

Evaluation of MIPS Prelinking

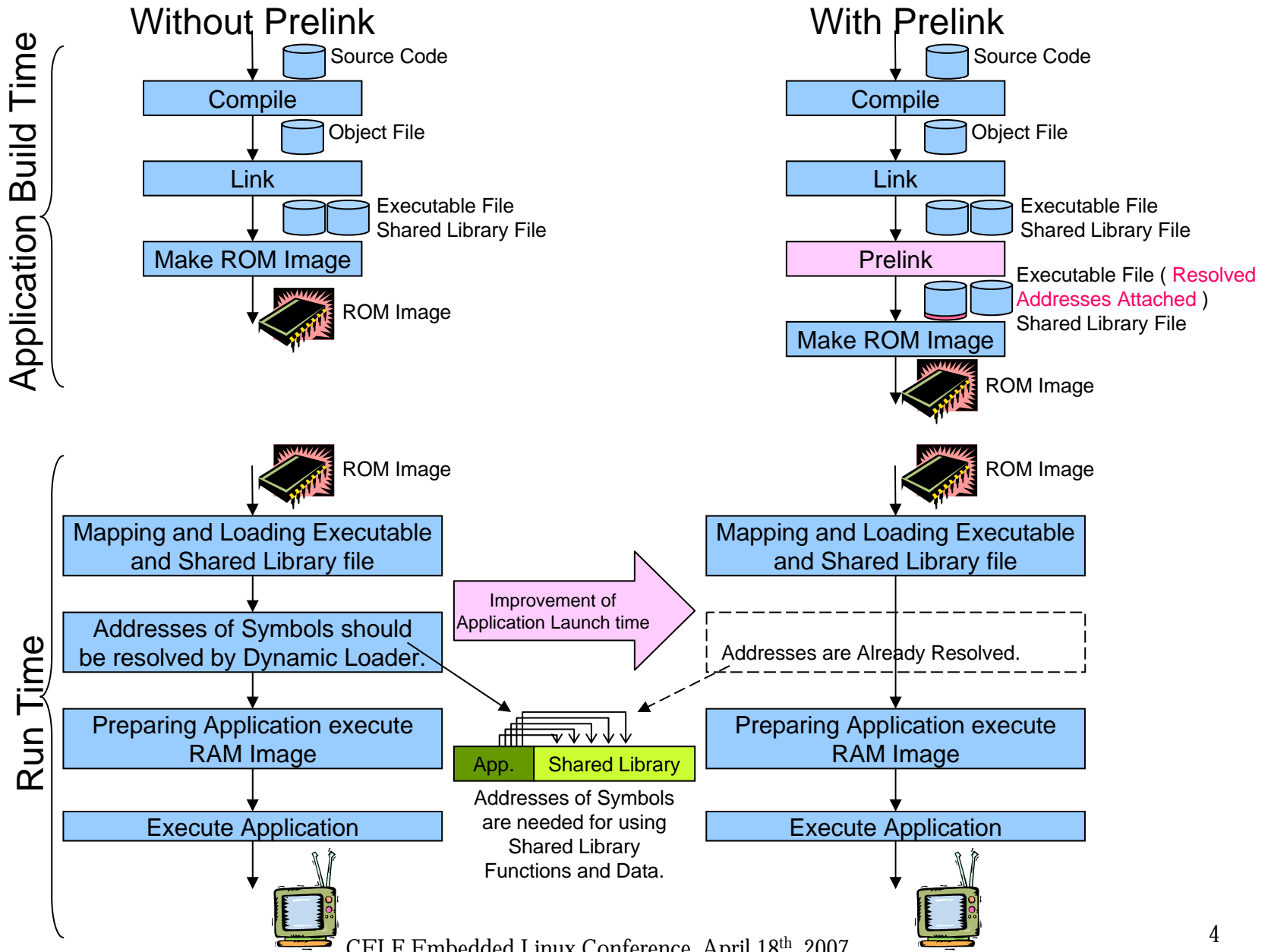
Shin'ichi TSURUMOTO

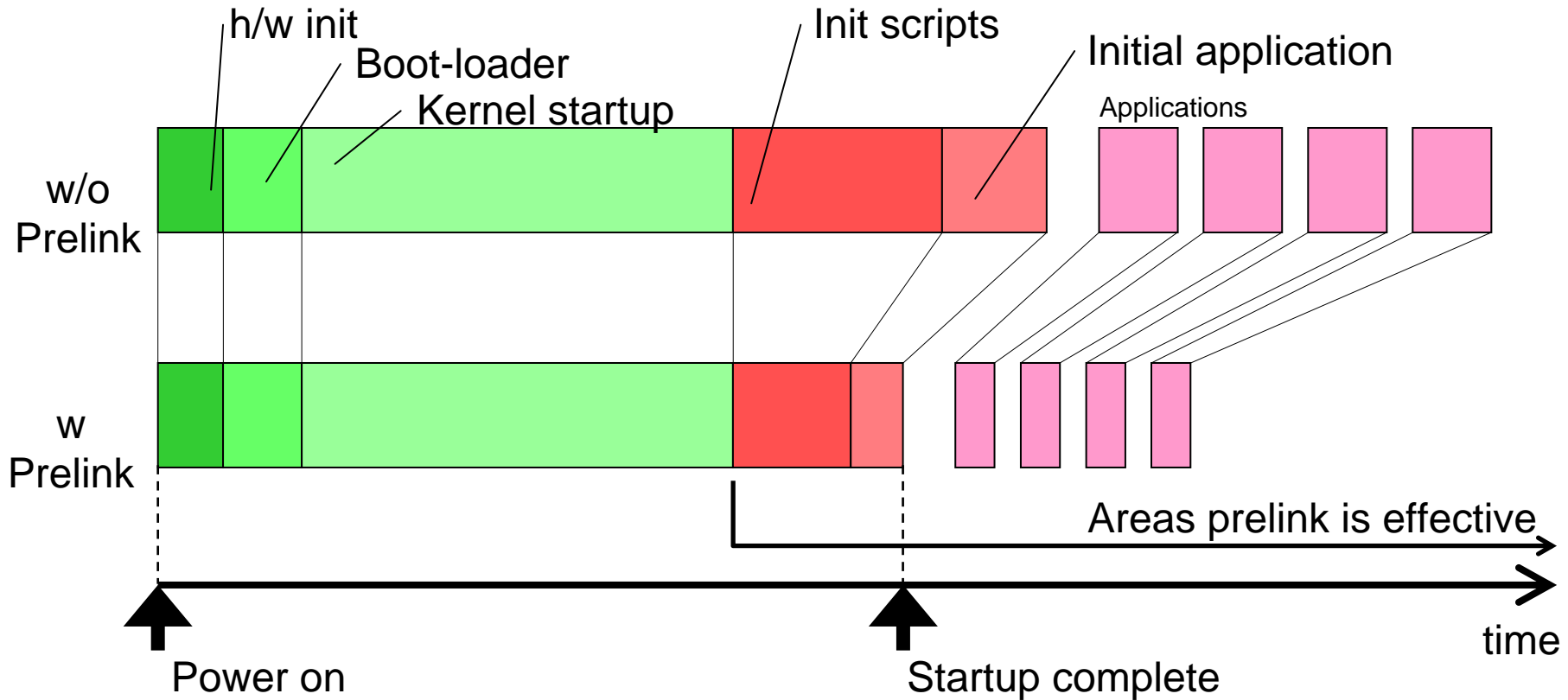
MITSUBISHI Electric Corporation
Advanced Technology R&D Center

- ◆ Obtained prelinker for MIPS, compiler and libraries, and ran them on our target board
- ◆ Measured application startup time on our target board, and verified that the prelinking is effective

- ◆ On June 2005, effectiveness of prelink technique was presented at CELF International Technical Conference in Yokohama
- ◆ On November 2005, I questioned on prelink on MIPS architecture at Tokyo Technical Jamboree
- ◆ On April 2006, I requested to implement prelink for MIPS at MIPS-SIG piggybacked on ELC (Worldwide Embedded Linux Conference)
- ◆ On July 2006, MIPS Technologies reported at Ottawa Linux Symposium that they have started to develop prelink
- ◆ On October 2006, MIPS Technologies and CodeSourcery reported at Tokyo Technical Jamboree that they completed the development

What is Prelink?





