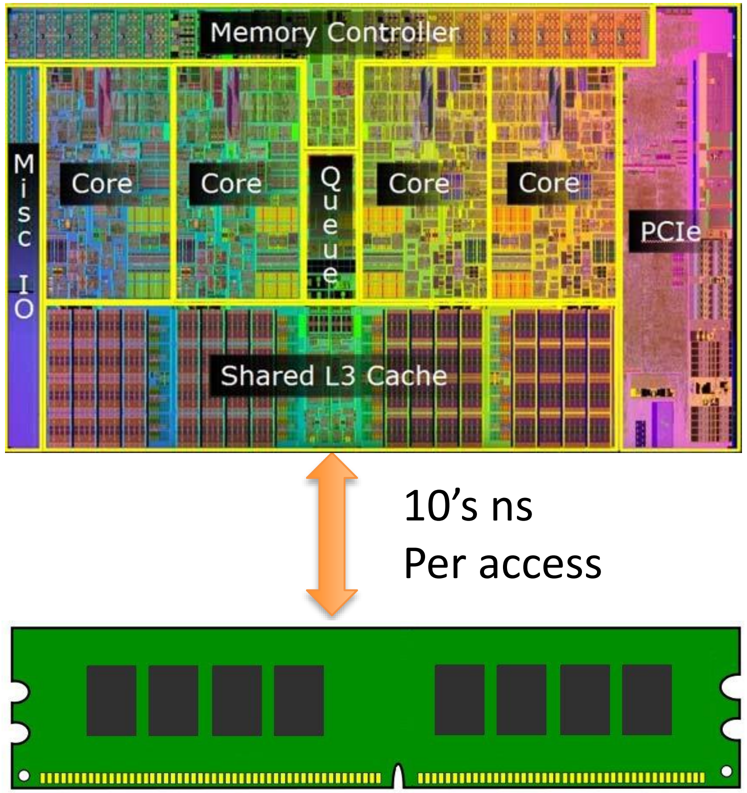
Knob=off

Run application X times on test data

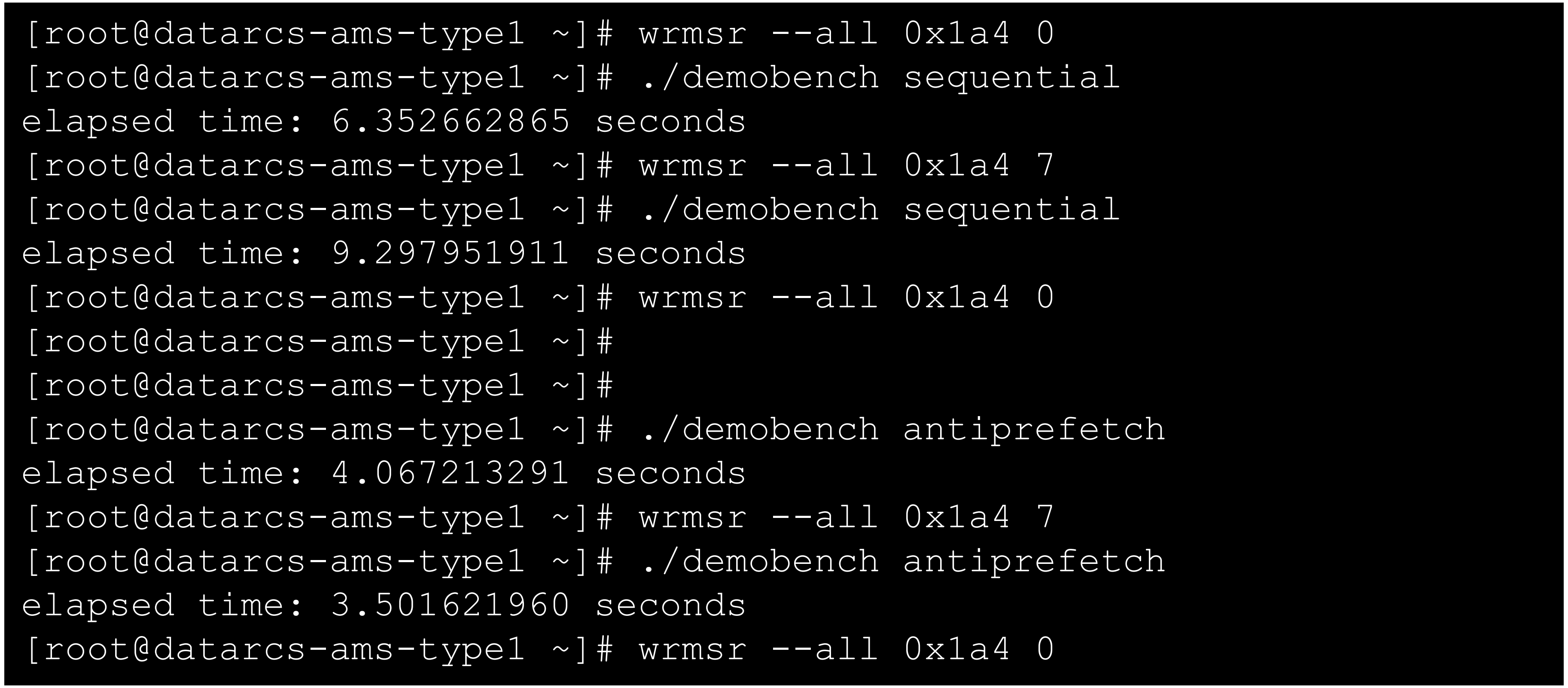
Compare and set to best setting

Example knob: CPU Cache Prefetching

- ❑ On-chip memory is 10x-100x faster than off-chip memory
- ❑ CPU Cache Prefetching – fetches data from off-chip memory **before** the CPU asks for it.
- ❑ Prefetcher for latency reduction:
    - ❑ ***Predicts*** which data the CPU will need in the future
    - ❑ ***Predicts*** which data the CPU will not require in the future
    - ❑ Fetches data that the prefetcher predicted the CPU will need and store it in the cache instead of data that the prefetcher predicted the CPU will not need



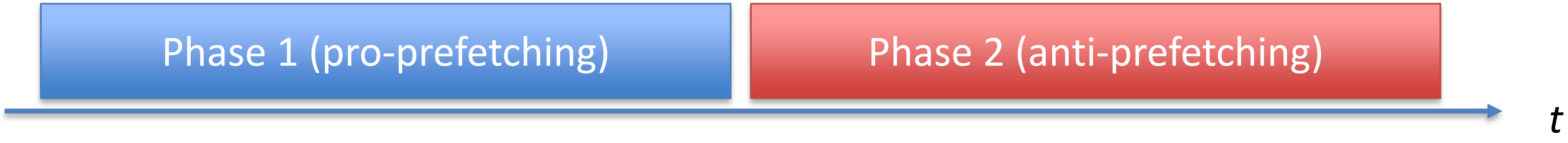
Tuning cache prefetching



```
[root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 0
[root@datarcs-ams-type1 ~]# ./demobench sequential
elapsed time: 6.352662865 seconds
[root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 7
[root@datarcs-ams-type1 ~]# ./demobench sequential
elapsed time: 9.297951911 seconds
[root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 0
[root@datarcs-ams-type1 ~]#
[root@datarcs-ams-type1 ~]#
[root@datarcs-ams-type1 ~]# ./demobench antiprefetch
elapsed time: 4.067213291 seconds
[root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 7
[root@datarcs-ams-type1 ~]# ./demobench antiprefetch
elapsed time: 3.501621960 seconds
[root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 0
```

