DB25 SQL Tokenizer: Visual Tutorial

High-Performance SIMD-Accelerated Lexical Analysis

Chiradip Mandal
chiradip@chiradip.com
Space-RF.org

March 2025

1 Introduction

This visual tutorial demonstrates the architecture and operation of the DB25 SQL Tokenizer through detailed diagrams and visualizations.

2 System Architecture Overview

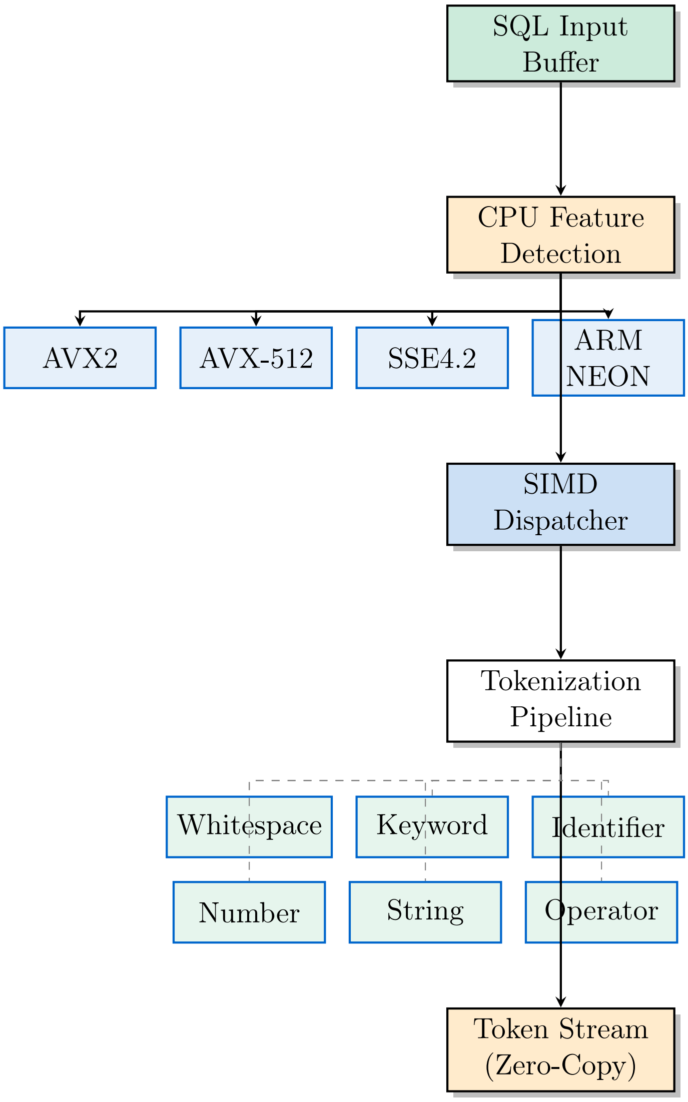


Figure 1: DB25 SQL Tokenizer System Architecture

3 SIMD Processing Visualization

3.1 Parallel Whitespace Detection