◆ Obtaining patches

- ◇ prelink.tar.gz
(<svn://sourceware.org/svn/prelink>)
- ◇ glibc-2.3.6-prelink.patch
(sourceware.org/glibc)
- ◇ binutils-2.16.1-prelink.patch
(sourceware.org)

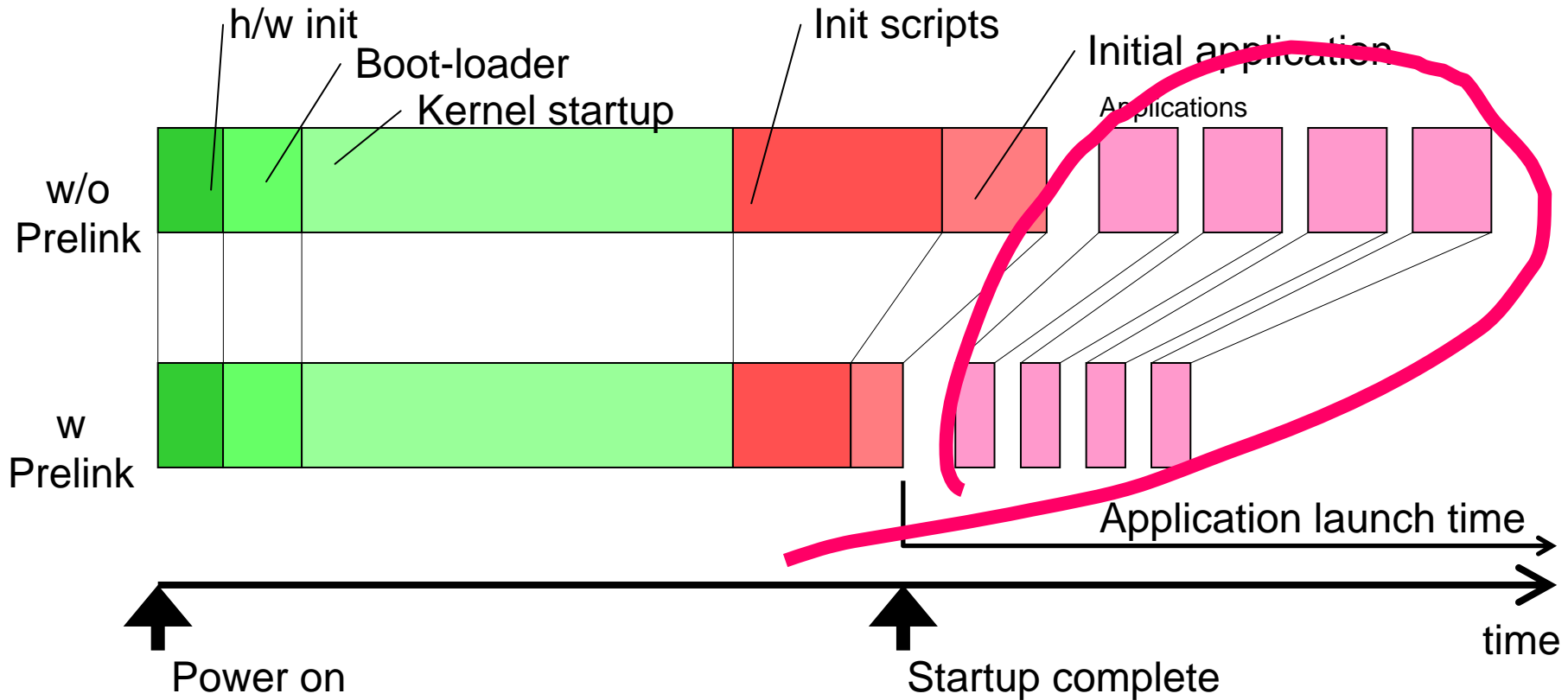
◆ Applying patches and rebuilding toolchain