Program phases

- Programs have phases, for example:

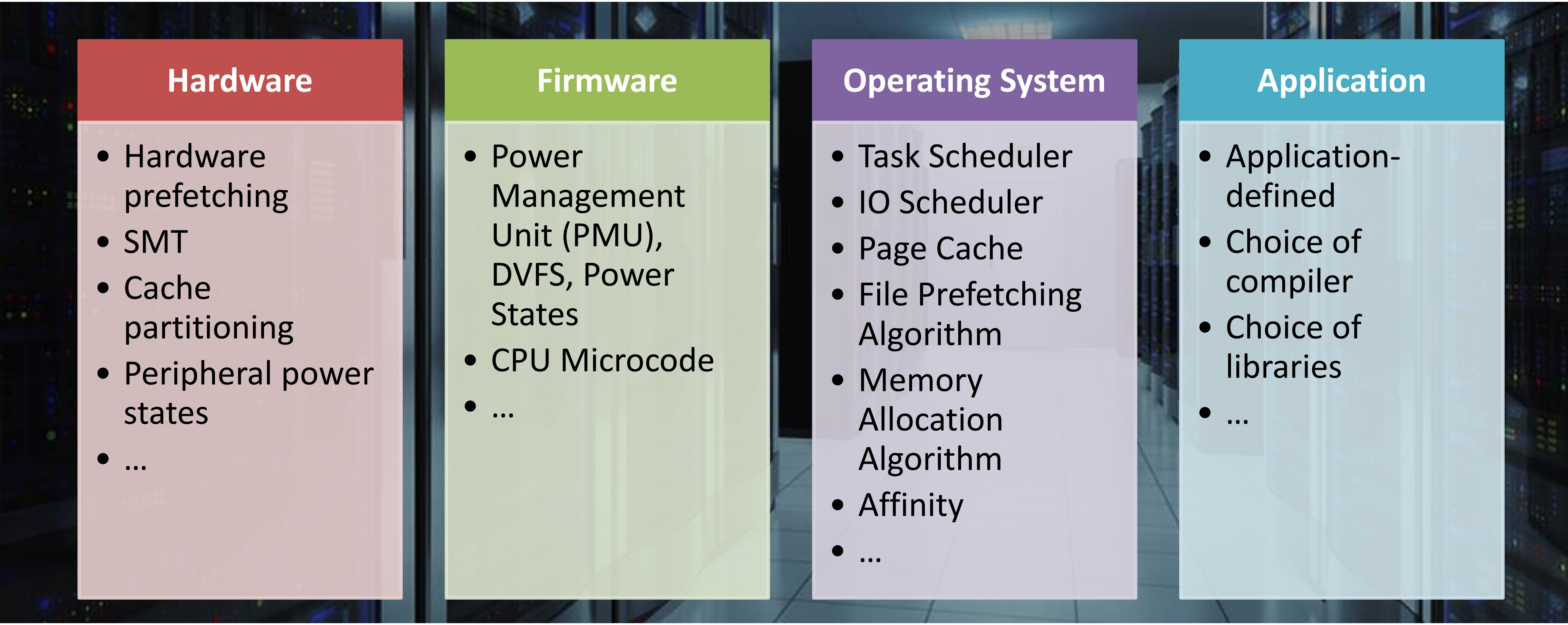


- In this example:

```
 [root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 0
 [root@datarcs-ams-type1 ~]# ./demobench phases
elapsed time: 55.852871715 seconds
 [root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 7
 [root@datarcs-ams-type1 ~]# ./demobench phases
elapsed time: 56.189048409 seconds
 [root@datarcs-ams-type1 ~]# wrmsr --all 0x1a4 0
```

- Can we do better?

Tunable Knobs in Today’s Systems



Hardware

- Hardware prefetching
- SMT
- Cache partitioning
- Peripheral power states
- …

Firmware

- Power Management Unit (PMU), DVFS, Power States
- CPU Microcode
- …

Operating System

- Task Scheduler
- IO Scheduler
- Page Cache
- File Prefetching Algorithm
- Memory Allocation Algorithm
- Affinity
- …

Application

- Application-defined
- Choice of compiler
- Choice of libraries
- …

100’s of different knobs to tune!

