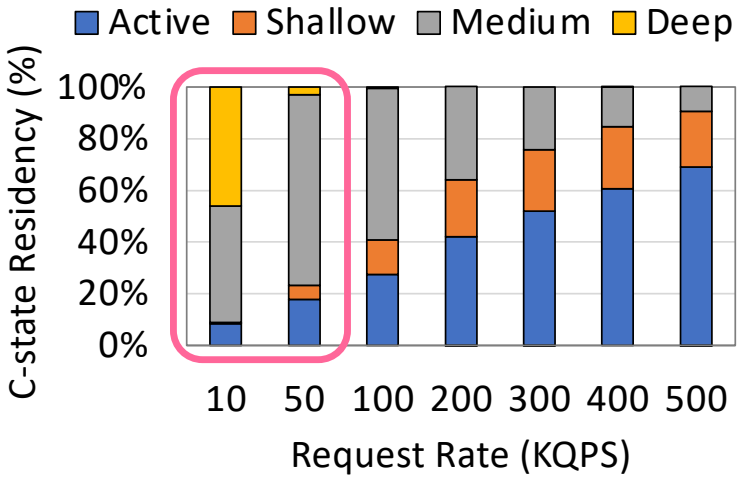
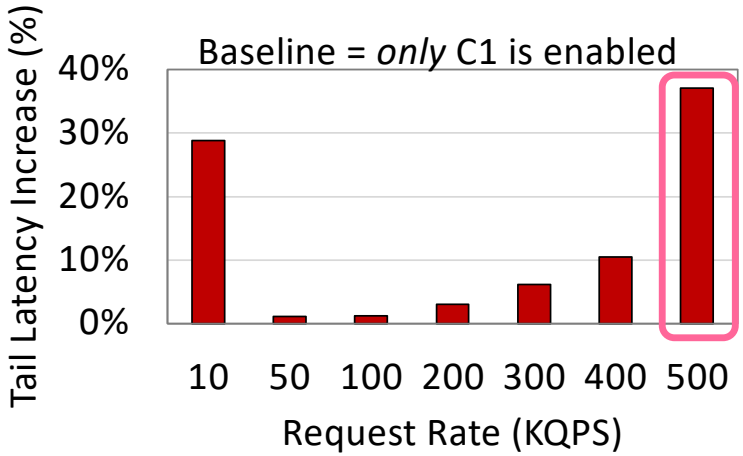
AgileWatts

Motivation

- The load of user-facing applications is unpredictable and bursty
  - with microsecond-scale idleness – as known as the *"Killer microseconds" idleness*
- This load behavior prevents CPU cores from entering deep idle states during idle periods, limiting power savings when the CPU core is idle
  - A CPU core running Memcached never enters Deep idle state when running at ≥20% load
- Idle states transition times can increase tail latency significantly
  - The tail latency of Memcached increases by up to 37% when enabling Medium and Deep idle states





Server vendors disable C-state deeper than CC1 (I)

- Server vendors (e.g., Cisco, Dell, Lenovo) recommend disabling the deep idle C-state (Global C-State Control in BIOS) in AMD EPYC Rome-/Milan-based servers to reduce performance impact

- When disabling the Global C-State Control, the CPU cores will operate only at C0 (active) and C1 (clock-gating)

Table 27. BIOS recommendations for computation-intensive, I/O-intensive, energy-efficient, and low-latency workloads

BIOS options	BIOS values (platform default)	Computation intensive	I/O intensive	Energy efficient	Low latency
Memory					
NUMA Nodes per Socket	Auto (NPS1)	NPS4	NPS4	NPS1	Auto
IOMMU	Auto (Enabled)	Auto*	Auto	Auto	Disabled*
Power/Performance					
SMT Mode	Auto (Enabled)	Auto	Auto	Auto	Disabled
Core Performance Boost	Auto (Enabled)	Auto	Auto	Auto	Disabled
Global C-State Control	Auto (Enabled)	Auto	Auto	Auto	Disabled

Global C-state control

C-states are a processor's CPU core inactive power states. C0 is the operational state in which instructions are processed, and higher numbered C-states (C1, C2, etc.) are low-power states in which the core is idle. The Global C-state setting can be used to enable and disable C-states on the server. By default, the Global C-state control is set to Auto, which enables cores to enter lower power states and can cause jitter due to frequency transitions of the processor cores. When this setting is disabled, the CPU cores will operate at the C0 and C1 states. Table 21 summarizes the settings.

Source: Performance Tuning for Cisco UCS C225 M6 and C245 M6 Rack Servers with 3rd Gen **AMD EPYC** Processors White Paper. Cisco Systems, Inc 2022.
https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/performance-tuning-wp.html

Server vendors disable C-state deeper than CC1 (II)

- Server vendors recommend disabling C-states deeper than C1 in Intel-based servers' processors to maximize performance
- Server vendors recommend even to disable C1E (like C1 with lower voltage/frequency) state
- Disabling C1E reduces the opportunity to enter Turbo since the processor cannot gain enough thermal capacitance by entering C1 state only
  - But on the other hand, C1E is not recommended for latency sensitive workloads
- So, in these systems C1E is disabled and therefore Turbo is effectively disabled
- Note the recommendation to disable L1/L0p of UPI (now CXL) - which effectively disables package C-states

Table 1 UEFI Settings for operating modes on ThinkSystem servers with Intel processors.