◆ Compiling prelink command (we built as self-prelinker on target)

◆ Building target image

◆ Executing prelink on target

- ◆ CPU core : MIPS32(4KEc)
- ◆ Clock : 300MHz
- ◆ DRAM : 384MByte
- ◆ Flash : 64MByte



src-gen/main.c

```
#include "incgen.h"

int main( int argc, char **argv )
{
    return 0;
}
```

inc-gen/incgen.h

```
extern void func001();
extern void func002();
extern void func003();
:
```

lib-gen/func001.c

```
#include "incgen.h"
void func001() {}
```

- ◆Compiled source code above so that each library function is an independent shared library file
- ◆Used GNU autotools to compile
- ◆Written shell script to form infinite loop; measured time between start and stop of the command with using time command, repeating ten times to get the average value

src-gen/main.cpp

```
#include "incgen.h"

int main( int argc, char **argv )
{
    return 0;
}
```

inc-gen/incgen.h

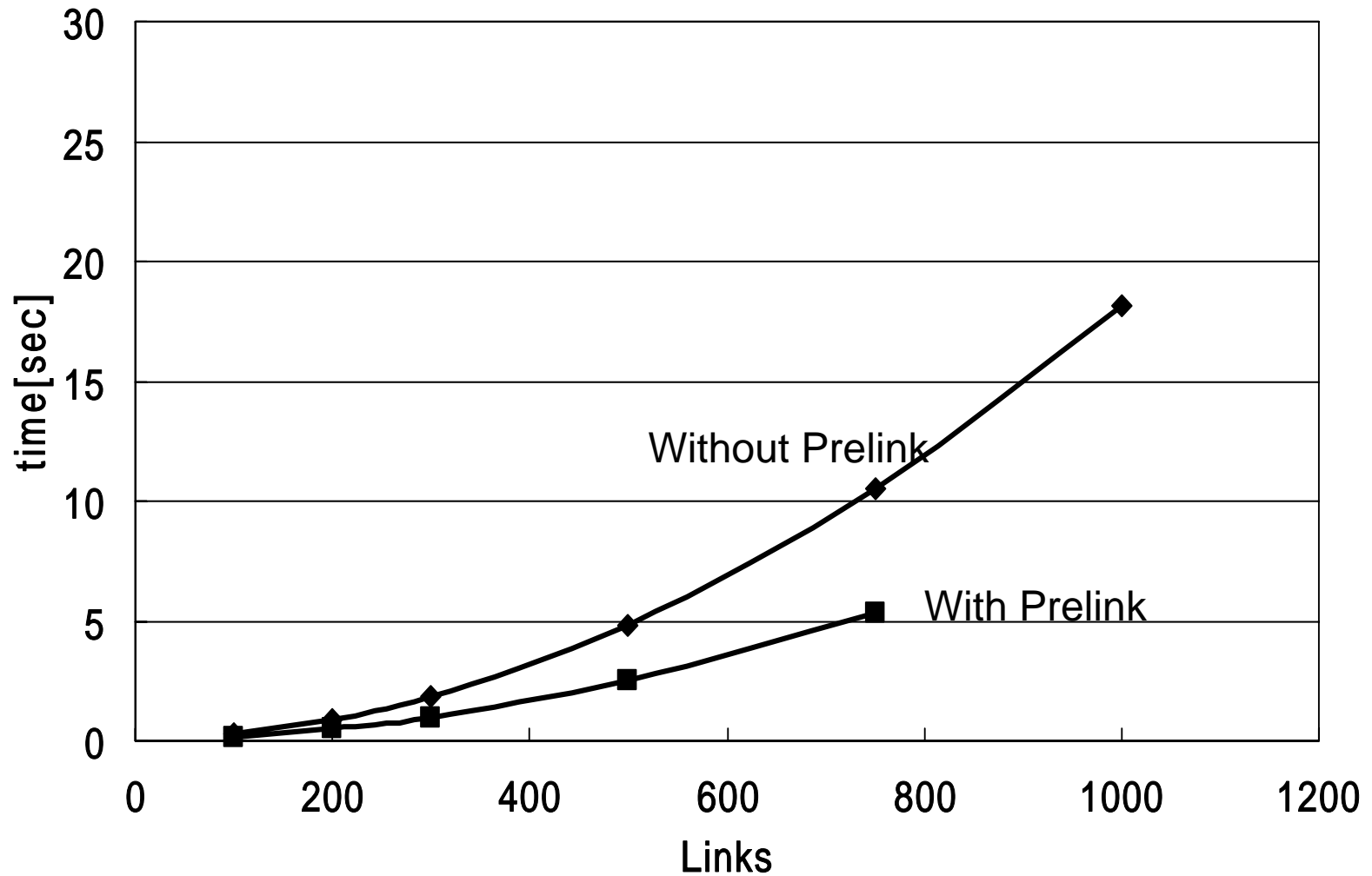
```
extern void func001();
extern void func002();
extern void func003();
:
```