Some of the knobs (module-related) that can be tuned in Linux…

- 8250.nr_uarts
- 8250.share_irqs
- 8250.skip_txen_test
- acpi.aml_debug_output
- acpi.ec_busy_polling
- acpi.ec_delay
- acpi.ec_event_clearing
- acpi.ec_polling_guard
- acpi.ec_storm_threshold
- acpi.immediate_undock
- ahci.marvell_enable
- apparmor.audit
- apparmor.audit_header
- apparmor.debug
- apparmor.lock_policy
- apparmor.logsyscall
- apparmor.mode
- apparmor.paranoid_load
- apparmor.path_max
- battery.cache_time
- cfg80211.cfg80211_disable_40mhz_24ghz
- debug_core.kgdbreboot
- debug_core.kgdb_use_con
- dm_mod.dm_mq_nr_hw_queues
- dm_mod.dm_mq_queue_depth
- dm_mod.dm_numa_node
- dm_mod.reserved_bio_based_ios
- dm_mod.reserved_rq_based_ios
- dm_mod.use_blk_mq
- drm_kms_helper.dp_aux_i2c_speed_khz
- drm_kms_helper.dp_aux_i2c_transfer_size
- drm_kms_helper.edid_firmware
- drm_kms_helper.fbdev_emulation
- drm_kms_helper.poll
- drm.debug
- drm.timestamp_monotonic
- drm.timestamp_precision_usec
- drm.vblankoffdelay
- dynamic_debug.verbose
- efi_pstore.pstore_disable
- firmware_class.path
- fuse.max_user_bgreq
- fuse.max_user_congthresh
- hid.debug
- hid.ignore_special_drivers
- i8042.debug
- i8042.unmask_kbd_data
- i915.enable_cmd_parser
- i915.enable_fbc
- i915.enable_hangcheck
- i915.enable_ips
- i915.enable_psr
- i915.fastboot
- i915.invert_brightness
- i915.load_detect_test
- i915.lvds_use_ssc
- i915.mmio_debug
- i915.nuclear_pageflip
- i915.panel_ignore_lid
- i915.prefault_disable
- i915.reset
- i915.use_mmio_flip
- i915.verbose_state_checks
- ima.ahash_bufsize
- ima.ahash_minsize
- intel_powerclamp.duration
- intel_powerclamp.window_size
- kdb.cmd_enable
- kdb.enable_nmi
- kernel.ignore_rlimit_data
- kernel.initcall_debug
- kernel.panic
- kernel.panic_on_warn
- kernel.pause_on_oops
- keyboard.brl_nbchords
- keyboard.brl_timeout
- kgdb_nmi.knock
- kgdb_nmi.magic
- kgdboc.kgdboc
- kvm.allow_unsafe_assigned_interrupts
- kvm.halt_poll_ns
- kvm.halt_poll_ns_grow
- kvm.halt_poll_ns_shrink
- kvm.ignore_msrs
- kvm.lapic_timer_advance_ns
- kvm.min_timer_period_us
- kvm.tsc_tolerance_ppm
- libahci.devslp_idle_timeout
- libata.acpi_gtf_filter
- libata.ignore_hpa
- libata.zpodd_poweroff_delay
- mac80211.beacon_loss_count
- mac80211.ieee80211_default_rc_algo
- mac80211.max_nullfunc_tries
- mac80211.max_probe_tries
- mac80211.minstrel_vht_only
- mac80211.probe_wait_ms
- module.sig_enforce
- mousedev.tap_time
- mousedev.xres
- mousedev.yres
- netpoll.carrier_timeout
- nf_conntrack_ipv4.hashsize
- nf_conntrack.acct
- nf_conntrack.hashsize
- nf_conntrack.nf_conntrack_helper
- nf_conntrack.tstamp
- overlay.check_copy_up
- parport_pc.verbose_probing
- pcie_aspm.policy
- pciehp.pciehp_debug
- pciehp.pciehp_force
- pciehp.pciehp_poll_mode
- pciehp.pciehp_poll_time
- pci_hotplug.debug
- pci_hotplug.debug_acpi
- pci_slot.debug
- ppp_generic.mp_protocol_compress
- printk.always_kmsg_dump
- printk.console_suspend
- printk.ignore_loglevel
- printk.time
- processor.ignore_ppc
- processor.ignore_tpc
- processor.latency_factor
- psmouse.proto
- psmouse.rate
- psmouse.resetafter
- psmouse.resolution
- psmouse.resync_time
- psmouse.smartscroll
- pstore.update_ms
- rcupdate.rcu_cpu_stall_suppress
- rcupdate.rcu_cpu_stall_timeout
- rcutree.jiffies_till_first_fqs
- rcutree.jiffies_till_next_fqs
- rcutree.jiffies_till_sched_qs
- rcutree.kthread_prio
- rng_core.current_quality
- rng_core.default_quality
- scsi_mod.default_dev_flags
- scsi_mod.eh_deadline
- scsi_mod.inq_timeout
- scsi_mod.max_luns
- scsi_mod.scsi_logging_level
- scsi_mod.use_blk_mq
- sg.allow_dio
- sg.def_reserved_size
- sg.scatter_elem_sz
- shpchp.shpchp_debug
- shpchp.shpchp_poll_mode
- shpchp.shpchp_poll_time
- snd_hda_codec_hdmi.static_hdmi_pcm
- snd_hda_codec.dump_coef
- snd_hda_intel.align_buffer_size
- snd_hda_intel.bdl_pos_adj
- snd_hda_intel.power_save
- snd_hda_intel.power_save_controller
- snd_seq_midi.input_buffer_size
- snd_seq_midi.output_buffer_size
- snd_seq.seq_default_timer_card
- snd_seq.seq_default_timer_class
- snd_seq.seq_default_timer_device
- snd_seq.seq_default_timer_resolution
- snd_seq.seq_default_timer_sclass
- snd_seq.seq_default_timer_subdevice
- spurious.irqfixup
- spurious.noirqdebug
- sr_mod.xa_test
- suspend.pm_test_delay
- sysrq.reset_seq
- sysrq.sysrq_downtime_ms
- tcp_cubic.beta
- tcp_cubic.fast_convergence
- tcp_cubic.hystart
- tcp_cubic.hystart_ack_delta
- tcp_cubic.hystart_detect
- tcp_cubic.hystart_low_window
- tcp_cubic.initial_ssthresh
- tcp_cubic.tcp_friendliness
- thermal.act
- thermal.crt
- thermal.psv
- tpm.suspend_pcr
- usbcore.authorized_default
- usbcore.autosuspend
- usbcore.initial_descriptor_timeout
- usbcore.old_scheme_first
- usbcore.usbfs_memory_mb
- usbcore.usbfs_snoop
- usbcore.usbfs_snoop_max
- usbcore.use_both_schemes
- usbhid.ignoreled
- usbhid.mousepoll
- usb_storage.delay_use
- usb_storage.option_zero_cd
- usb_storage.quirks
- usb_storage.swi_tru_install
- video.allow_duplicates
- video.brightness_switch_enabled
- video.report_key_events
- vt.color
- vt.cur_default
- vt.default_blu
- vt.default_grn
- vt.default_red
- vt.default_utf8
- vt.global_cursor_default
- vt.italic
- vt.underline
- workqueue.debug_force_rr_cpu
- workqueue.watchdog_thresh
- x86_pkg_temp_thermal.notify_delay_ms
- xhci_hcd.link_quirk
- zswap.compressor
- zswap.enabled
- zswap.max_pool_percent
- zswap.zpool

Limitations of manual tuning

- 100’s of knobs – too many to manually tune
- Dependencies among different knobs
- Knob settings depend on hardware
- Knob settings depend on applications and input data
- No practical way to “see” program phases
- Labor intensive task
- Requires expertise which is absent in most organizations