Menu Item	Page	Category	Minimal Power	Efficiency – Favor Power	Efficiency – Favor Performance	Maximum Performance
Operating Mode	9	Suggested	Minimal power	Efficiency – Favor Power	Efficiency – Favor Performance	Maximum Performance
CPU P-State Control	10	Recommended	Autonomous	Autonomous	Autonomous	None
C-States	11	Recommended	Autonomous	Autonomous	Autonomous	Disabled
MONITOR/ MWAIT	12	Suggested	Enabled	Enabled	Enabled	Disabled
C1E Enhanced Mode	12	Recommended	Enabled	Enabled	Enabled	Disabled
UPI Link Frequency	13	Recommended	Minimal power	Minimal Power	Max Performance	Max performance
UPI Link Disable	13	Recommended	Minimum Number of Links Enabled	Minimum Number of Links Enabled	Enabled All Links	Enabled all links
UPI Power Management	14	Recommended	L1: Enabled L0p: Enabled	L1: Enabled L0p: Enabled	L1: Enabled L0p: Enabled	L1: Disabled L0p: Disabled
Turbo Mode	14	Suggested	Disabled	Disabled	Enabled	Enabled

C1E can help to provide power savings in those circumstances where cache coherency is paramount. Those applications which thread well and can maintain utilization of processor cores (virtualization, HPC and database workloads) do not benefit and under certain circumstances may be hindered by C1E. If a user is attempting to achieve maximum opportunity for Turbo Mode to engage, C1E is recommended. C1E is not recommended for latency sensitive workloads.

Source: Tuning UEFI Settings for Performance and Energy Efficiency on **Intel Xeon** Scalable Processor-Based ThinkSystem Servers, Lenovo 2021
https://lenovopress.lenovo.com/lp1477.pdf

Server vendors disable C-state deeper than CC1 (III)

Intel Select Solution configuration for servers running Microsoft SQL recommends disabling C-states

Intel® Select Solutions for SQL Server* Enterprise Data Warehouse ensure that a company has validated hardware and software stacks that provide a fast path for taking advantage of big data—both structured and unstructured—massive data volumes, and rapid data analysis. Proven to scale with Intel® Xeon® Scalable processors, these pre-tuned and tested configurations are workload-optimized and let organizations deploy the optimum data warehouse infrastructure quickly and with less tuning.

FIRMWARE AND SOFTWARE OPTIMIZATIONS	Intel® Hyper-Threading Technology (Intel® HT Technology) enabled Intel® Turbo Boost Technology enabled Intel® Speed Shift technology, Hardware P-states (HWP) native Intel NVMe drivers C-states disabled Operating system power management and plan set for performance

Source: Intel® Select Solutions for Microsoft SQL Server https://www.intel.com/content/www/us/en/analytics/select-solutions-for-data-warehouse-brief.html

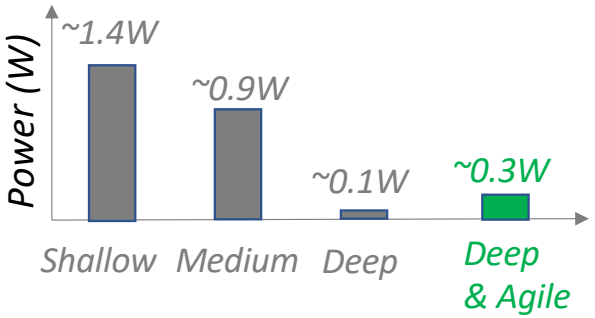
Despite having the capabilities to place individual idle cores in a deep low-power state

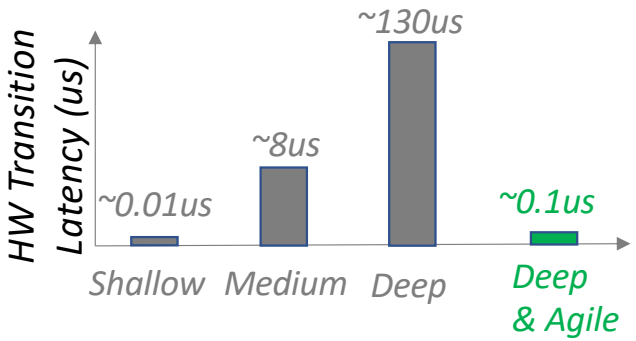
these capabilities are typically disabled in modern servers running latency-critical applications to reduce their performance impact,

which significantly increases the energy consumption of these servers

Our Goal

- To eliminate the *killer microseconds effect* that prevent servers running latency critical application from entering deep energy saving states, we propose AgileWatts:
  - A Deep & Agile low power state with
  - Nanosecond-scale transition latency





- We design AgileWatts with two design goals in mind:
  1. Drastically reducing the transition latency of deep core idle power states, making deep C-states usable
  2. Retaining most of the power savings of deep idle states

Outline

- Motivation and Goal
- **AgileWatts**
  - **Units Fast Power-Gating**
  - **Caches Coherency and Sleep Mode**
  - **C6A Power Management Flow**
- Evaluation

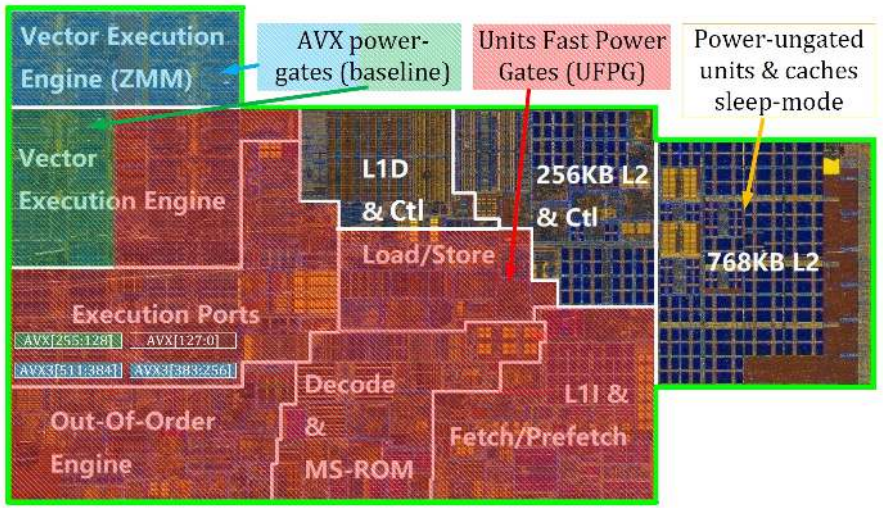
AgileWatts

We achieve AgileWatts goals with three key components:

