

Project Defence: Energy Consumption Analysis of CPU Tasks in Operating Systems

By: Nicholas Fantino-Dyer

Output Generation

- Data collected using:
 - Linux tools: stress-ng, Intel PCM, and sensors.
 - Windows tools: Heavy Load, Intel PCM.
- Scripts automated the process of data collection.
- Temperature and energy metrics captured and merged for synchronized analysis.
- Some of the parameters measured were CPU utilization, energy per instruction, cache misses per joule, as well as thread utilization comparison.
- Data was analyzed using pandas, matplotlib, numpy, and sklearn.

Sample Output of PCM Data

[illegible]

Zoomed in Example With less columns:

Date,Time,EXEC,IPC,FREQ,AFREQ,CFREQ,L3MISS,L2MISS,L3HIT,L2HIT,L3MPI,L2MPI,READ,WRITE,INST,ACYC,TIME(ticks),PhysIPC,PhysIPC%,INSTnom,INSTnom%,C0res%,C1res%,C3res%,C6res%,C7res%,C0res%,C2res%,C3res%,C6res%,C7res%,C8res%,C9res%,C10res%,Proc Energy (Joules),Power Plane 0 Energy (Joules),Power Plane 1 Energy (Joules)

2024-12-03,16:10:46.522,0.00,0.46,0.00,1.03,4.32,0.01,0.16,0.90,0.48,0.0004,0.0050,0.01,0.01,32.66,70.37,4202.65,0.93,23.21,0.00,0.05,0.20,1.72,0.10,1.08,96.90,100.00,0.00,0.00,0.00,0.00,0.00,0.00,12.41,0.00,0.00,0.00,0.00,0.46,0.00,1.03,4.32,0.01,0.16,0.90,0.48,0.0004,0.0050,0.01,0.01,0.00,0.01,0.00,54.32,66.70,37.42,02.65,0.93,23.21,0.00,0.05,0.20,1.72,0.10,1.08,96.90,100.00,0.00,0.00,0.00,0.00,0.00,0.00,12.41,1.23,0.00,0.00,0.44,0.00,1.06,4.43,0.00,0.02,0.90,0.41,0.0006,0.0060,0.20,0.96,0.05,0.14,98.65,54.39,0.87,4202.65,0.89,22.22,0.00,0.05,0.00,0.45,0.00,1.05,4.42,0.00,0.01,0.95,0.49,0.0002,0.0051,0.09,1.15,0.00,1.20,97.57,55.17,4.38,4202.65,0.91,22.69,0.00,0.02,0.00,0.12,0.00,1.04,4.35,0.00,0.01,0.73,0.33,0.0031,0.0119,0.08,3.95,0.33,2.97,92.66,56.04,3.53,4202.65,0.24,6.09,0.00,0.01,0.00,0.18,0.00,0.94,3.95,0.00,0.01,0.88,0.30,0.0014,0.0128,0.07,1.21,0.00,0.00,98.73,55.04,6.25,4202.65,0.36,8.92,0.00,0.01,0.00,0.65,0.00,1.03,4.34,0.00,0.02,0.90,0.52,0.0003,0.0034,0.18,0.98,0.05,0.14,98.65,54.49,6.76,4202.65,1.29,32.36,0.00,0.06,0.00,0.61,0.00,1.04,4.38,0.00,0.02,0.81,0.57,0.

Formatting CSV for Analysis

Why Was Correction Necessary?

- Adding Data Types For Influx
- Mismatch in Data Headers
- Line Count Discrepancies
- Formatting Inconsistencies

Solution: Corrected Data Scripts

- `CorrectDataWindows.ps1` and `CorrectDataLinux.ps1`
- Row Alignment Fix
- Column Formatting

Key Challenges in Data Collection

Windows 10 Limitation:

- Limited availability of Windows 10 machines for testing restricted the ability to diversify hardware configurations.
- Tests were performed only on an Intel i7-7700K machine running Windows 10.

Performance Testing:

- Required use of Heavy Load to simulate stress since stress-ng is not available on Windows.
- Inability to directly control load percentages with Heavy Load limited precise comparisons.

Key Challenges in Data Collection Continued

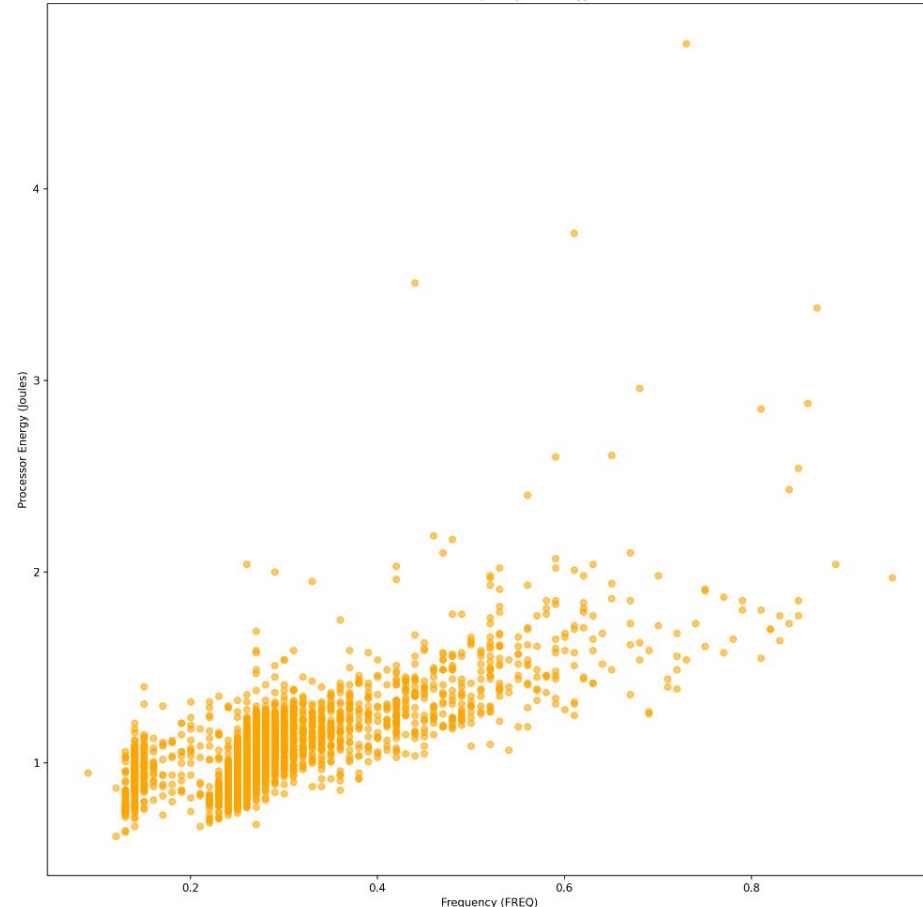
Linux Flexibility:

- Performed tests on two machines:
 - Intel i7-7700K for baseline data.
 - Intel i7-13700K for generational comparison.
- Used stress-ng, which allowed detailed control over load percentages and threading.
- Was able to use temperature sensor data.

