

# Performance impact of the modular architecture in the incremental SGLR parsing algorithm

## 1. Background

**JSGLR2** is a modular implementation of the **Scannerless Generalized LR parsing** algorithm. The modular architecture is believed to negatively impact the performance.

## 2. Research question

- How can the modularity impact the performance?
- Does “inlining” reduce the overhead ?
- What is the overhead?
- What is the impact on different languages?

## 3. Methodology

### Inline and compare

1. Inline the modular algorithm.
2. Compare the two versions.

## 4. Results

### Batch parsing time speedup:

- 2% for Java
- 10% for SDF3
- 16% for WebDSL

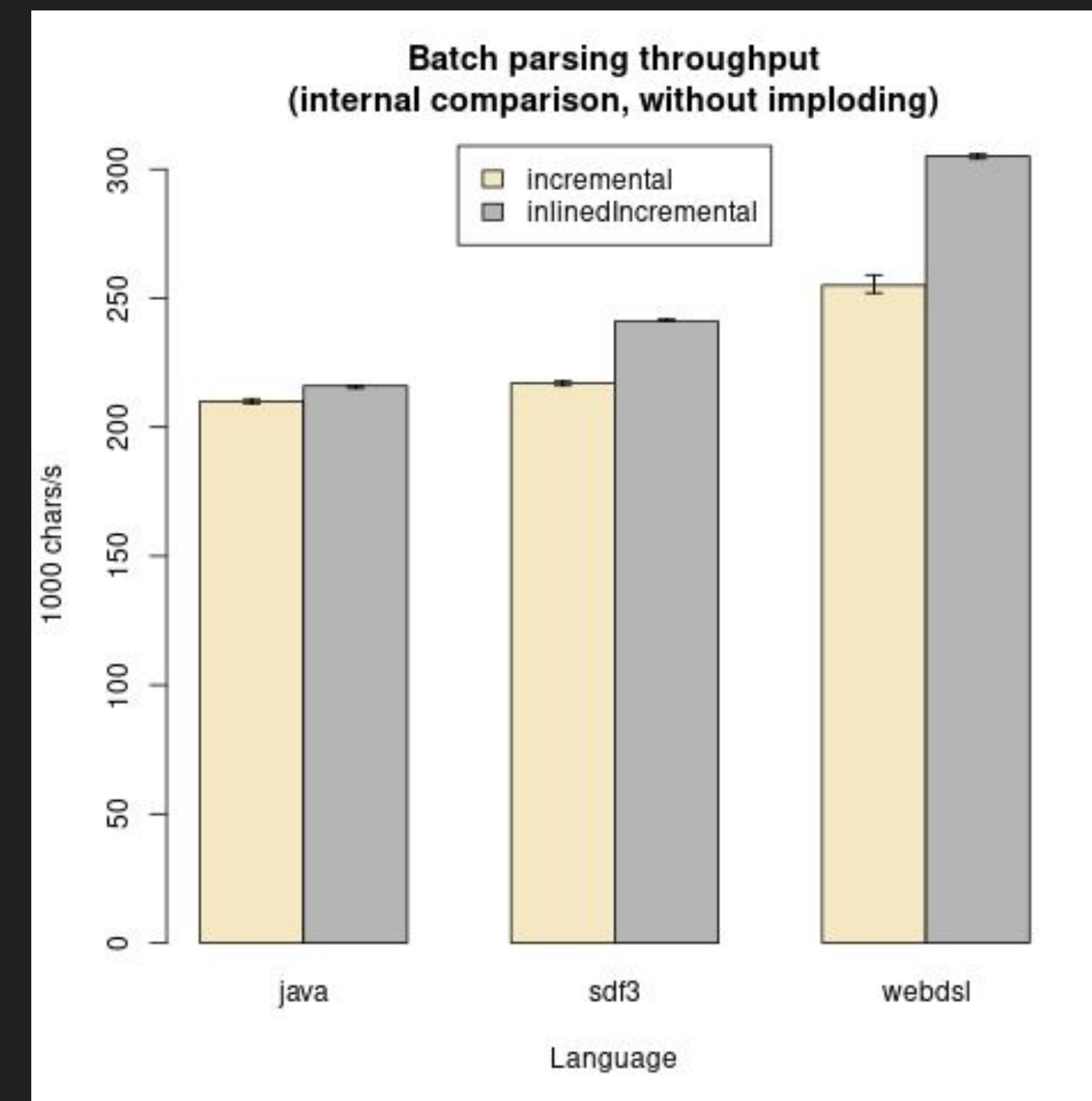
### Incremental parsing time speedup:

- 2%
- 6%
- 10%

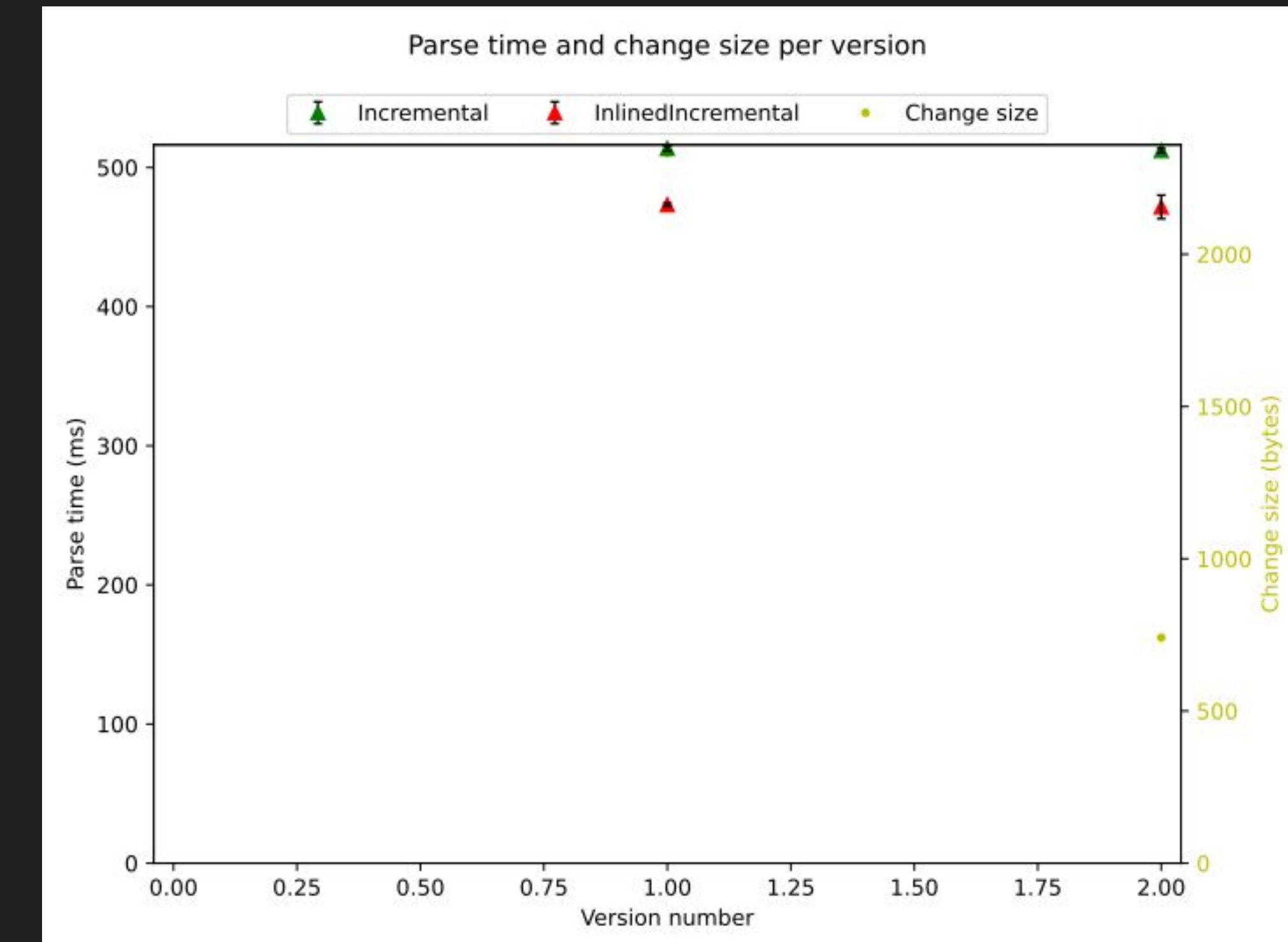
## 5. Conclusion

- Removing the modular structure improves performance
- Inlining can be used to optimize a given codebase

Batch parsing throughput without imploding



Incremental parsing time for a Java repository



Batch parsing throughput with imploding



Incremental parsing time for a SDF3 repository