1. Units Fast Power-Gating
2. Cache Coherence and Sleep Mode
3. C6A Power Management Flow

1. Units Fast Power-Gating

- Units Fast Power-Gating is a low-latency power-gating (PG) architecture that
  - Shuts off most of the core units while retaining the context in place
  - thus, enabling a transition latency of tens of nanoseconds



- AgileWatts retains the CPU context in place, completely removing save/restore latency overhead at a very small additional idle power cost

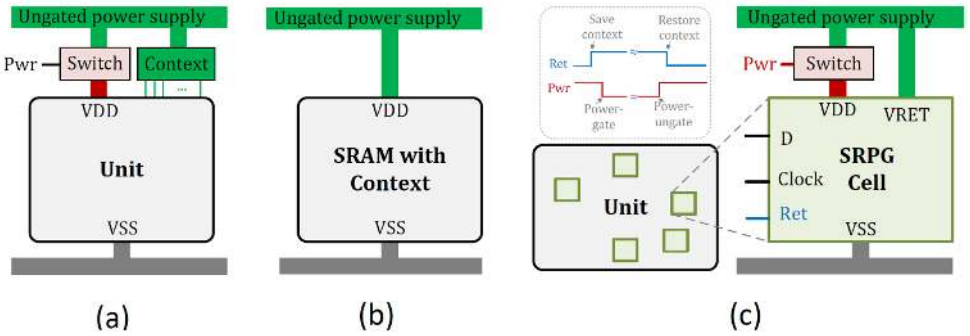
Retaining the Context in Place

- AgileWatts leverages three techniques to efficiently retain the context in C6A

a) Placing Unit Context
in the Ungated Domain

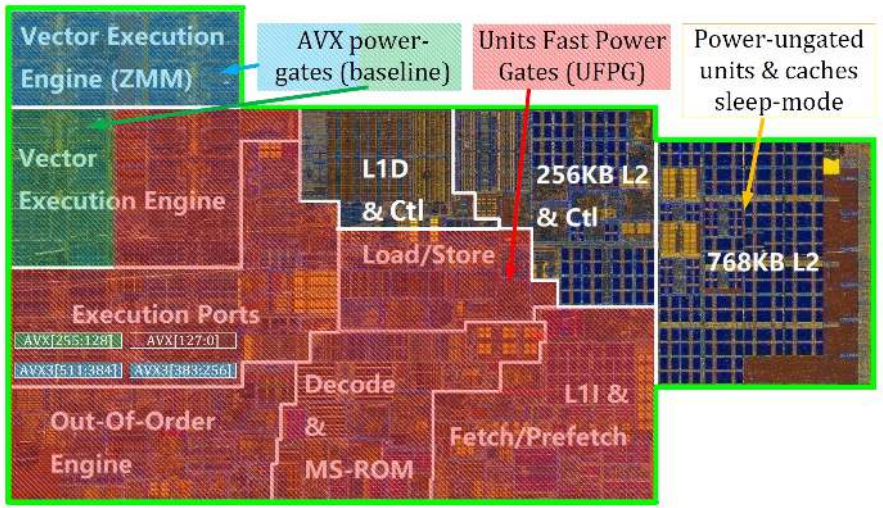
b) Place SRAM Context
in the Ungated Power Supply

c) State Retention Power Gates (SRPGs)



2. Caches Coherency and Sleep Mode (I)

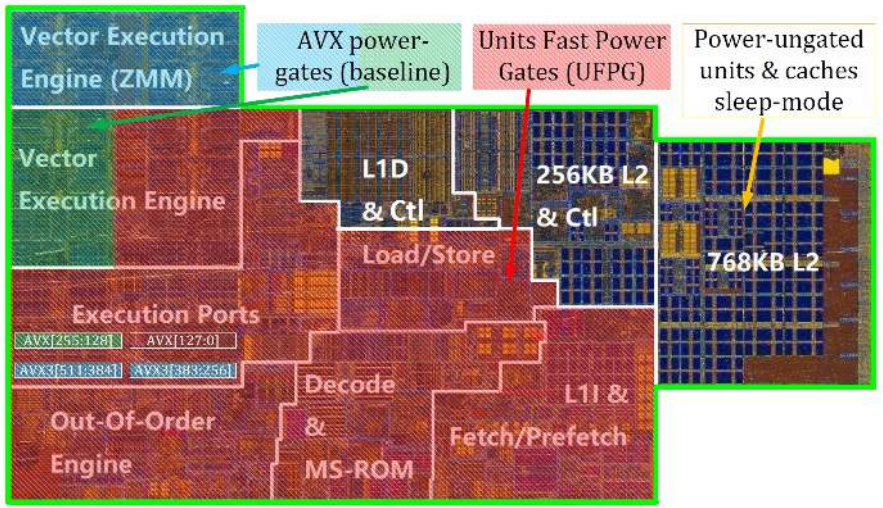
- To avoid the high latency to flush private caches to power-gate them, AgileWatts keeps them power-ungated when transitioning to C6A



- This has two design implications:
  - AgileWatts needs to employ other power-saving techniques to reduce the power of the caches
  - A core in C6A state still needs to serve coherence requests (i.e., snoops)

2. Caches Coherency and Sleep Mode (II)

- AgileWatts employs two key techniques to reduce the power consumption of the power-ungated private cache domain:

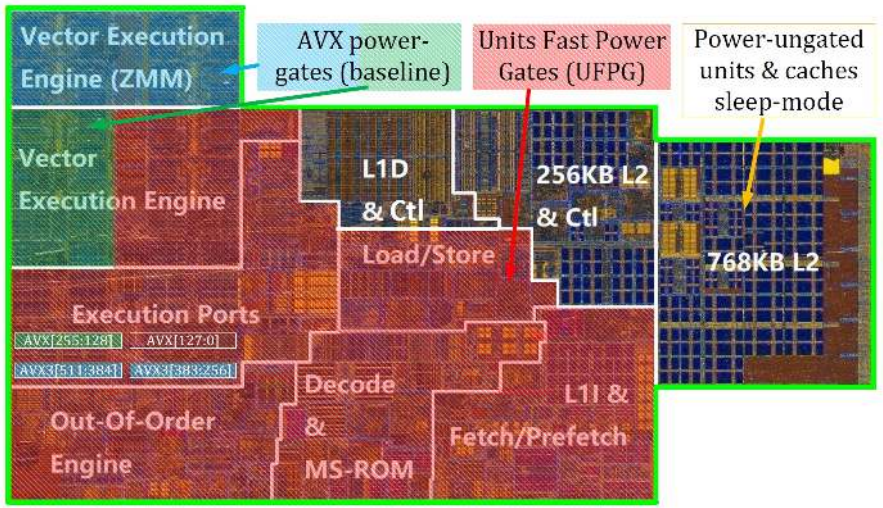


1. Unless a coherency request is being served, AgileWatts keeps this domain clock-gated to save its dynamic power

2. AgileWatts leverages the cache sleep-mode technique, which adds sleep transistors to the SRAM array of private caches
   - These sleep transistors reduce the SRAM array's supply voltage to the lowest level that can safely retain the SRAM content while significantly reducing leakage power

2. Caches Coherency and Sleep Mode (III)

- Since private caches are not flushed when a core enters C6A, AgileWatts must allow the core to respond to snoop requests

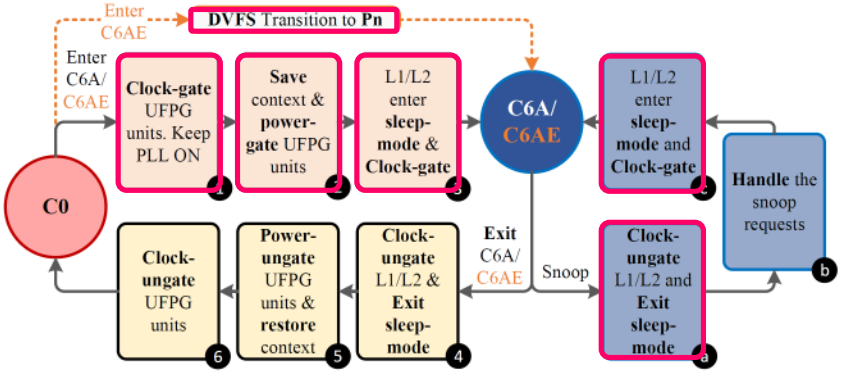


- AgileWatts keeps the logic required to handle cache snoops in the power-ungated domain together with the private caches

- As soon as this logic detects incoming snoop traffic
  - it temporarily increases the SRAM array voltage through the sleep transistors and reactivates the clock of the private caches for the time required to respond to snoop requests

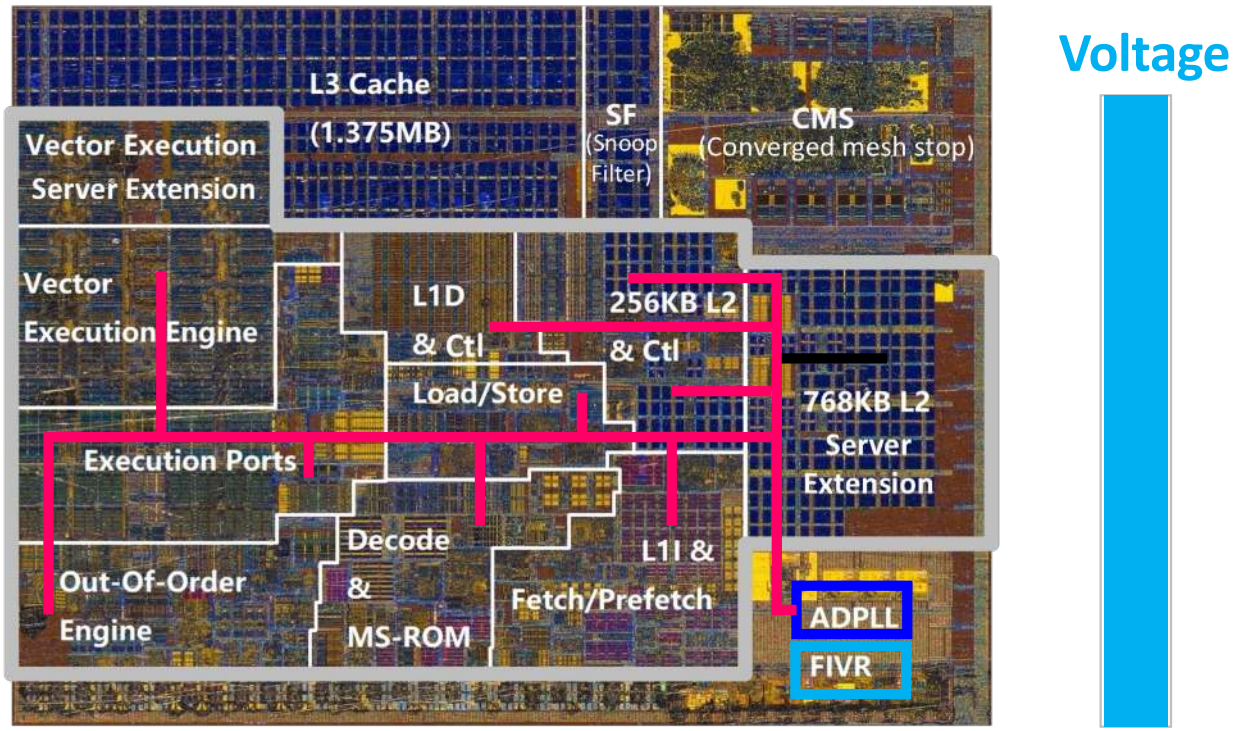
3. C6A Power Management Flow

- AgileWatts flow orchestrates the transitioning between the C0 and C6A C-states and handles coherence traffic while in C6A state
- The entry flow:
  - Clock-gates the gated-domain (UFPG) and keeps the core PLL (phase-locked loop) powered-on
    - When entering C6AE (C6A Enhanced), it initiates a non-blocking transition to Pn – the P-state with lowest frequency and voltage
  - Saves (in place) the gated-domain's context and shuts down its power
  - Sets the private caches into sleep mode and shuts down their clock