The Era of Self-Tuning Servers - DatArcs Optimizer©

Step 1: Install

Step 2: Learn

Step 3: Optimize

DatArcs Optimizer Advantages

Automatic

Optimizer tunes without any user input

Adaptive

Program phase and workload detection

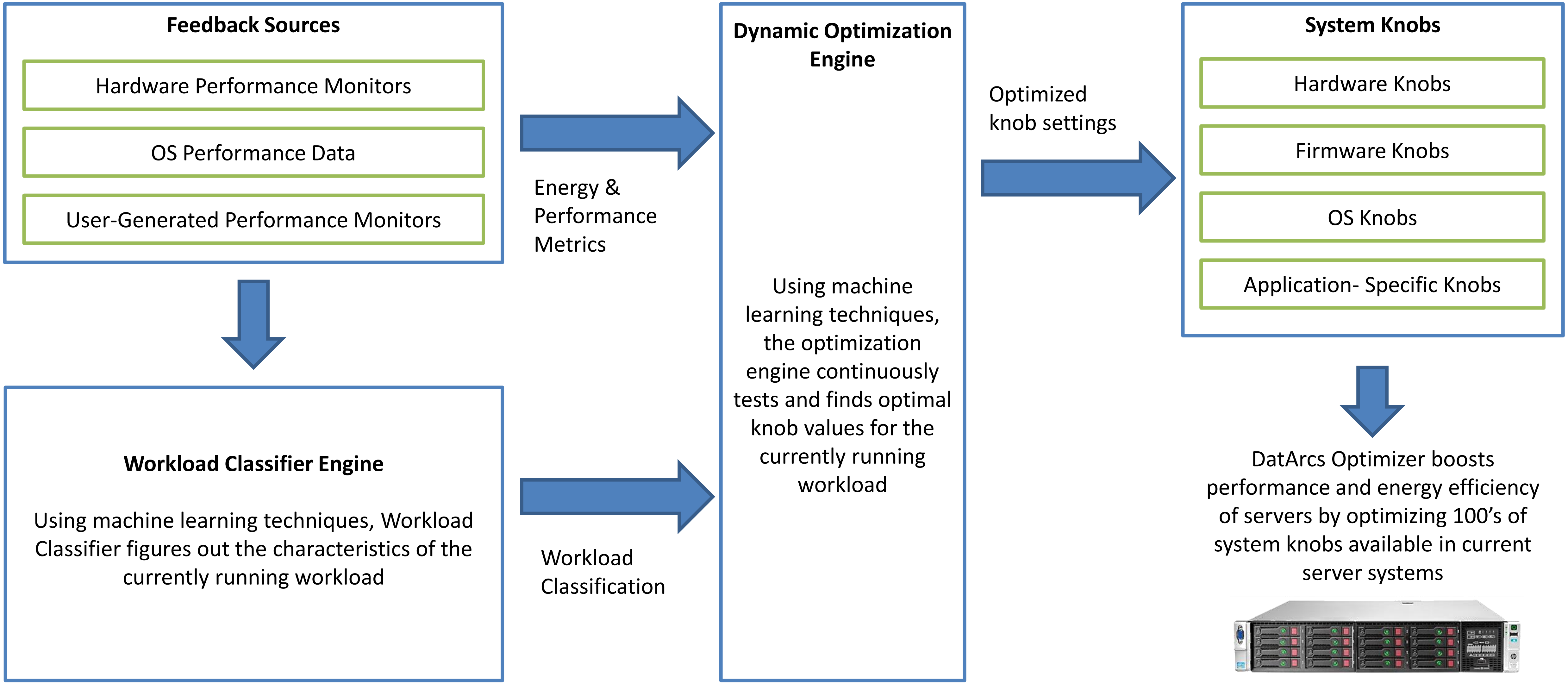
Flexible

Tunes for speed, energy, power, power cap, etc.

Extensible

Users can add application-specific knobs

DatArcs Optimizer In a Nutshell



Feedback Sources

- Hardware Performance Monitors
- OS Performance Data
- User-Generated Performance Monitors

Energy & Performance Metrics

Workload Classifier Engine

Using machine learning techniques, Workload Classifier figures out the characteristics of the currently running workload

Workload Classification

Dynamic Optimization Engine

Using machine learning techniques, the optimization engine continuously tests and finds optimal knob values for the currently running workload

Optimized knob settings

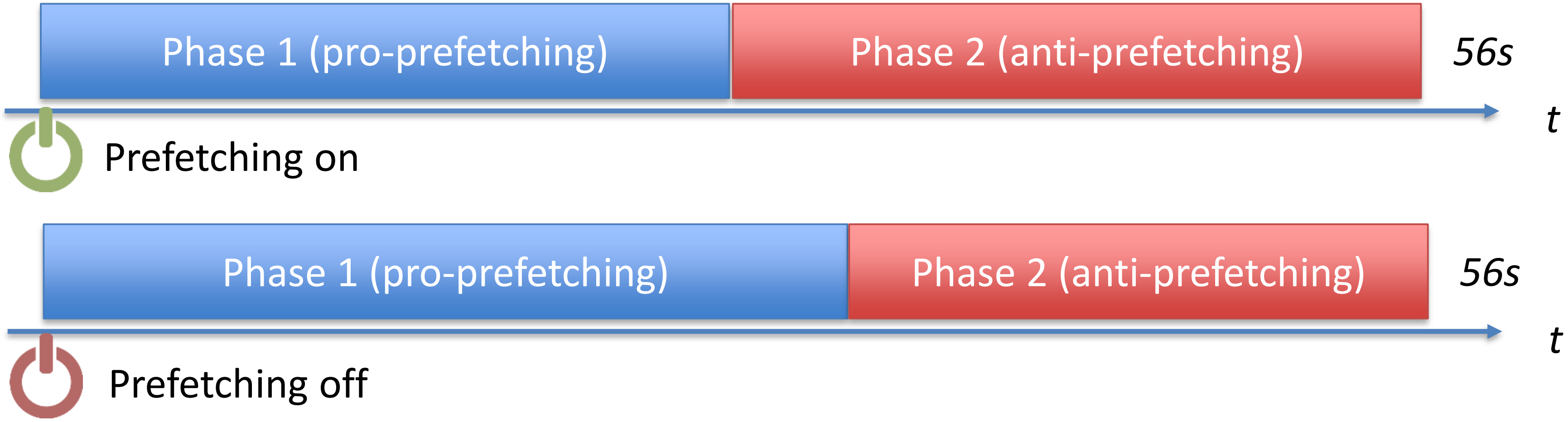
System Knobs

- Hardware Knobs
- Firmware Knobs
- OS Knobs
- Application- Specific Knobs

DatArcs Optimizer boosts performance and energy efficiency of servers by optimizing 100’s of system knobs available in current server systems

