Abstract

In language systems that support separate compilation, the header files are internalized over and over again when the source files that depend on them are compiled. Making a compiler a long-lived server eliminates such redundant processing of header files, thus reducing the compilation time. Modern JVM implementations interleave execution with compilation of "hot" methods to achieve reasonable performance. Since compilation overhead impacts the execution time of the application and induces run-time pauses, it is better to offload compilation onto a compilation server. Compilation server is the server which compiles and optimizes Java byte codes on behalf of its clients. It provides the benefit of lower execution and pause times due to reducing the overhead of optimization. Compilation server is able to handle more than 50 concurrent clients while still allowing them to outperform best performing adaptive configuration.

Reference

Designing Compilation Server

[14] T. Onodera, Reducing Compilation Time by a Compilation Server IBM Research, Tokyo Research Laboratory, 5-19 Sanbancho, Chiyoda-ku, Tokyo 102, Japan
**Index Terms**

<table>
<thead>
<tr>
<th>Computer Science</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td></td>
</tr>
</tbody>
</table>

**Key words**

- Compilers
- Compilation Server
- JVM
- execution time
- pause time