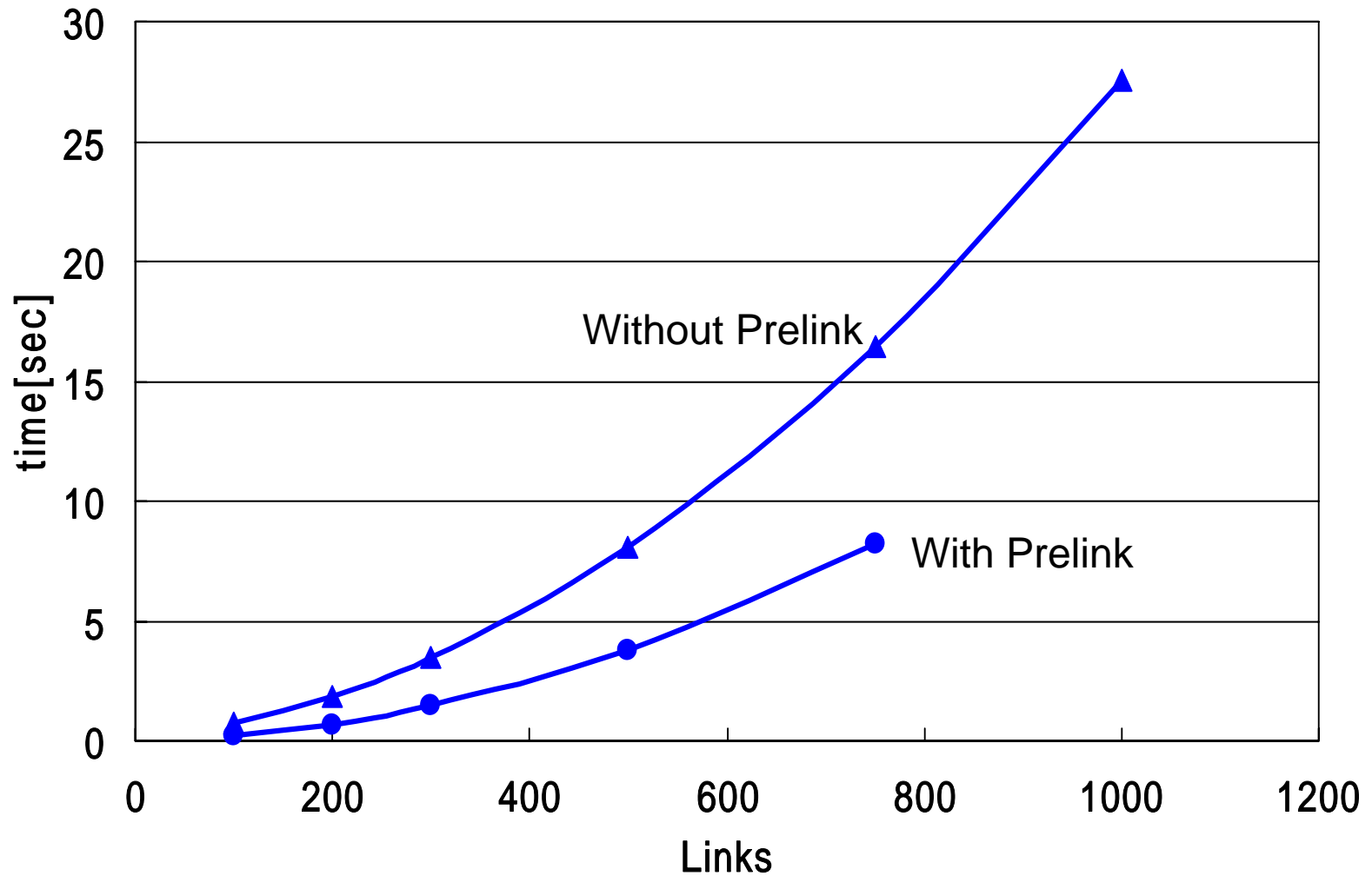
lib-gen/func001.cpp