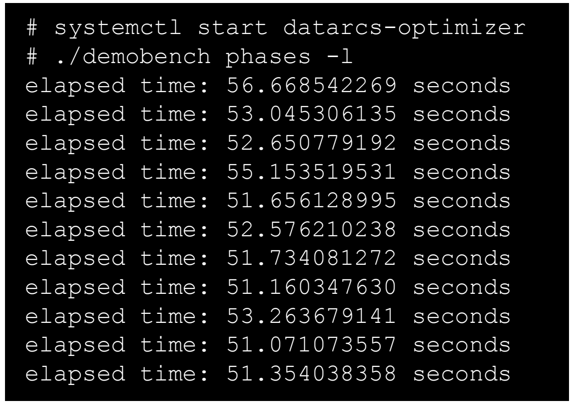
Dynamic Tuning demo

Manual Tuning

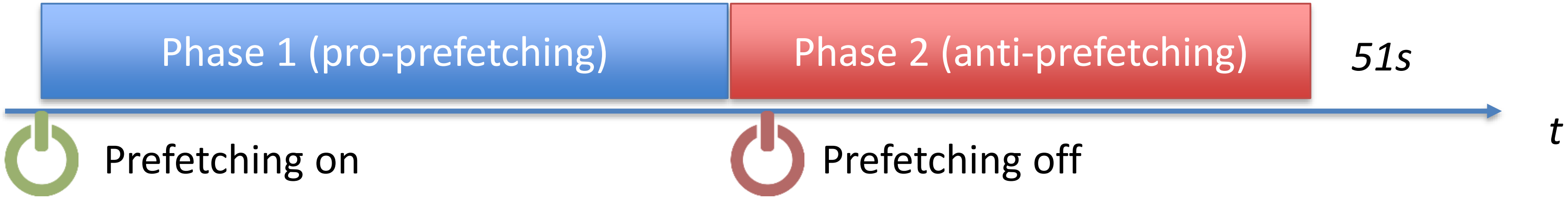


Phase 1 (pro-prefetching)
Phase 2 (anti-prefetching)
56s
t
Prefetching on
Phase 1 (pro-prefetching)
Phase 2 (anti-prefetching)
56s
t
Prefetching off

Dynamic Tuning

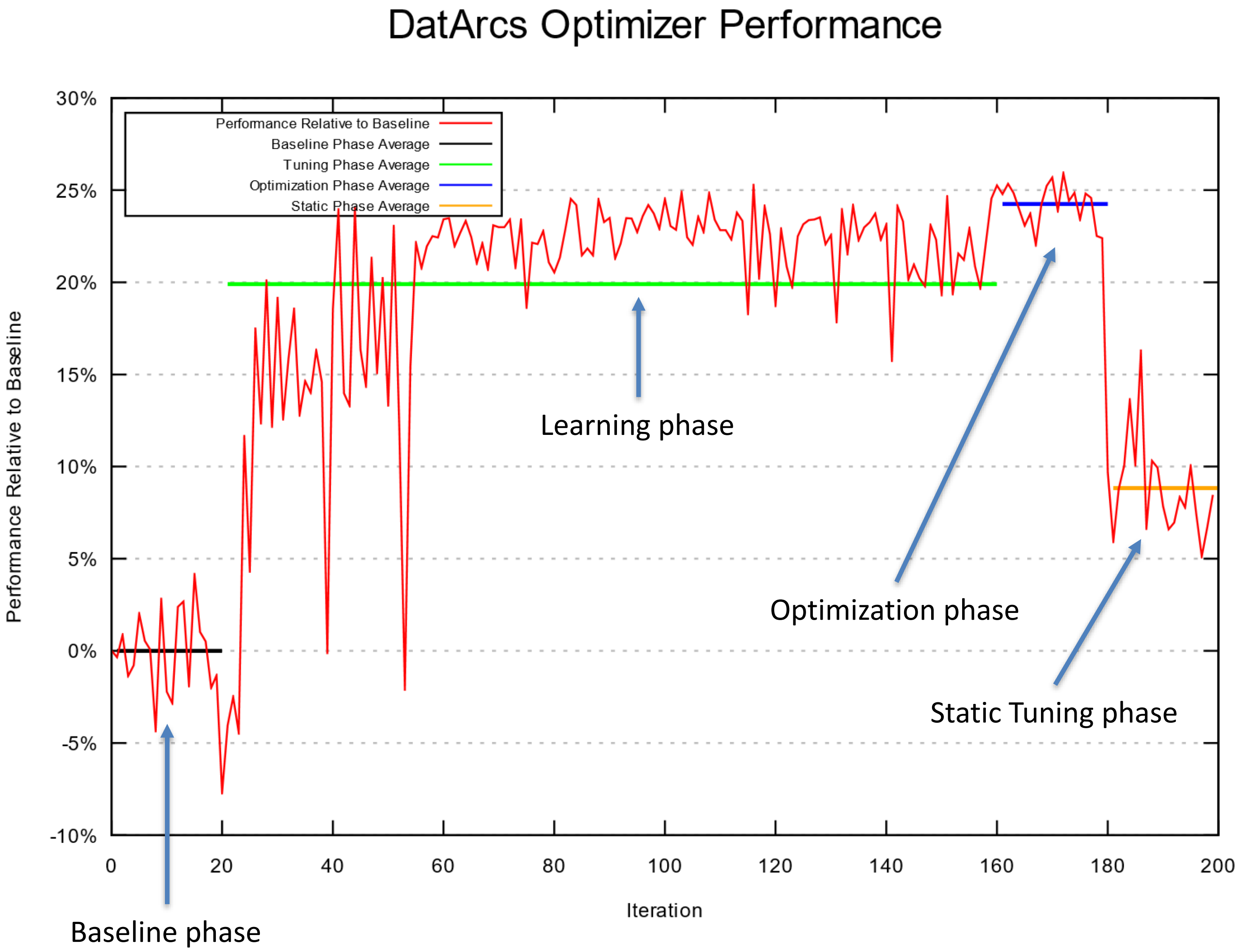


```
# systemctl start datarcs-optimizer
# ./demobench phases -l
elapsed time: 56.668542269 seconds
elapsed time: 53.045306135 seconds
elapsed time: 52.650779192 seconds
elapsed time: 55.153519531 seconds
elapsed time: 51.656128995 seconds
elapsed time: 52.576210238 seconds
elapsed time: 51.734081272 seconds
elapsed time: 51.160347630 seconds
elapsed time: 53.263679141 seconds
elapsed time: 51.071073557 seconds
elapsed time: 51.354038358 seconds
```



DatArcs Optimizer outperformed the manual method!

Apache Web Server @ packet type-2 servers



Benchmark

- Phoronix test of Apache web server

Experiment phases

- ***Baseline*** - 20 runs without Optimizer
- ***Learning*** - 140 runs with Optimizer in learning mode
- ***Optimization*** - 20 runs with Optimizer in optimization mode
- ***Static*** – 20 runs after Optimizer applied the best knob values and exited