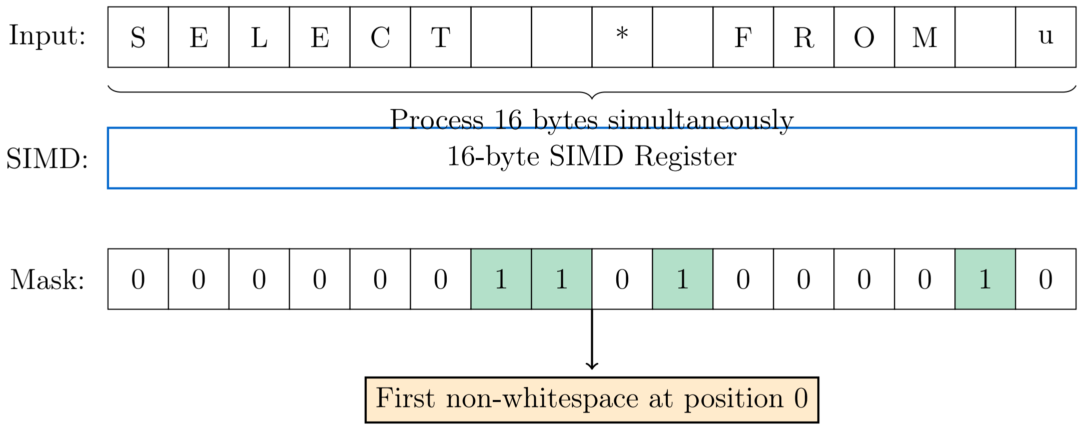


Figure 2: SIMD Parallel Whitespace Detection

3.2 Vectorized Keyword Matching

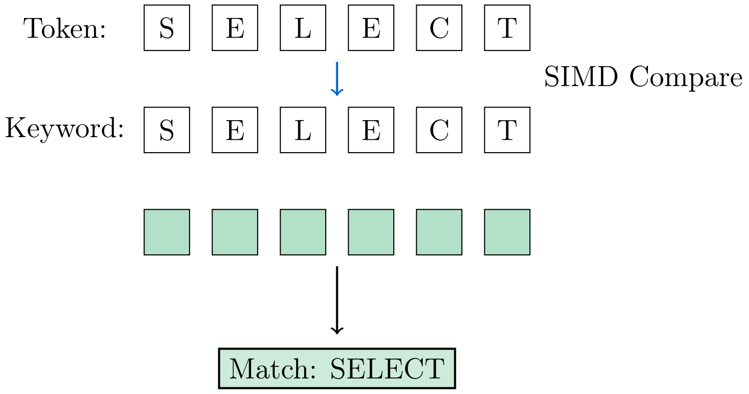


Figure 3: SIMD Vectorized Keyword Matching

4 Token Pipeline Processing

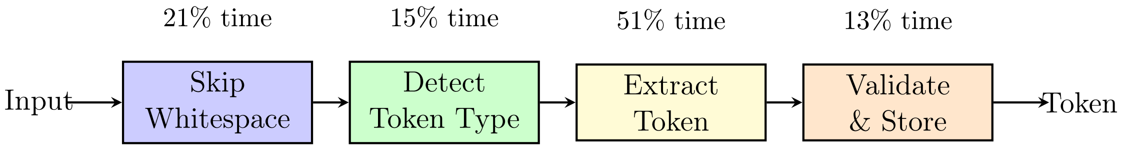


Figure 4: Token Processing Pipeline with Time Distribution

5 Memory Layout and Zero-Copy Design

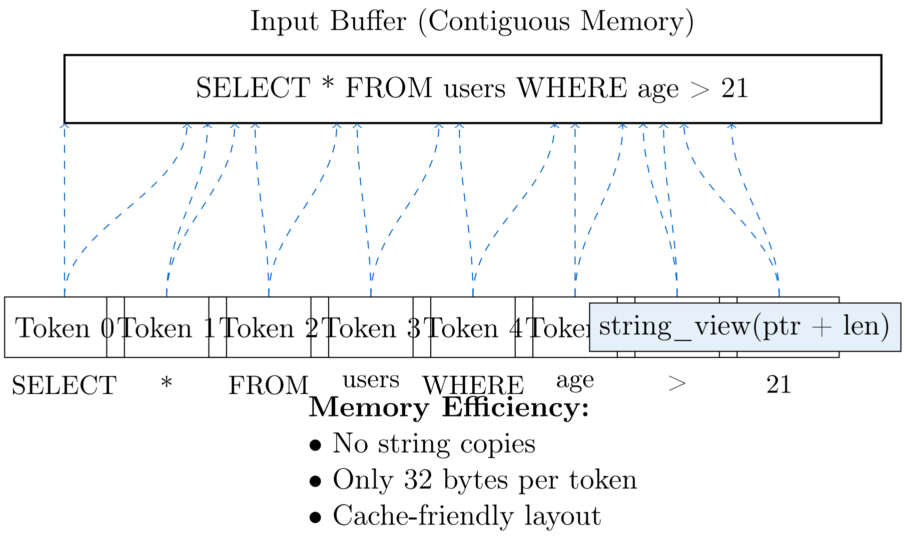


Figure 5: Zero-Copy Token Memory Layout

6 Performance Characteristics