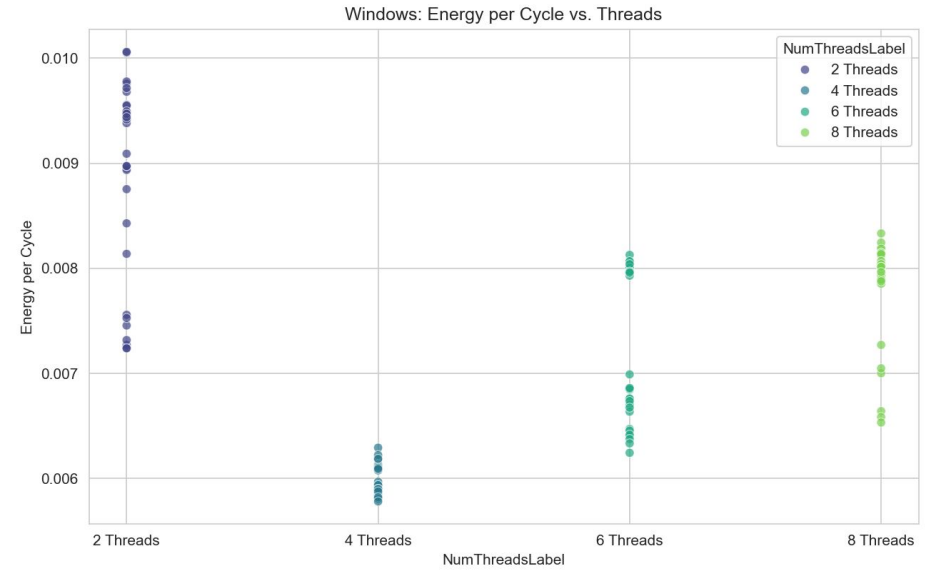
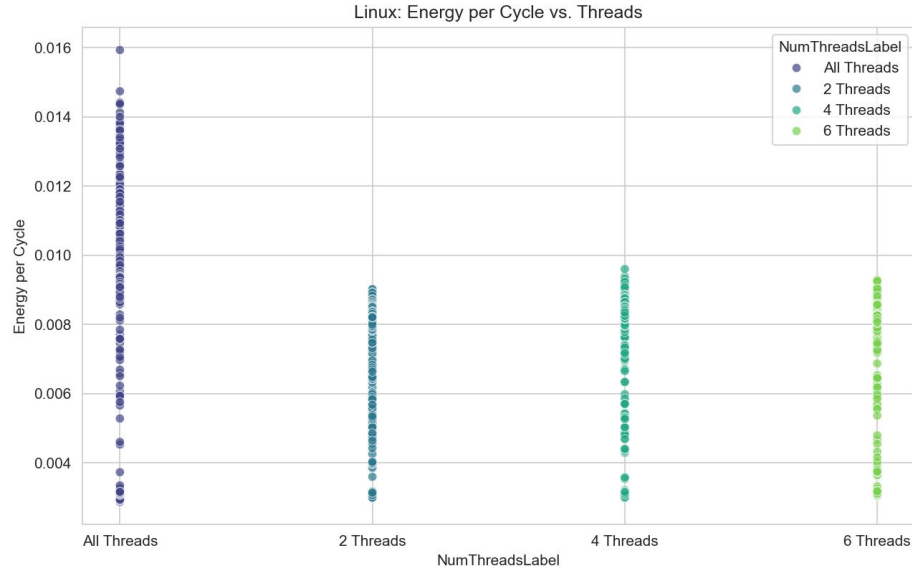
Output Analysis

- Windows:
 - Higher variability in energy consumption, with inefficient energy use at low and high loads.
 - Observed anomalies like dips in energy consumption.
 - Higher cache misses per joule.
- Linux:
 - More consistent energy efficiency across workloads.
 - Peaks at 40% load, followed by gradual declines likely due to potential PCM inaccuracies or throttling.
 - Lower cache misses per joule.

A scatter plot showing the relationship between Frequency (FREQ) on the x-axis and an unlabeled variable on the y-axis. The x-axis ranges from 0.00 to 0.14 with major ticks every 0.02. The y-axis ranges from 2 to 10 with major ticks every 2 units. The data points are blue dots, showing a positive correlation. There is a dense cluster of points at low frequencies (0.00 to 0.03) with y-values between 2 and 4.5. As frequency increases, the y-values generally increase, with a notable jump in y-value around a frequency of 0.10, where points reach up to 8. The highest frequency shown is approximately 0.14, with a y-value of about 10.5.

