SourcererCC: Scaling Code Clone Detection to Big Code

This paper introduced a tool SourcererCC that consider overlapping when tokenizing the source code. Such approach showed great accuracy over previous tools based on tokens. SourcererCC is also the first tool claimed by the authors that had a real focus on scaling. It can handle a large code base up to 100's MLOC. The tool is also capable of detecting Type-3 of code clone which is syntactically similar code fragments that differ at the statement level. This is really ground breaking work since the other tool compared by the authors with the capability of looking at large code base does not feature this.

Things I would like to see discussed:

- What makes it fast and scalable? The paper discusses the algorithm but never went in to analysis of the complexity. The efficiency is only showed by experiment.
- Now we have huge clusters with massive computing power and memory. This paper's comparison used a commodity machine. I would like to discuss about whether clusters would make a difference.
- Configuration. The author mentioned that configuration would make a huge difference when comparing source code detection tool. What causes the difference and why configuration matters?