Server Configuration

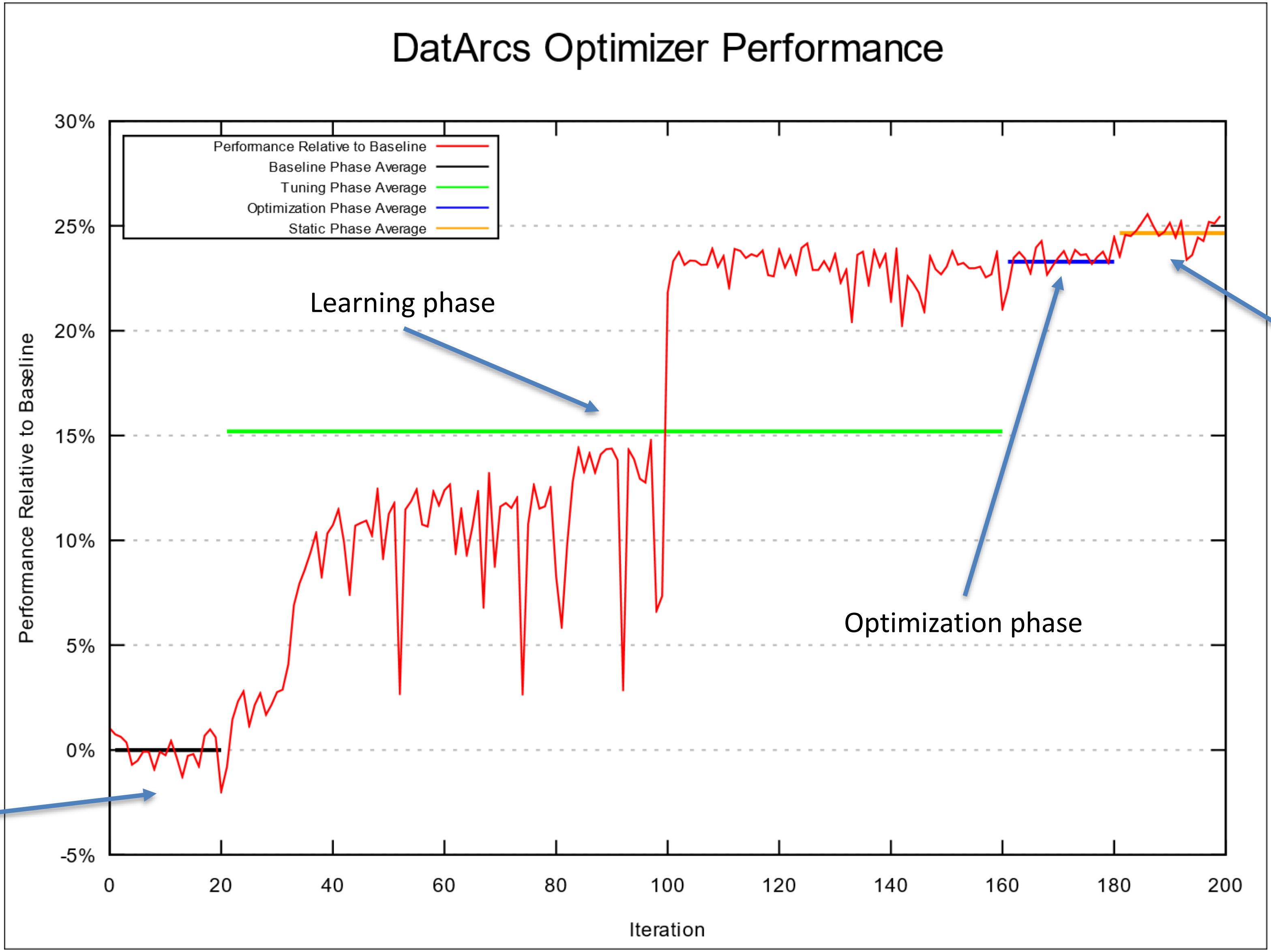
- Packet type-2 server: 2x Intel E5-2650 @ 2.2GHz, 24 cores, 48 threads

Results

- **DatArcs Optimizer correctly identified the effects of various knobs on the application performance, and achieved ~25% boost in performance**

Apache
HTTP SERVER PROJECT

Apache Web Server @ Intel E3-1270 V3 3.5Ghz (4 cores)



Learning phase

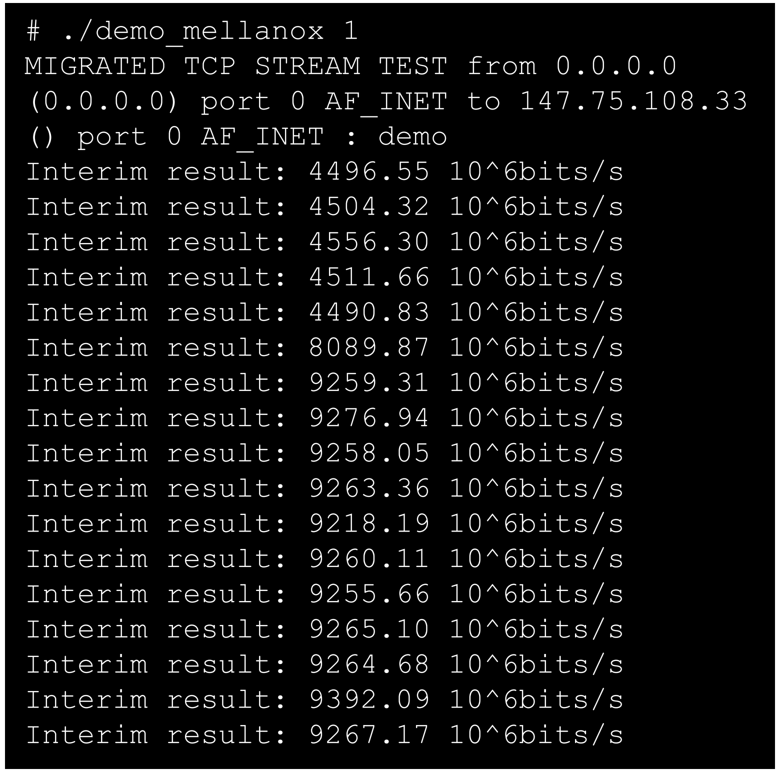
Optimization phase

Static Tuning phase

Baseline phase

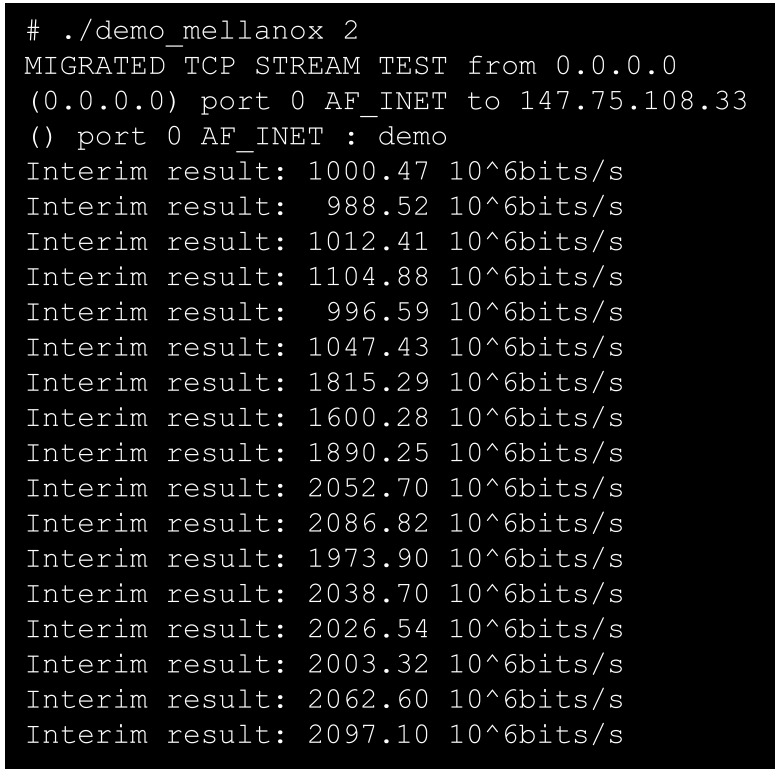
High-Speed Networking Results

- ❑ One knob: tx-usecs = (16 or 256)
- ❑ Two benchmarks, each prefers a different setting
- ❑ DatArcs Optimizer detects best setting for each benchmark



