

Performance Report: Python VS. Rust

Rust Performance

Metric	Value
Execution Time	1.702875 ms
Memory Usage Before	17,029,037 KB
Memory Usage After	17,029,037 KB
Memory Consumed	0 KB

Python Performance

Metric	Value
Execution Time	0.30 ms
Memory Usage Before	33,984 KB
Memory Usage After	34,000 KB
Memory Consumed	16 KB

Comparison summary

Language	Execution Time	Memory Usage Before	Memory Usage After	Memory Consumed
Rust	1.702875 ms	17,029,037 KB	17,029,037 KB	0 KB
Python	0.30 ms	33,984 KB	34,000 KB	16 KB

Improvements in speed and resource usage

- **Speed:** Python was faster for this simple task, with an execution time of 0.30 ms versus Rust's 1.70 ms. Python's high-level optimizations can sometimes give it an edge in simpler operations.
- **Memory Efficiency:** Rust showed zero memory growth, keeping memory usage constant, while Python consumed an additional 16 KB. Rust's strict memory management makes it highly efficient, especially for larger applications.
- **Best Use Cases:** Rust: Ideal for resource-intensive tasks where memory predictability and efficiency are crucial. Python: Great for rapid prototyping and tasks where speed in development outweighs strict resource control.
- **Conclusion:** Rust offers advantages in resource management and predictable memory usage, which can lead to better performance in high-demand applications. Python, however, remains a strong choice for rapid development and tasks where execution speed is prioritized over strict memory management.