- When a snoop request arrives, the flow:
  - Clock-ungates the private cache domain and adjusts its supply voltage to exit sleep mode
  - When all outstanding snoop requests are serviced, the flow rolls back the changes in reverse order and brings the core back into full C6A (or C6AE) state

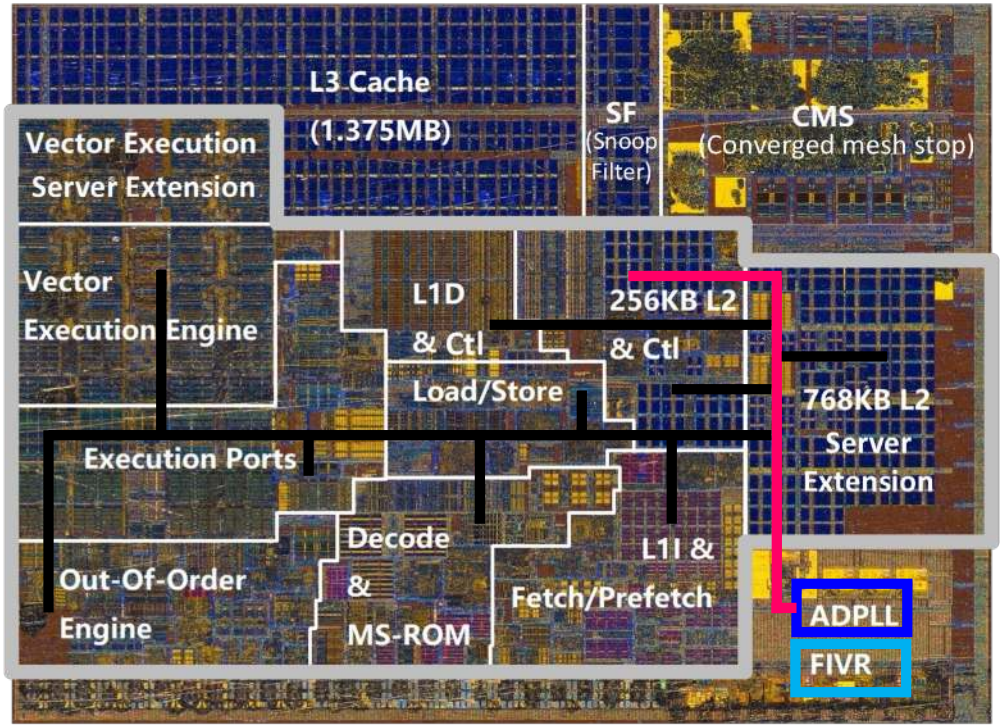
C6A (Deep & Agile) Core C-state



C-State	Clocks	ADPLL	L1/L2 Cache	Voltage	Context
C6A	Running	On	Coherent	Nominal	Maintained

C6A (Deep & Agile) Core C-state

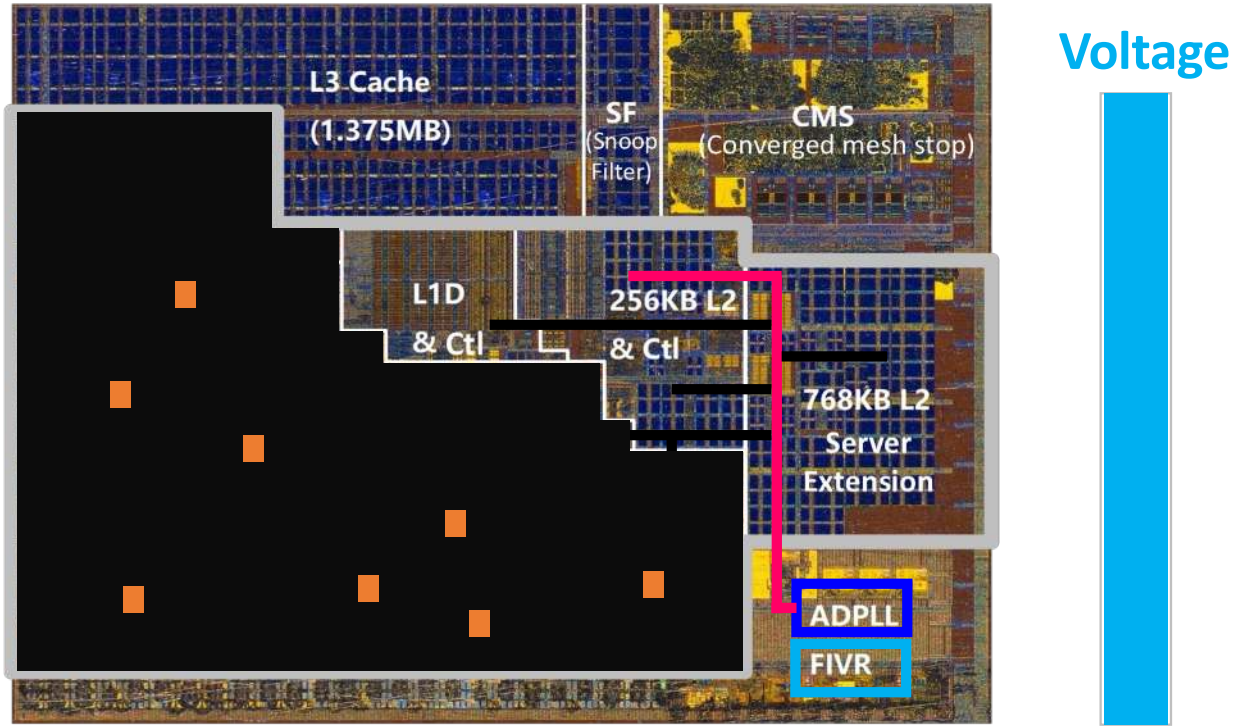
- **Clock-gate** most of the clock and keep the PLL On