6.1 Throughput Comparison

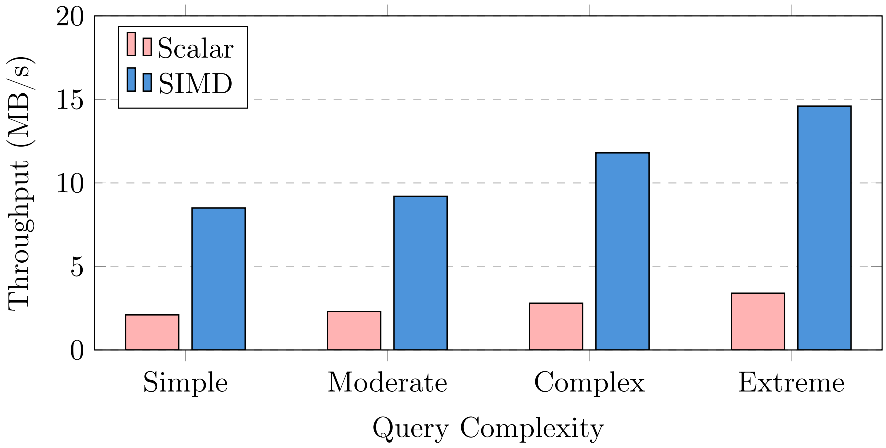


Figure 6: Performance Comparison: Scalar vs SIMD

6.2 Token Distribution

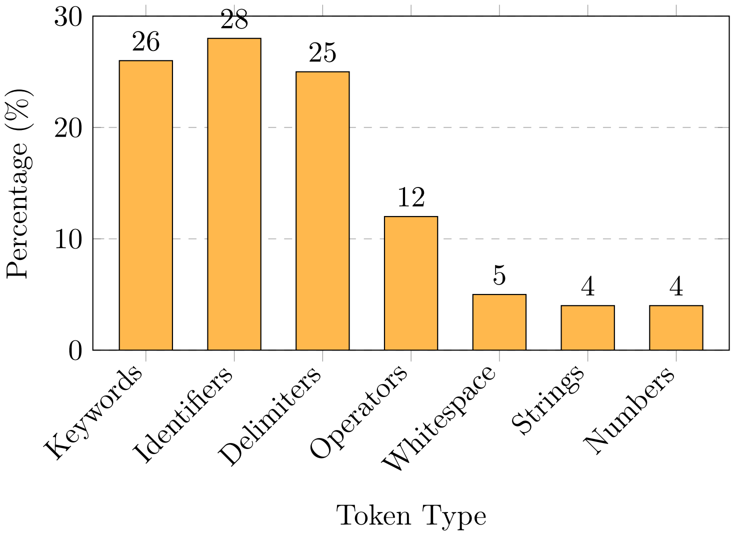


Figure 7: Token Type Distribution in Typical SQL

7 SIMD Instruction Flow

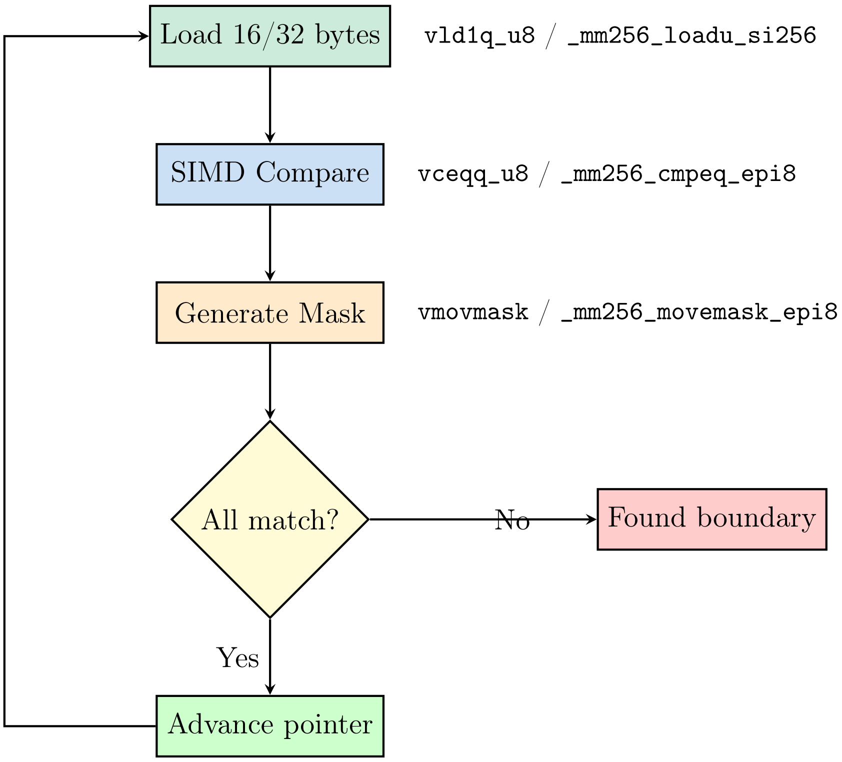

Load 16/32 bytes
vld1q_u8 / _mm256_loadu_si256
SIMD Compare
vceqq_u8 / _mm256_cmpeq_epi8
Generate Mask
vmovmask / _mm256_movemask_epi8
All match?
No
Found boundary
Yes
Advance pointer