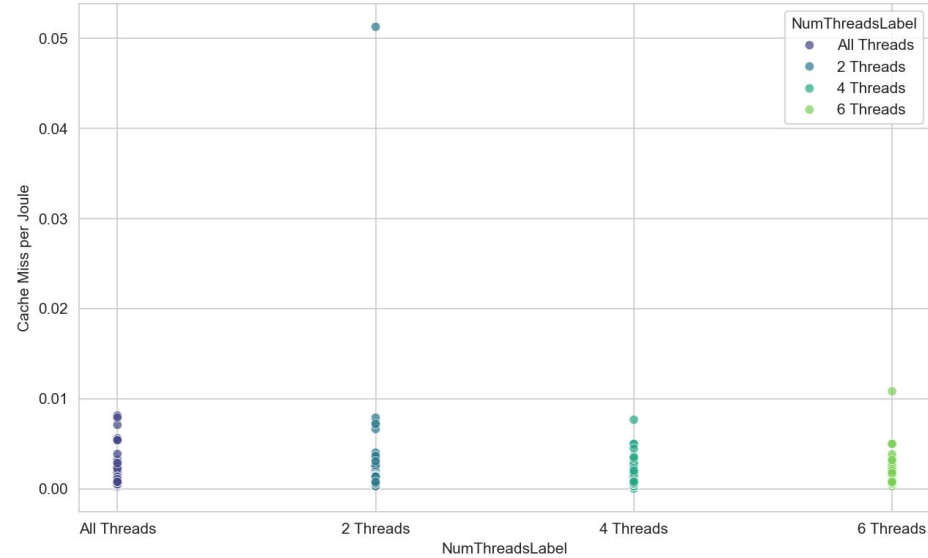


Energy per Cycle vs Threads

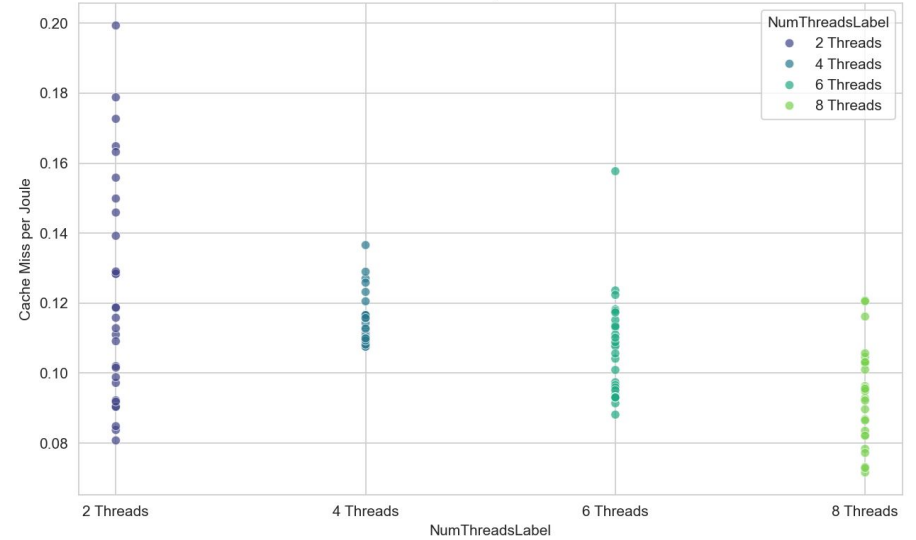


Cache Miss per Joule vs Threads

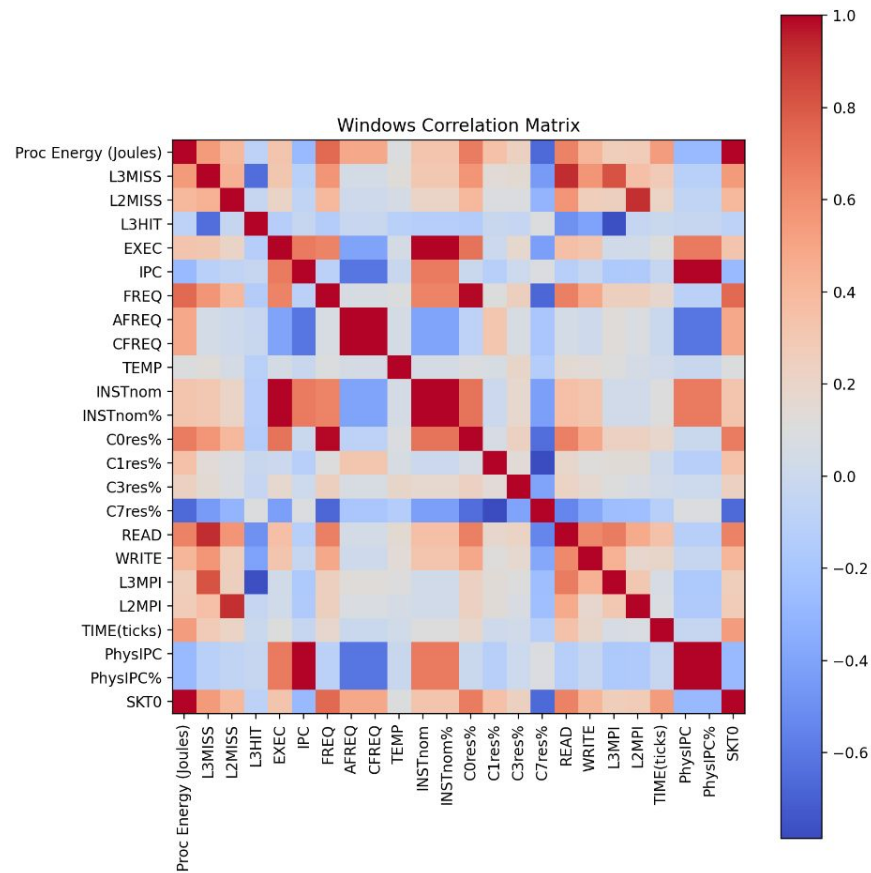
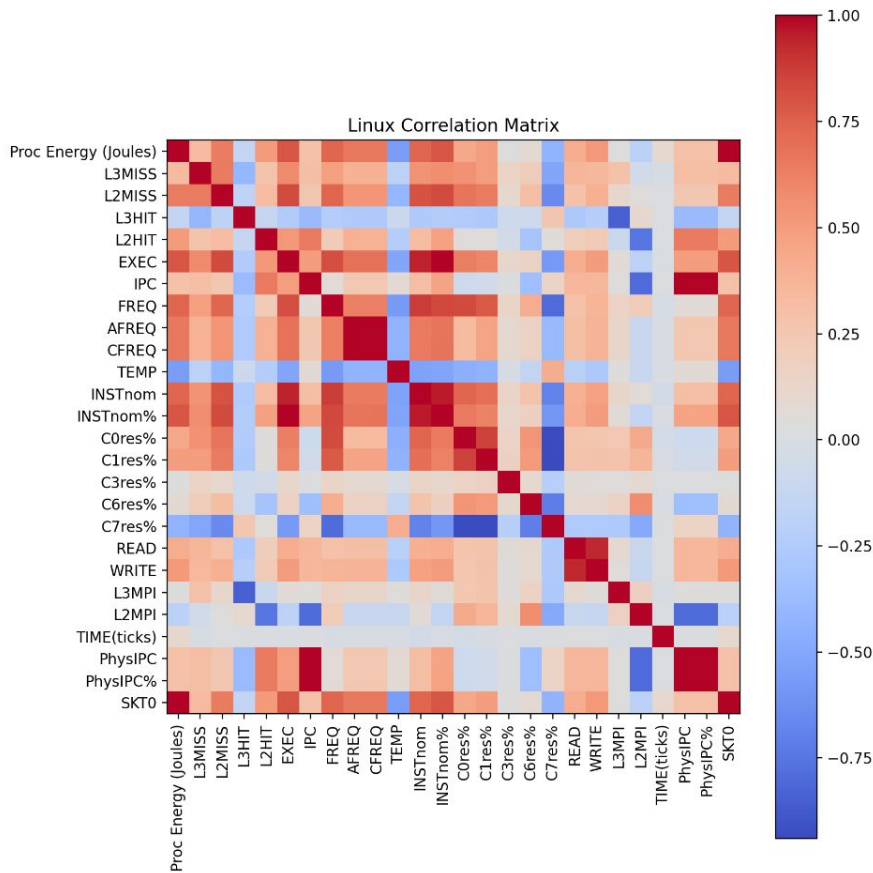
Linux: Cache Miss per Joule vs. Threads