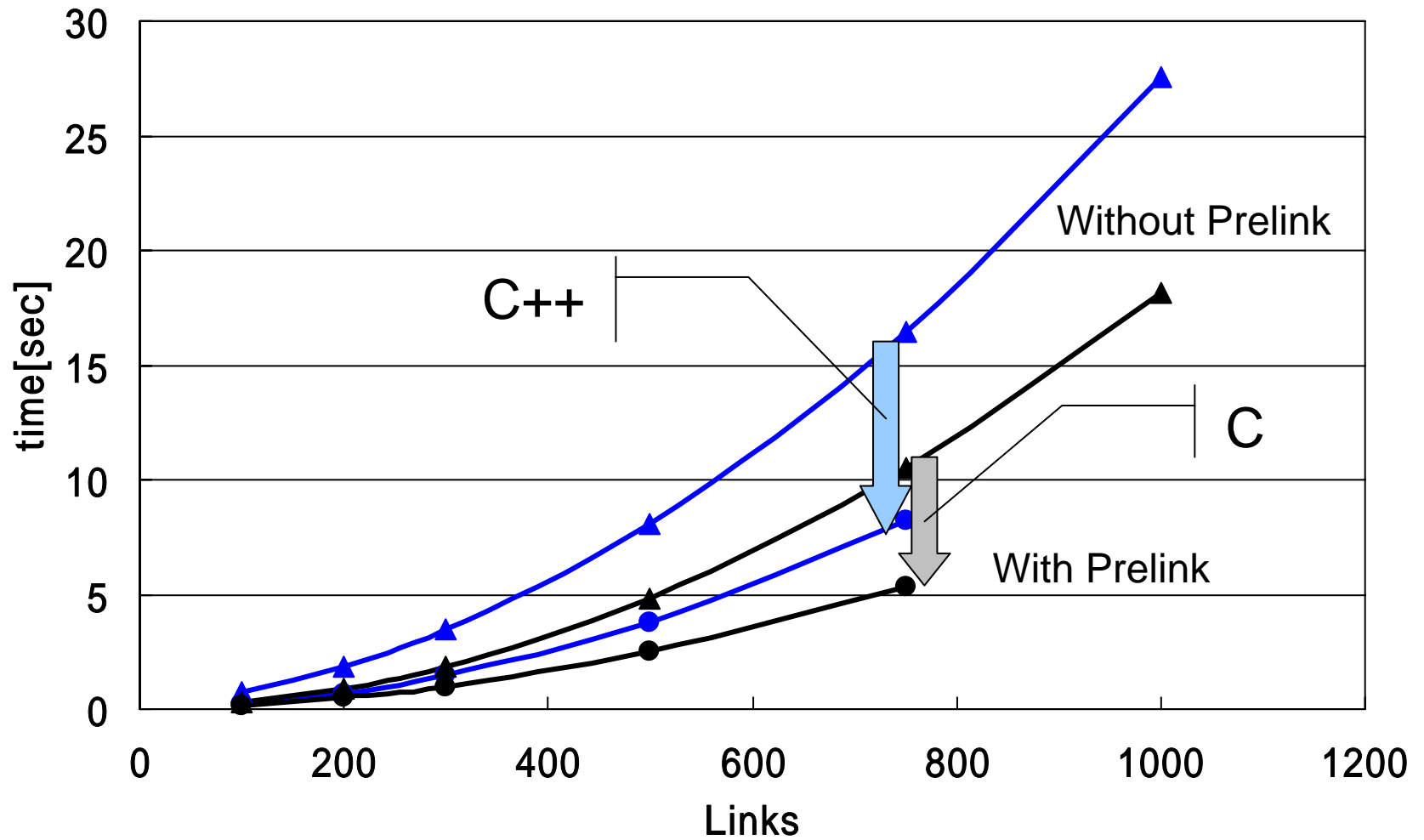
```
#include "incgen.h"
void func001() {}
```