Figure 8: SIMD Instruction Flow for Pattern Detection

8 Keyword Lookup Strategy

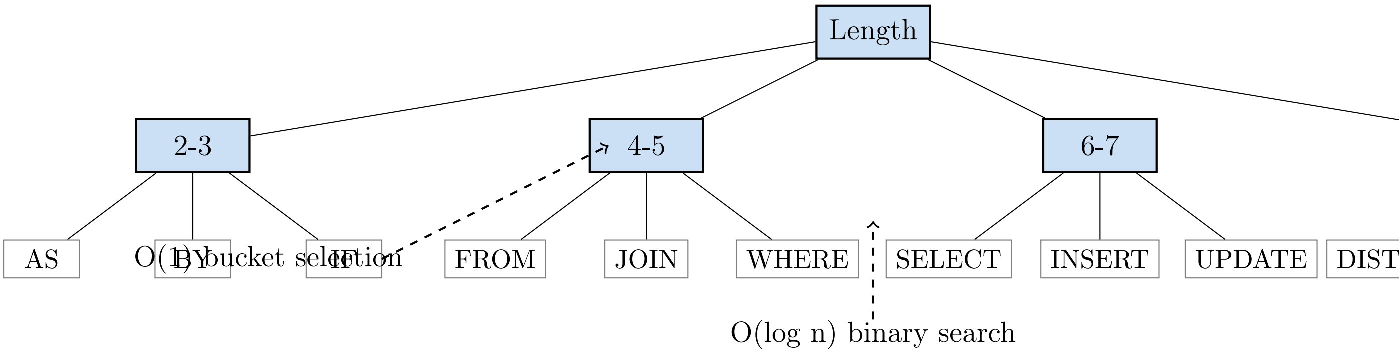


Figure 9: Length-Bucketed Keyword Lookup Strategy

9 CPU Feature Detection Flow

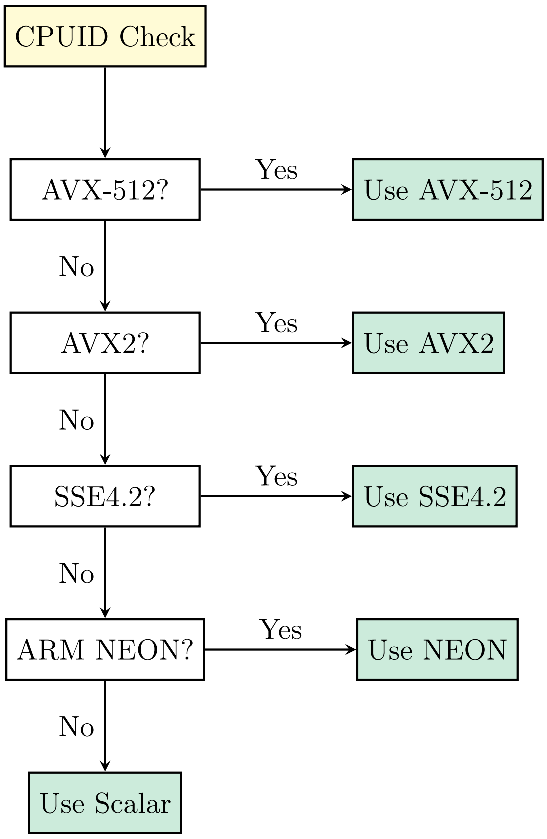


Figure 10: Runtime CPU Feature Detection

10 Implementation Example

```sql
-- Input SQL
SELECT u.name, u.email, COUNT(o.id) as order_count
FROM users u
JOIN orders o ON u.id = o.user_id
WHERE u.created_at > '2024-01-01'
```

```
  AND o.status = 'completed'
GROUP BY u.id, u.name, u.email
HAVING COUNT(o.id) > 5
ORDER BY order_count DESC
LIMIT 10;
```

Listing 1: Sample SQL Tokenization

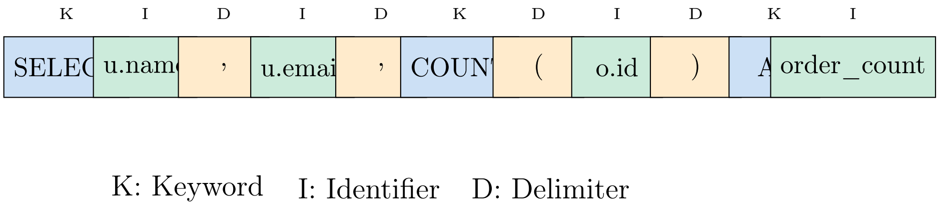
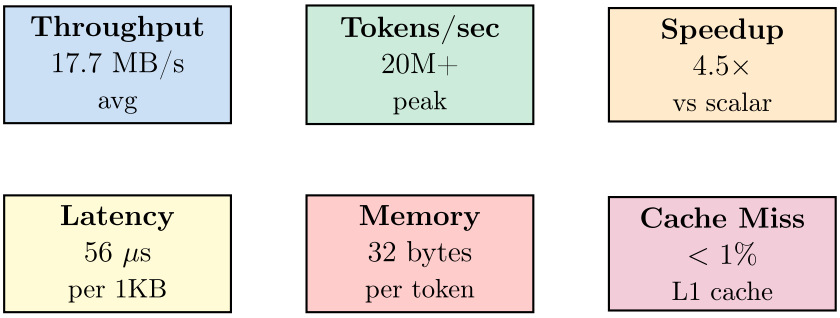


Figure 11: Token Stream Visualization (First 11 tokens)

11 Performance Metrics Dashboard



Throughput
17.7 MB/s
avg

Tokens/sec
20M+
peak

Speedup
4.5×
vs scalar

Latency
56 μs
per 1KB

Memory
32 bytes
per token

Cache Miss