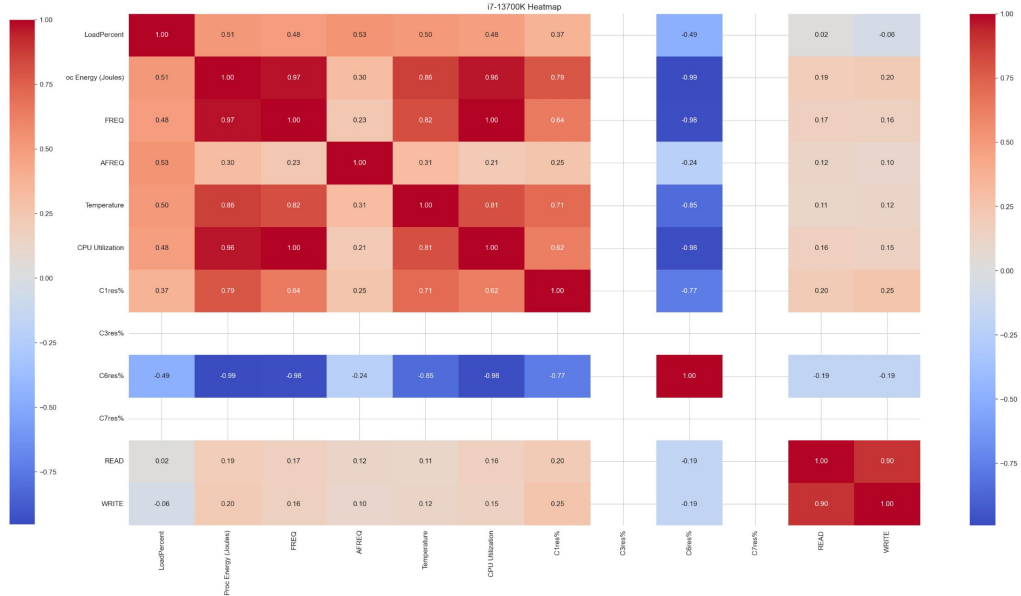
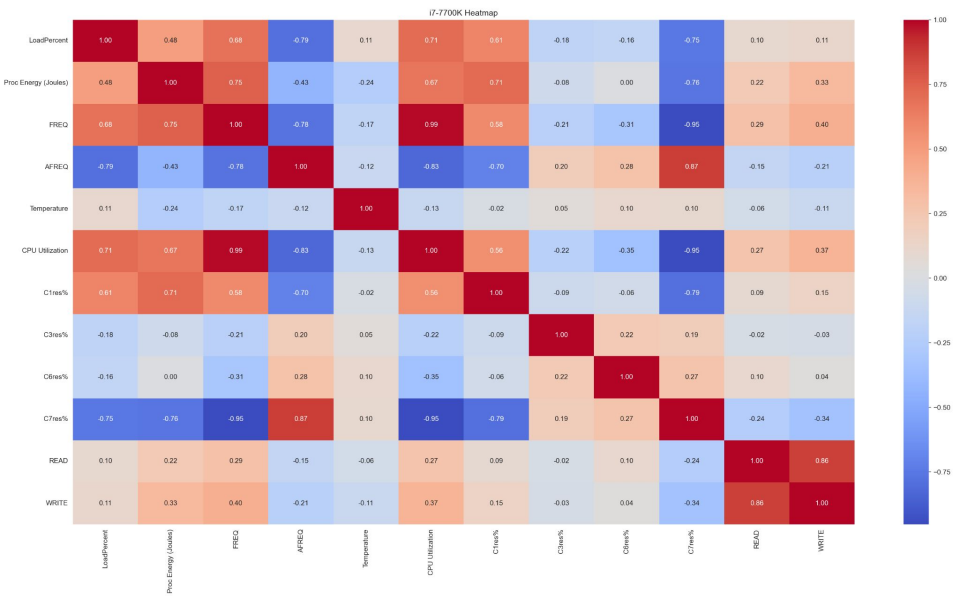
Windows: Cache Miss per Joule vs. Threads