C-State	Clocks	ADPLL	L1/L2 Cache	Voltage	Context
C6A	Most Stopped	On	Coherent	Nominal	Maintained

C6A (Deep & Agile) Core C-state

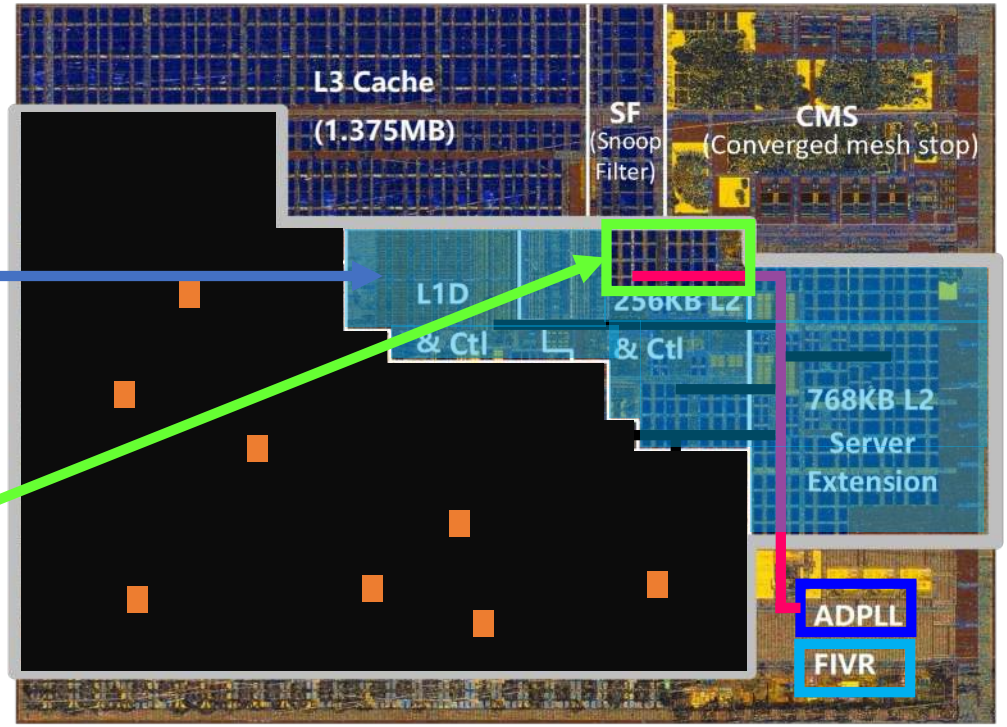
- **Power-gate** most of the core units
- Retaining the **context in place**



C-State	Clocks	ADPLL	L1/L2 Cache	Voltage	Context
C6A	Most Stopped	On	Coherent	**Most PG**	**Maintained**

C6A (Deep & Agile) Core C-state

- Reduce the caches supply voltage using **sleep transistors**
- Retain **a logic active** to wake-up the caches to respond to snoop requests



Voltage

C-State	Clocks	ADPLL	L1/L2 Cache	Voltage	Context
C6A	Most Stopped	On	Coherent	**PG/Ret/Nom**	**Maintained**

Outline

- Motivation and Goal
- AgileWatts
- **Evaluation**

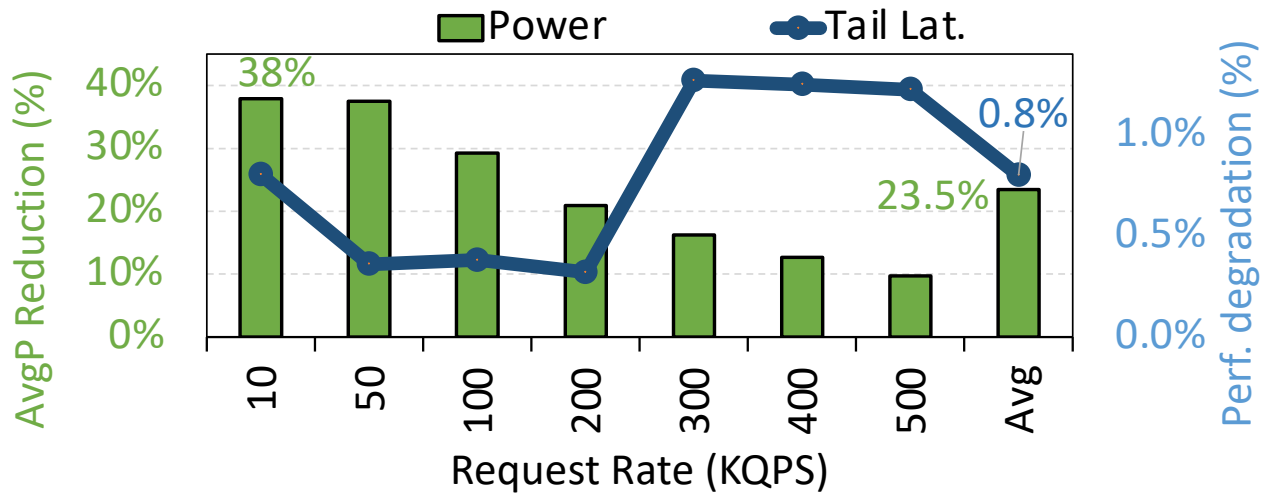
Evaluation Methodology

- **Power and Performance Model:** to evaluate AgileWatts, we use an accurate analytical power model, calibrated against an Intel Skylake server and considering the performance penalty