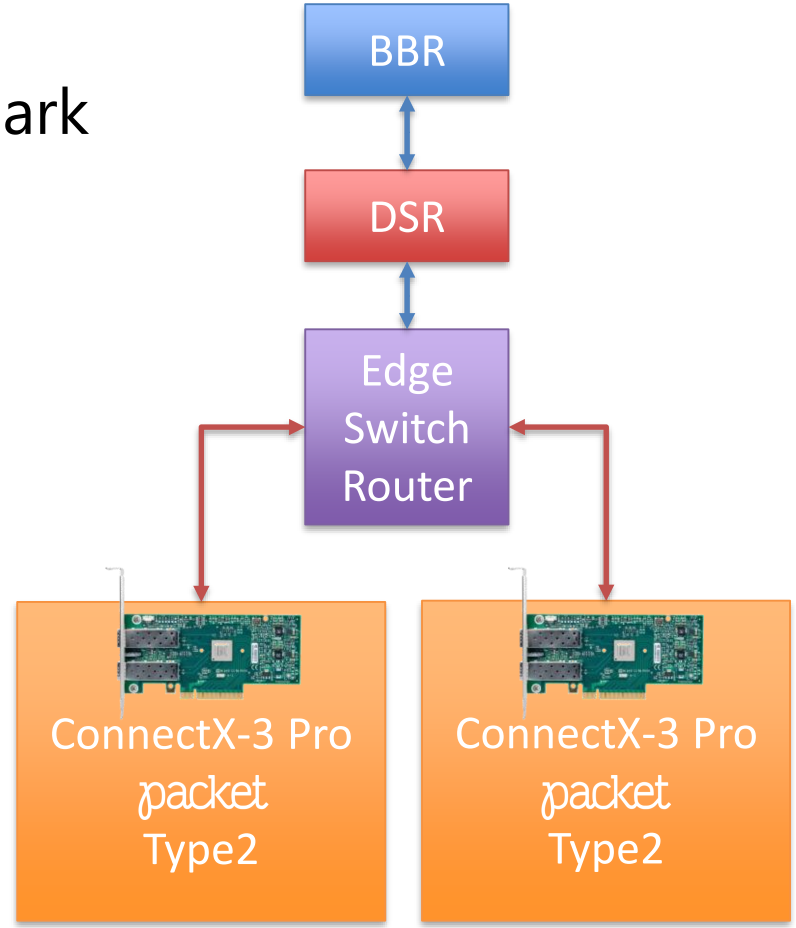
```
# ./demo_mellanox 1
MIGRATED TCP STREAM TEST from 0.0.0.0
(0.0.0.0) port 0 AF_INET to 147.75.108.33
() port 0 AF_INET : demo
Interim result: 4496.55 10^6bits/s
Interim result: 4504.32 10^6bits/s
Interim result: 4556.30 10^6bits/s
Interim result: 4511.66 10^6bits/s
Interim result: 4490.83 10^6bits/s
Interim result: 8089.87 10^6bits/s
Interim result: 9259.31 10^6bits/s
Interim result: 9276.94 10^6bits/s
Interim result: 9258.05 10^6bits/s
Interim result: 9263.36 10^6bits/s
Interim result: 9218.19 10^6bits/s
Interim result: 9260.11 10^6bits/s
Interim result: 9255.66 10^6bits/s
Interim result: 9265.10 10^6bits/s
Interim result: 9264.68 10^6bits/s
Interim result: 9392.09 10^6bits/s
Interim result: 9267.17 10^6bits/s
```

+106%



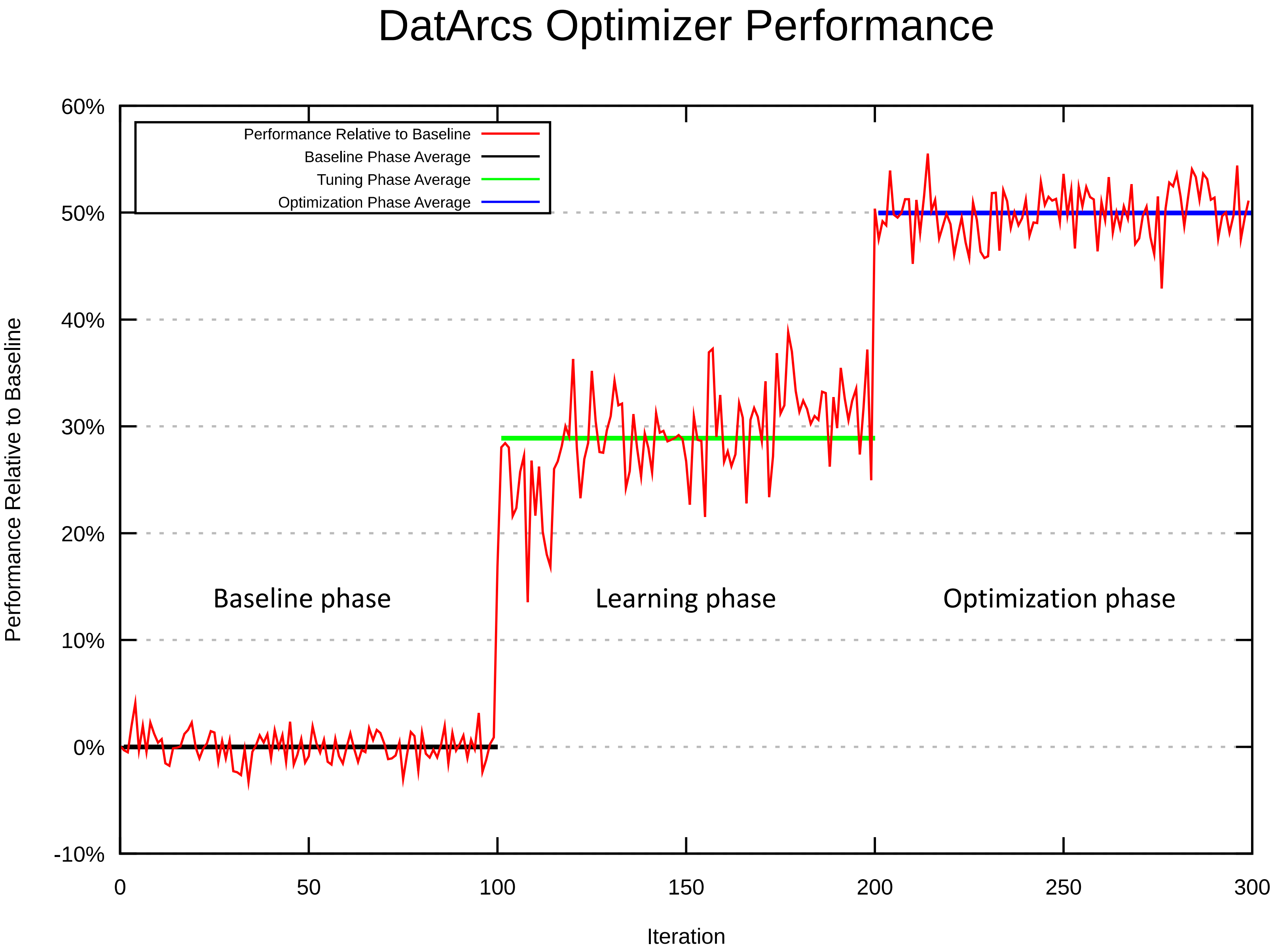
```
# ./demo_mellanox 2
MIGRATED TCP STREAM TEST from 0.0.0.0
(0.0.0.0) port 0 AF_INET to 147.75.108.33
() port 0 AF_INET : demo
Interim result: 1000.47 10^6bits/s
Interim result:  988.52 10^6bits/s
Interim result: 1012.41 10^6bits/s
Interim result: 1104.88 10^6bits/s
Interim result:  996.59 10^6bits/s
Interim result: 1047.43 10^6bits/s
Interim result: 1815.29 10^6bits/s
Interim result: 1600.28 10^6bits/s
Interim result: 1890.25 10^6bits/s
Interim result: 2052.70 10^6bits/s
Interim result: 2086.82 10^6bits/s
Interim result: 1973.90 10^6bits/s
Interim result: 2038.70 10^6bits/s
Interim result: 2026.54 10^6bits/s
Interim result: 2003.32 10^6bits/s
Interim result: 2062.60 10^6bits/s
Interim result: 2097.10 10^6bits/s
```