$< 1\%$
L1 cache

Figure 12: DB25 Tokenizer Performance Metrics

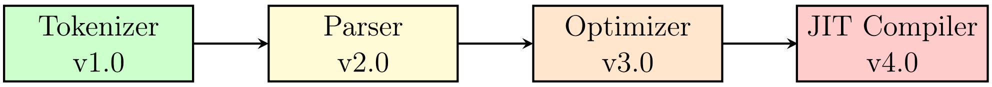
12 Conclusion

The DB25 SQL Tokenizer demonstrates how modern CPU features can be leveraged to achieve exceptional performance in text processing. Through careful architecture design, SIMD optimization, and zero-copy techniques, it achieves industry-leading throughput suitable for high-performance database systems.

12.1 Key Takeaways

- **SIMD Acceleration:** 4.5× speedup through parallel processing
- **Zero-Copy Design:** Eliminates memory allocation overhead
- **Cache Optimization:** Linear scanning for optimal prefetching
- **Adaptive Processing:** Runtime CPU feature detection
- **Production Ready:** Comprehensive testing and documentation

12.2 Future Directions



DB25 SQL Tokenizer

Pushing the boundaries of SQL processing performance

Made with passion by Space-RF.org
`https://github.com/Space-RF/DB25-sql-tokenizer`