- **Power and C-state Residency Measurements:** We measure C-state residency and number of transitions using processor's residency reporting counters
  - We use the RAPL interface to measure power consumption

- **Workloads:** We evaluate AgileWatts using three latency-critical workloads: Memcached, Apache Kafka, and MySQL

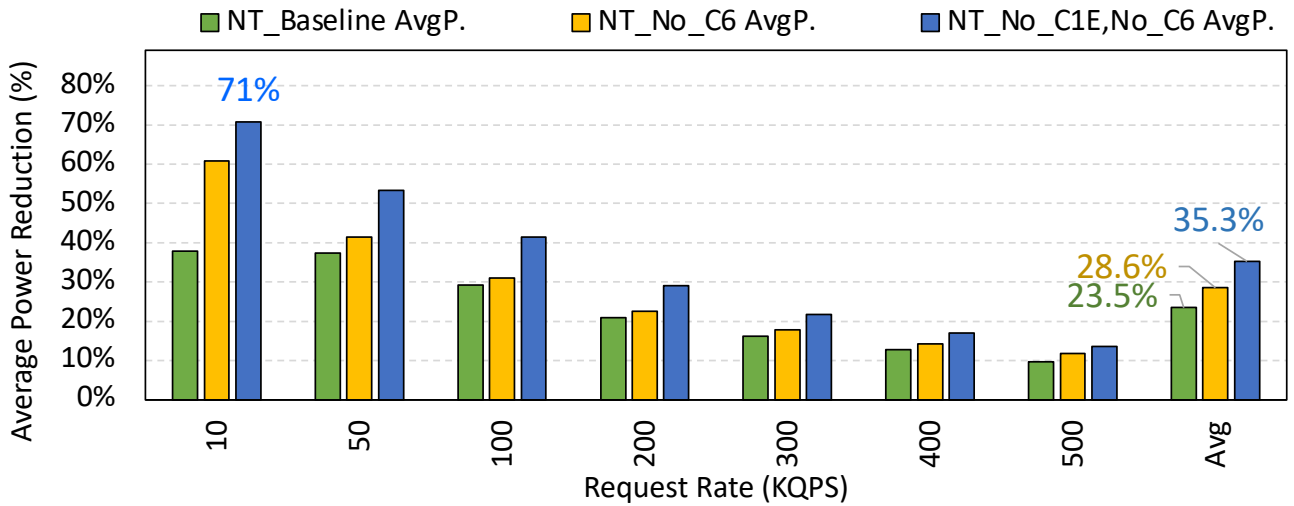
Power Savings and Overhead at Varying Load Levels



- AgileWatts reduces the average power consumption by up to 38% at low load
  - At high load, AgileWatts still provides 10% power savings

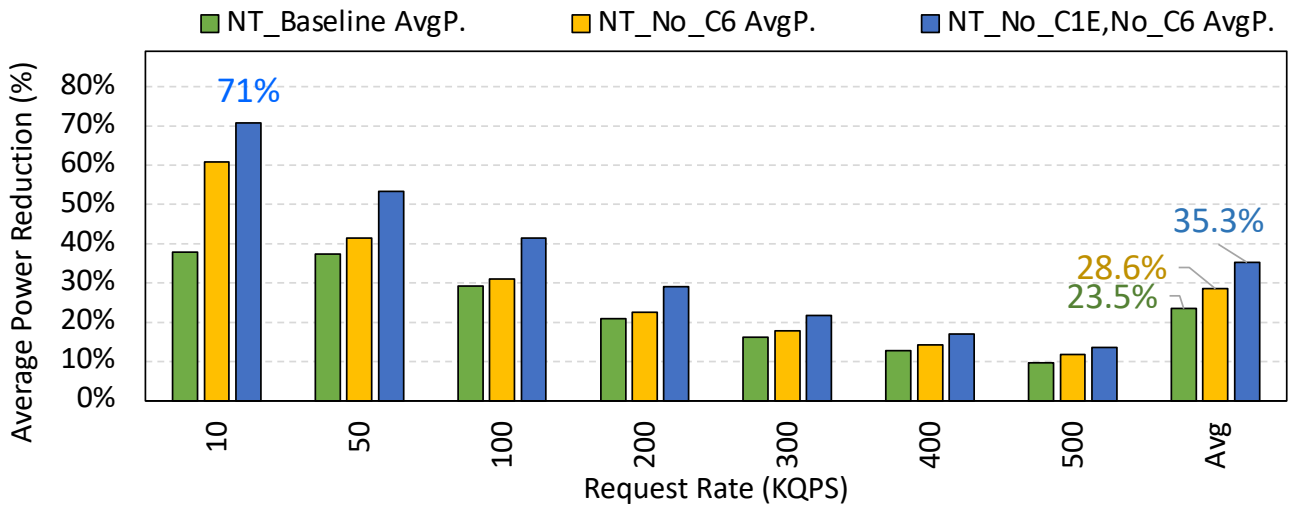
- AgileWatts has less than 1.3% impact on tail latency (server side)