+109%



CloudSuite

In-Memory Analytics: Recommendation System based on Spark @ type-2 packet



Benchmark

- Collaborative Filtering using Apache Spark
- Three containers on a single machine:
  - One Master
  - Two Workers

Server Configuration

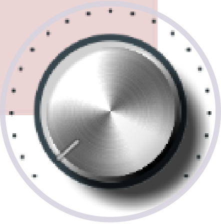
- Packet type-2 server: 2x Intel E5-2650 @ 2.2GHz, 24 cores, 48 threads

DatArcs

Dynamic Tuning Trends

Number of knobs is rising

- Hardware becomes more complex and configurable
- Operating systems also become more complex and configurable
- Heterogeneity: GPUs, Accelerators, FPGAs, etc.



Expertise is becoming scarce

- Wrong to assume that tuning is being taken care of by the “cloud”

Hardware matters!

- More tuning opportunities
- VMs Vs. Bare Metal and Containers

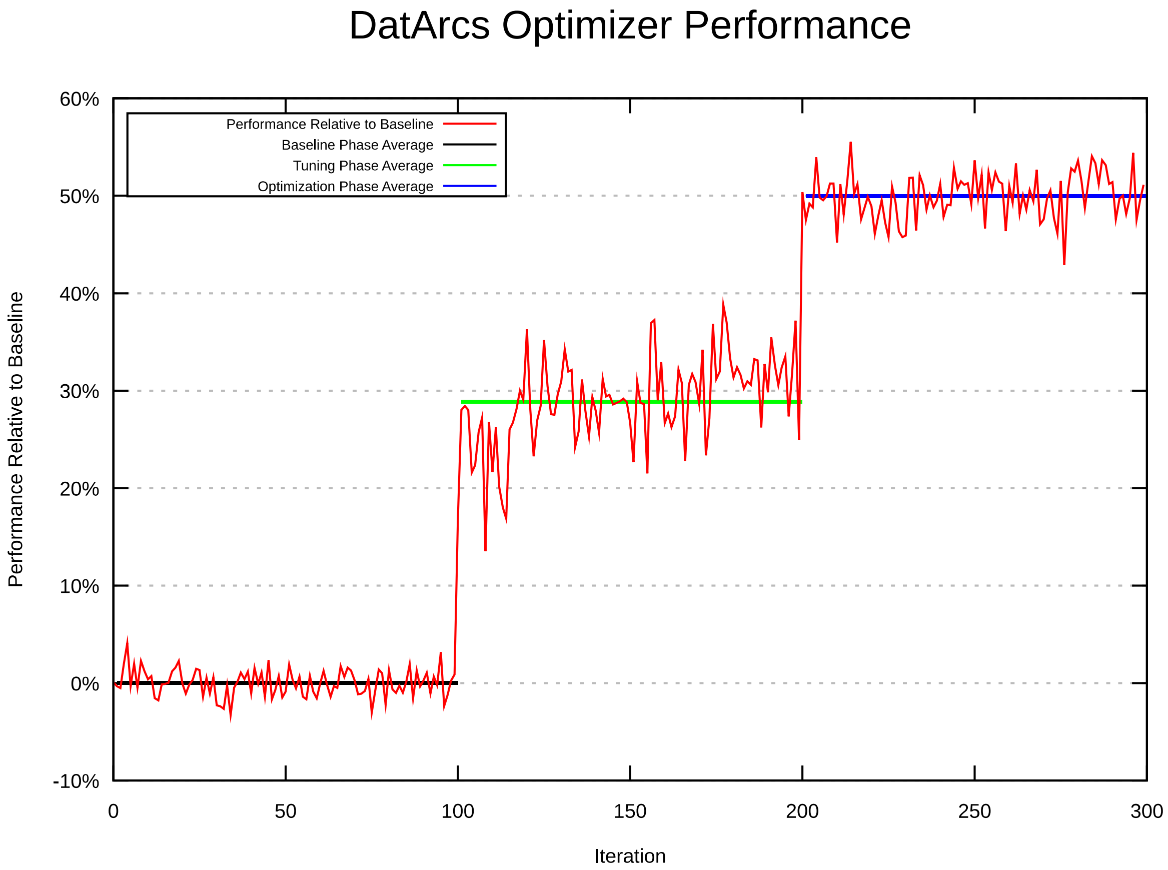
Summary

- ✓ **Server tuning as we know it is about to change!** Dynamic tuning is a quick and simple way to improve performance and energy efficiency of existing systems

- ✓ DatArcs Optimizer is the first feedback-based dynamic tuning software

- ✓ Significant improvement in benchmarks (~50%), and we've only just begun

- ✓ Version 0.5 now in closed beta – feedback welcomed!



DatArcs