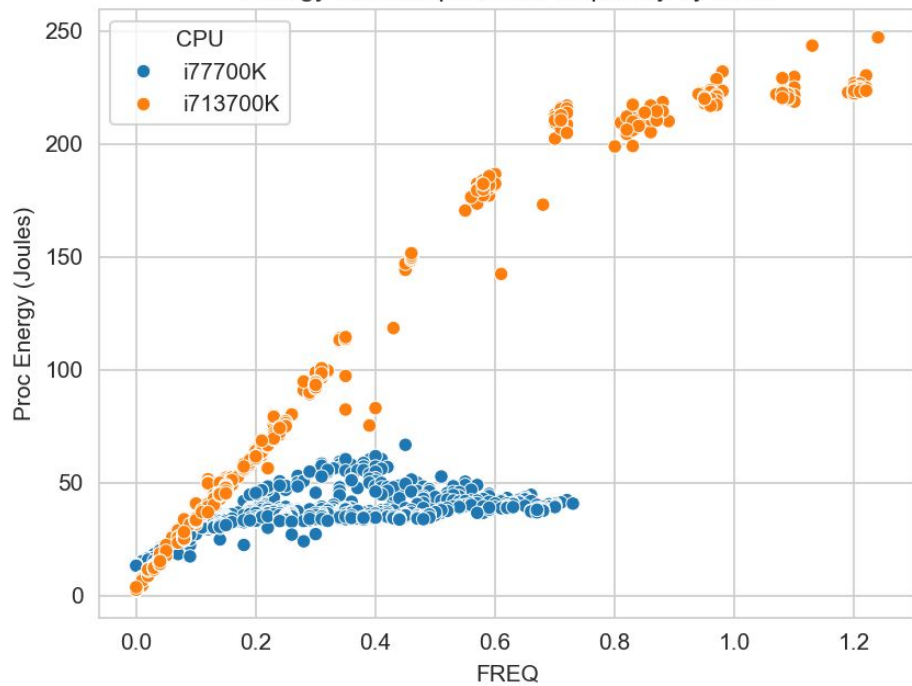
Correlation Matrices



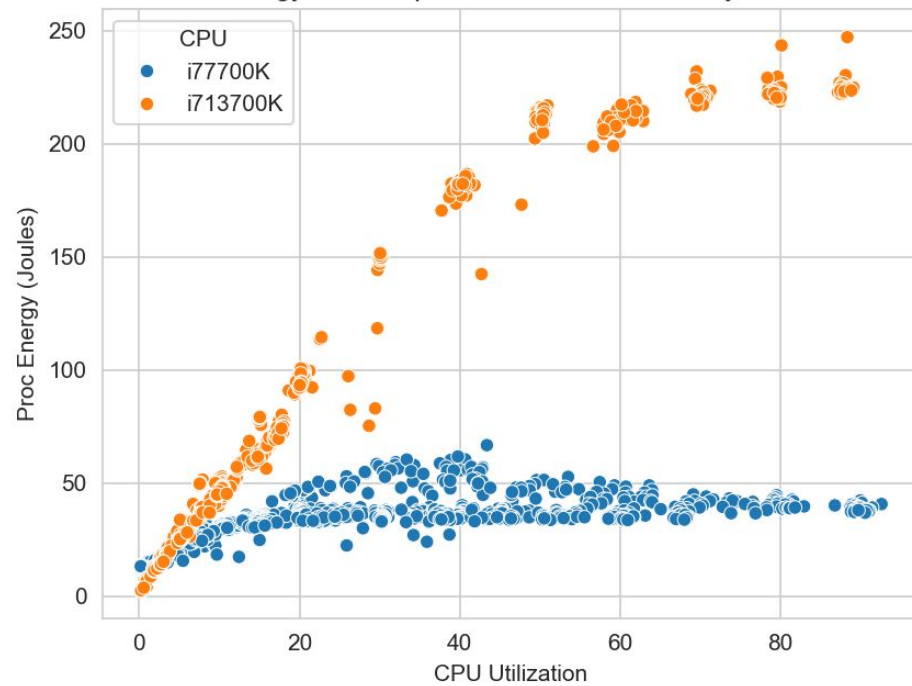
Correlation Matrices



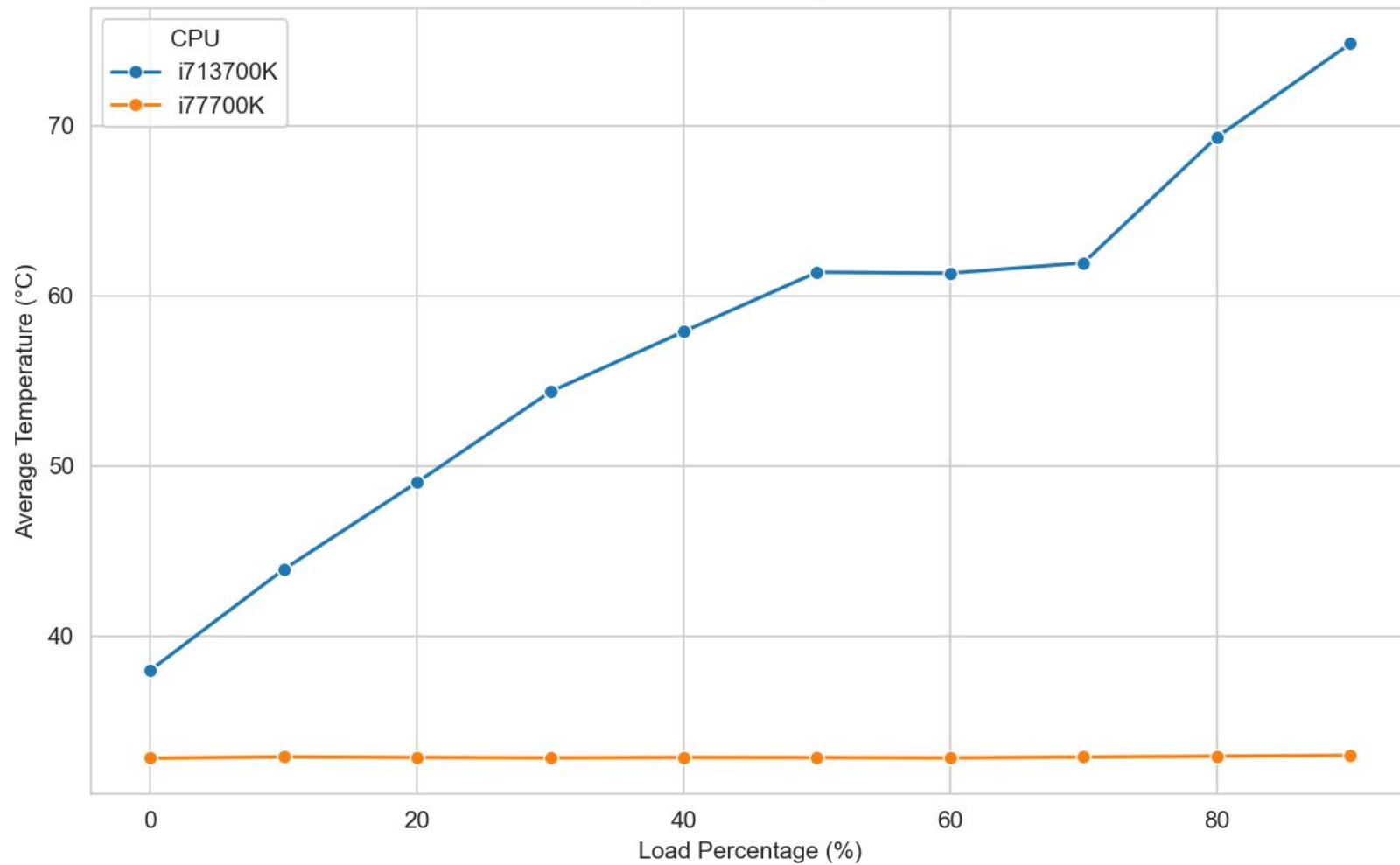
Energy Consumption vs Frequency by CPU



Energy Consumption vs CPU Utilization by CPU