Commonly used Configurations



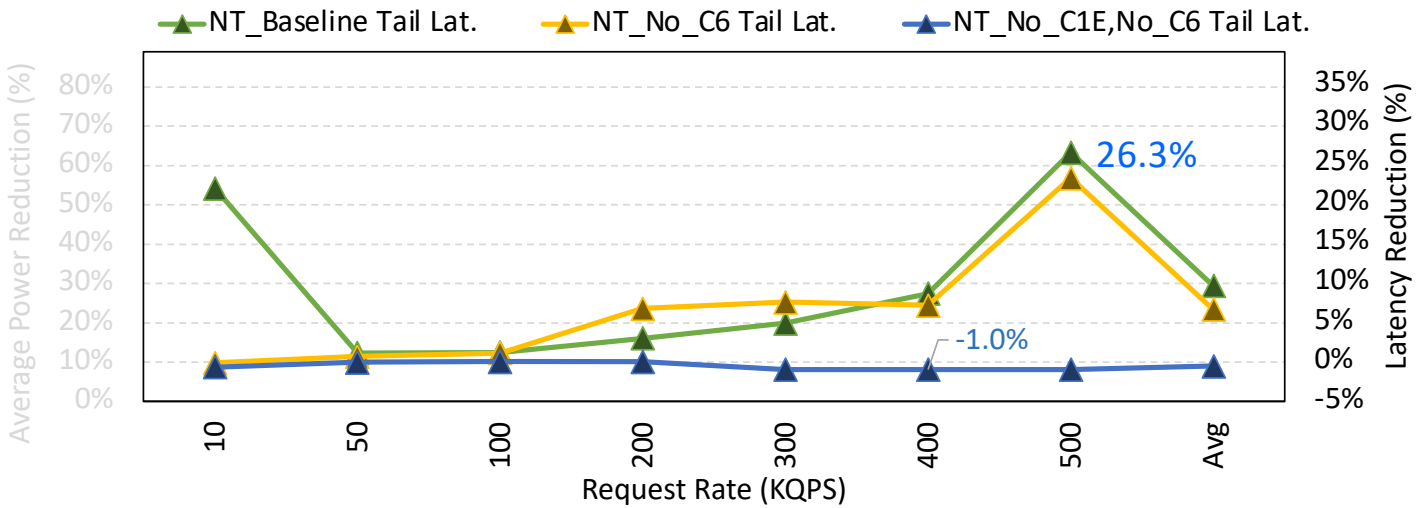
- Server vendors provide recommended system configurations, such as disabling certain C-states to increase system performance or disabling Turbo Boost to reduce power consumption
- We analyze three common configurations that *successively* disable Turbo, C6, and C1E in the baseline configuration (P-states disabled and Turbo and C-states enabled)
  1. Turbo disabled (NT_Baseline),
  2. Turbo and C6 disabled (NT_No_C6), and
  3. Turbo, C6 and C1E disabled (NT_No_C6,No_C1E)

Commonly used Configurations



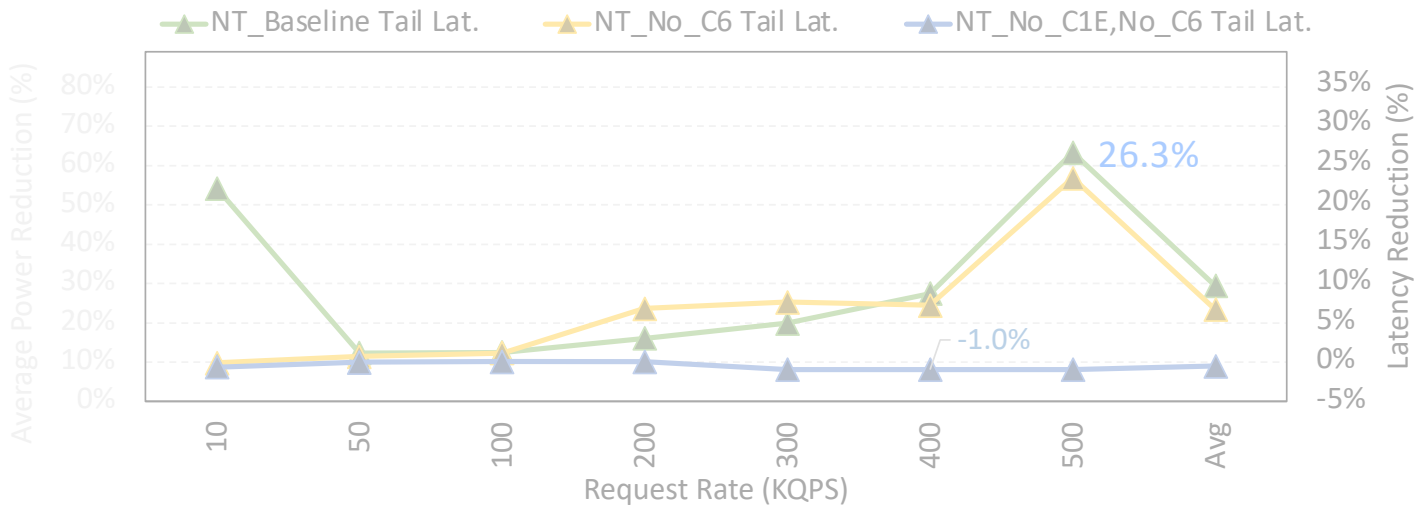
- AgileWatts reduces average power consumption (AvgP) up to 71% against all three tuned configurations
- The reason is that, in these workloads AgileWatts replaces the time that other configurations spend in C1 with C6A C-state, which has much lower power

Commonly used Configurations



- AgileWatts improves the performance by up to 26.3% compared to NT_Baseline and NT_No_C6 configurations
- AgileWatts only degrades the performance by up to 1% compared to NT_No_C6,No_C1E configuration

Commonly used Configurations



AgileWatts provides average and tail latencies comparable to the tuned configurations that disable some power management features, while reducing power significantly

Other Details in the Paper

- Implementation and hardware cost
- Idle power analysis
- C6A and C6AE latency
- Performance penalty
- Staggered unit wake-up
- Design complexity and effort
- Power savings and overhead at varying load levels
- Analysis of turbo boost performance improvement

Summary

- AgileWatts introduces a new deep C-state architecture that drastically reduces the transition latency, making deep C-states usable

- AgileWatts architecture can be achieved by leveraging agile power management techniques while retaining most of the power savings of existing deep idle states

- AgileWatts reduces energy consumption of servers running latency-critical applications by enabling the server's processor to enter deep energy-saving states during short idle periods with negligible performance impact
  - AgileWatts reduces the energy consumption of Memcached by up to 71% with up to 1% performance degradation