Temperature Comparison



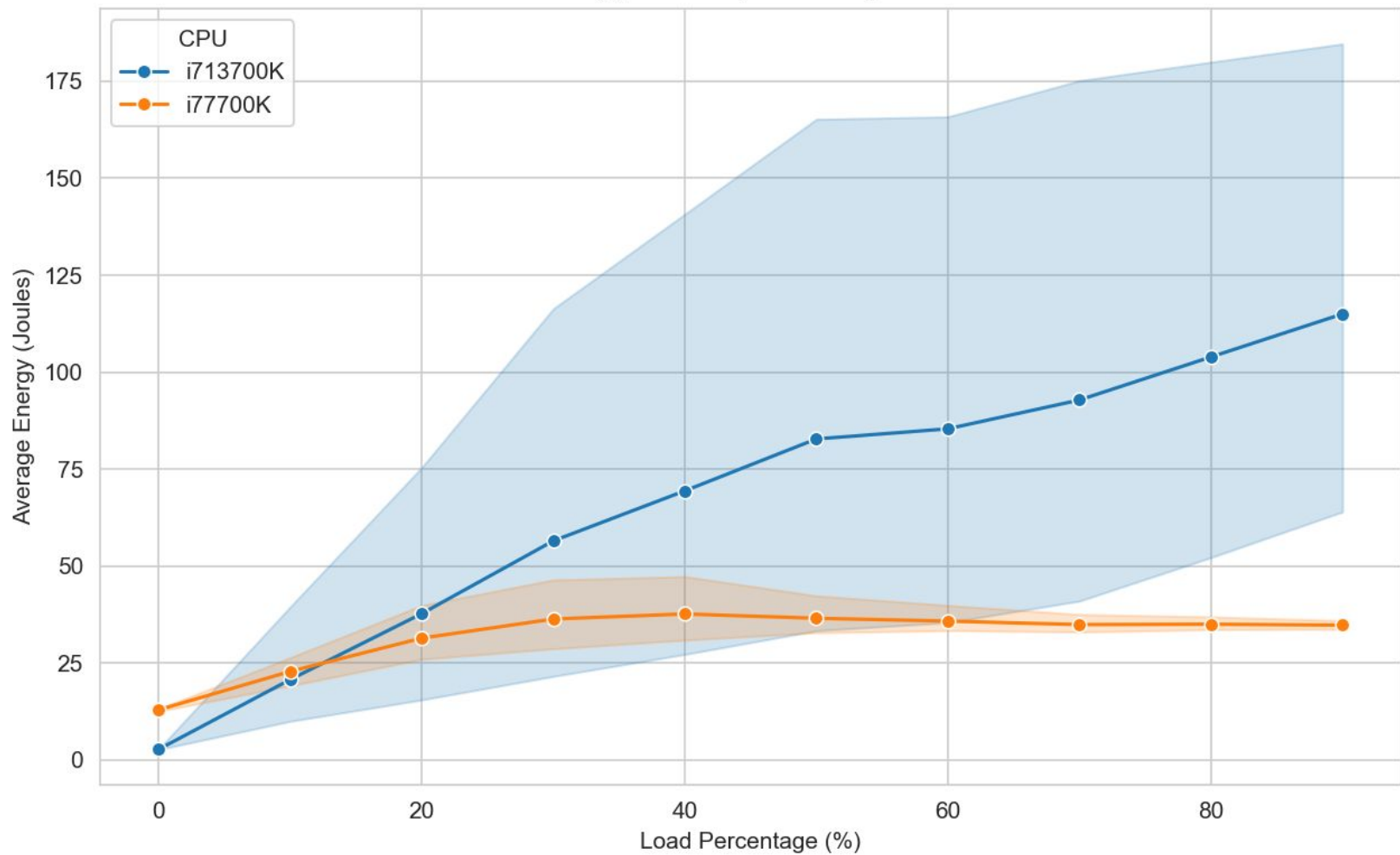
Hypothesis 1

"Multi-core CPUs are more energy efficient than single-core under high loads."

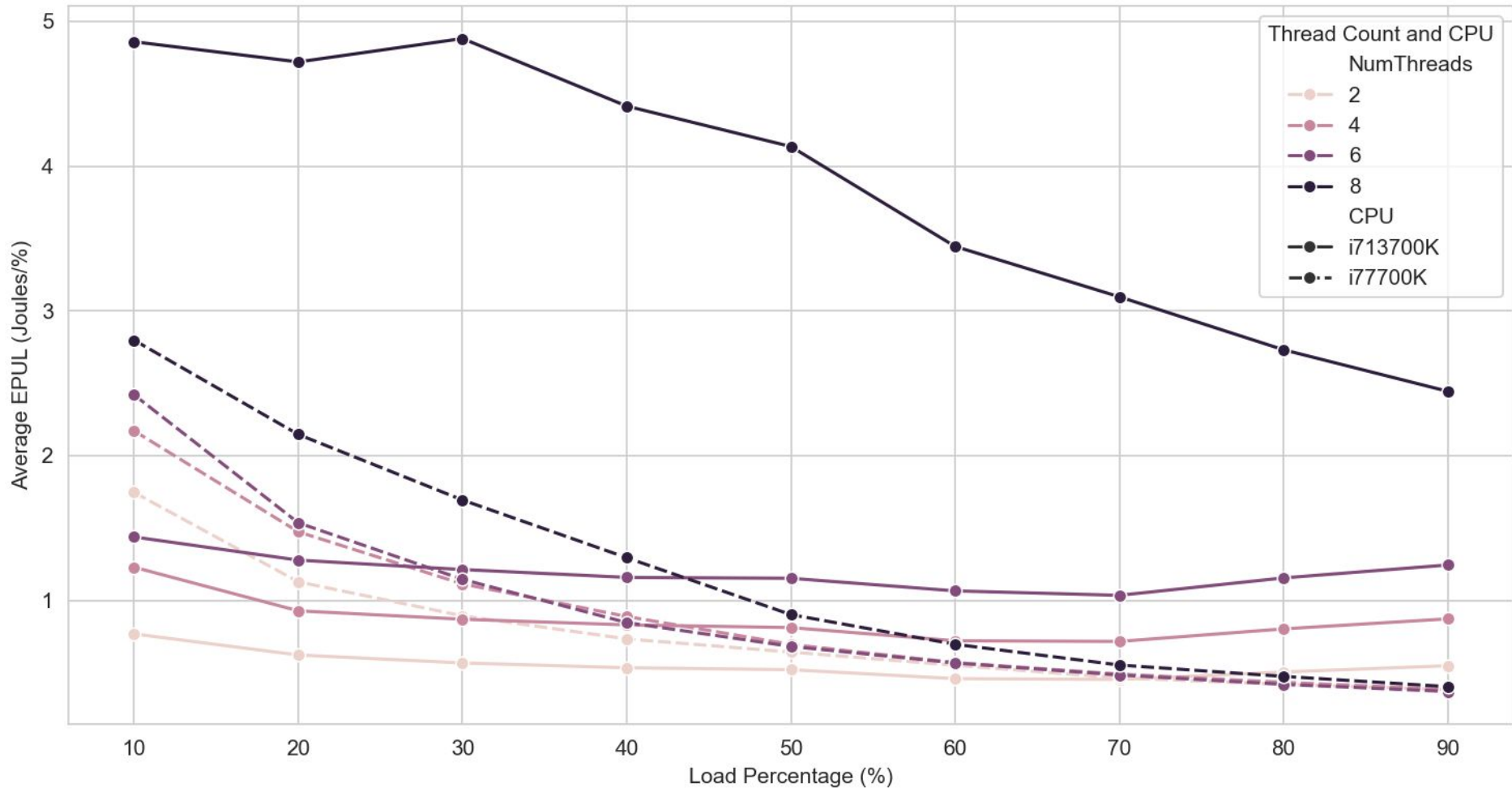
Clarification:

- **Single-Core Simulation:** Since I didn't have access to a single-core CPU, I simulated single-core performance by limiting the number of active threads (2, 4, 6, 8, or 24 threads) on multi-core CPUs.
- **Energy Efficiency Metrics Introduced:**
 - a. **Energy per Load Unit (EPUL):** $\text{Proc Energy (Joules)} / \text{LoadPercent}$
 - b. **Energy per Thread (EPT):** $\text{Proc Energy (Joules)} / \text{NumThreads}$
 - c. **Combined Efficiency Metric(CME):** $\text{Load} * \text{Threads} / \text{Energy}$

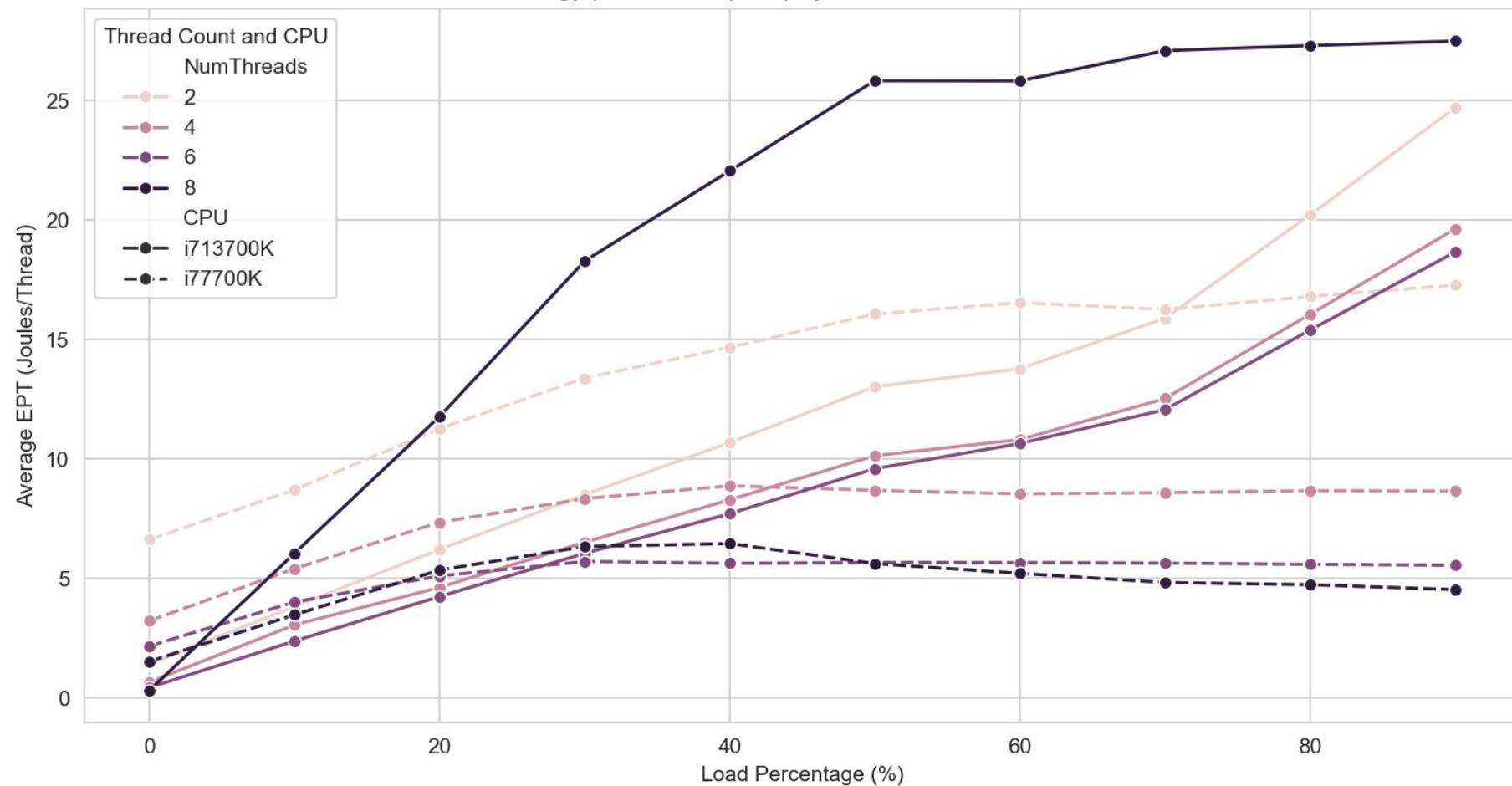
Energy Consumption Comparison



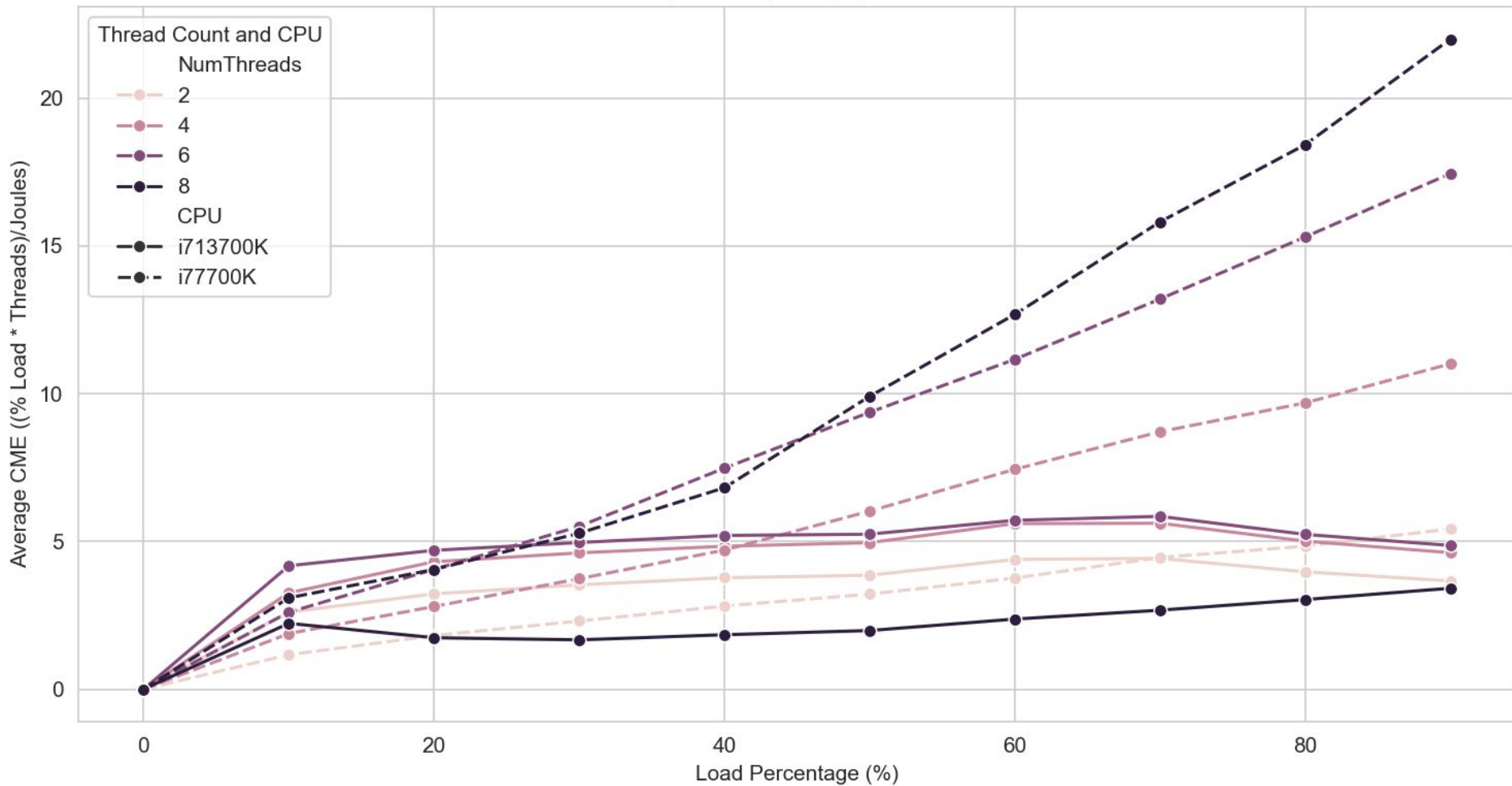
Energy per Load Unit (EPUL) by Thread Count and CPU



Energy per Thread (EPT) by Thread Count and CPU



Combined Efficiency Metric (CME) by Thread Count and CPU



Abnormal Cases

Introduction:

- During data collection and analysis, several abnormal cases and anomalies were identified.
- These anomalies impacted the accuracy and reliability of CPU energy consumption measurements.

Key Abnormal Cases Covered:

1. Unexpected Drops in Energy Consumption
2. Temperature Plateaus
3. Persistent Data Inaccuracies
4. Unexpected Energy Trends

Unexpected Drops in Energy Consumption

Observation:

- Energy consumption peaked around 30–40% CPU load and unexpectedly decreased at higher loads (>50%) for the i7-7700K.

Findings:

- Data from the i7-13700K aligned more closely with expected energy consumption trends, suggesting improved sensor accuracy and thermal management.
- The i7-7700K exhibited more pronounced energy consumption drops, indicating potential inefficiencies or tool limitations.

Temperature Plateaus

Observation:

- CPU temperatures rose with frequency up to a certain point, then plateaued despite increasing loads.

Findings:

- Temperature data plateaued inconsistently across CPUs, with the i7-13700K showing more accurate and reliable temperature trends.
- The i7-7700K's temperature readings were less precise, leading to difficulties in detecting subtle thermal changes.

Persistent Data Inaccuracies

Observation:

- Recurrent anomalies such as negative energy values, CPU utilizations exceeding 100%, and mismatched line counts between PCM and temperature data.

Findings:

- Despite mitigation, some inaccuracies persisted, highlighting the need for more reliable monitoring tools or cross validation methods.

Limited Temperature Precision

Technical Insight:

- Limited precision in temperature sensors (whole-number readings) hindered the detection of subtle thermal fluctuations.

Impact on Data:

- Flat temperature trends at high loads prevented accurate analysis of thermal dynamics.

Specific Case:

- The i7-13700K maintained accurate temperature trends due to advanced sensor technology, while the i7-7700K's data was less reliable.

Summary of Abnormal Cases

Key Abnormal Cases:

1. **Unexpected Drops in Energy Consumption**
2. **Temperature Plateaus**
3. **Persistent Data Inaccuracies**
4. **Unexpected Energy Trends**

Overall Implications:

- Highlights the need for reliable monitoring tools.
- Emphasize the impact of CPU architecture on data accuracy and energy efficiency analysis.

Discussion

- **Interpretation of Findings**
- **Impact of Tool Inaccuracies**
- **Architectural Insights**
- **Operating System Implications**

Interpretation of Findings

Energy Efficiency:

- OS Efficiency
- Multi-Core Benefits
- CPU Comparison

Unexpected Energy Trends:

- Energy Dips at High Loads

Temperature Dynamics:

- Thermal Stability
- Sensor Precision Limitations

Impact of Tool Inaccuracies

- Negative Energy Values
- CPU Utilization Over 100%
- Line Count Mismatches
- Temperature Resolution Limits
- Overall Impact

Architectural Insights

- i7-13700K Advancements
- i7-7700K Limitations
- Comparative Performance
- Implications for Research

Operating System Implications

- Linux vs. Windows Energy Efficiency
- Windows Variability
- OS Design Impact:
- Cross-OS Comparisons
- Future Research Directions

Summary and Conclusions

Project Focus:

- Investigated inaccuracies in CPU performance data collected from monitoring tools across Windows and Linux.

Key Findings:

- **Data Inaccuracies:** Significant anomalies identified in PCM and temperature sensor data.
- **CPU Architecture:** i7-13700K showed better scalability and measurement accuracy compared to i7-7700K.

Recommendations for Future Studies

Tool Enhancement and Validation

- **Cross-Validation with External Hardware**
- **Diversify Monitoring Tools**

Improved Data Collection Techniques

- **Synchronization Strategies**
- **Sampling Optimization**
- **Multi-Threaded and Multi-Core Scenarios**