- ◆Compiled source code above so that each library function is an independent shared library file
- ◆Used GNU autotools to compile
- ◆Written shell script to form infinite loop; measured time between start and stop of the command with using time command, repeating ten times to get the average value

Note: same test as of c edition, except for filename, extension and compiler







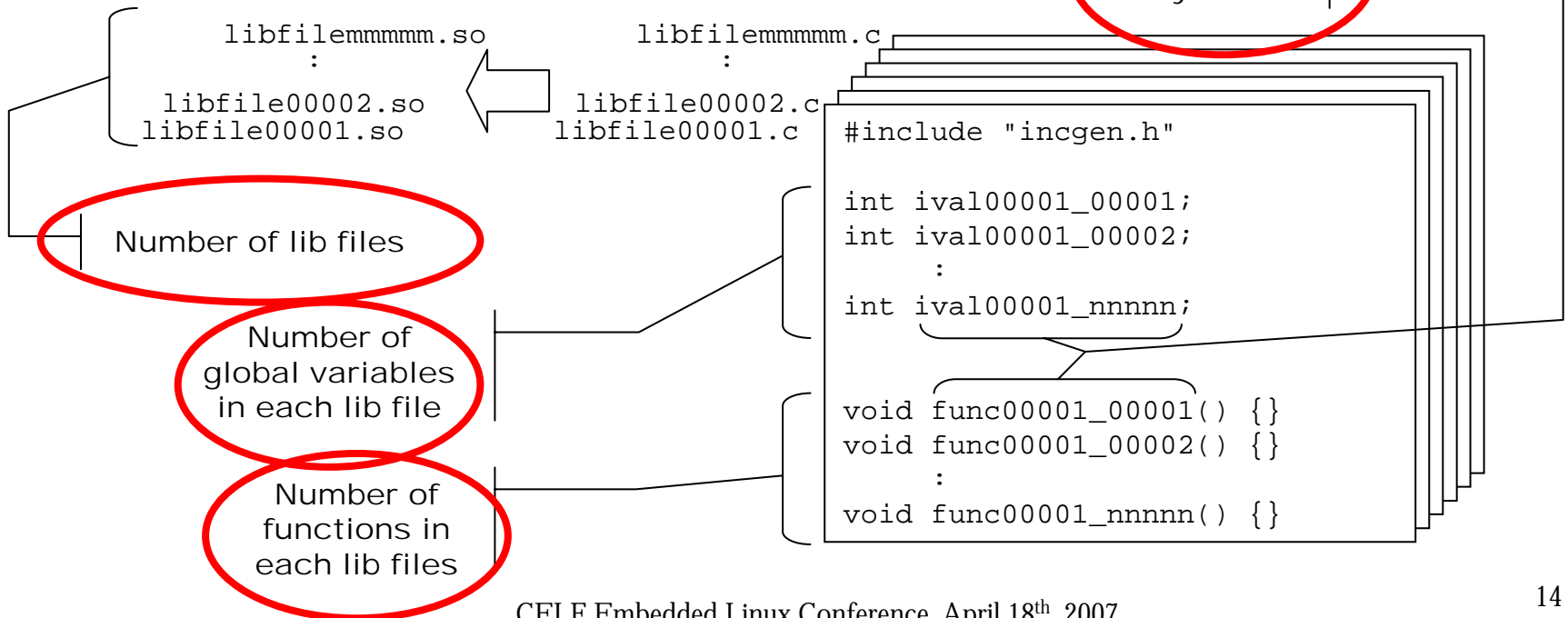
incgen.h

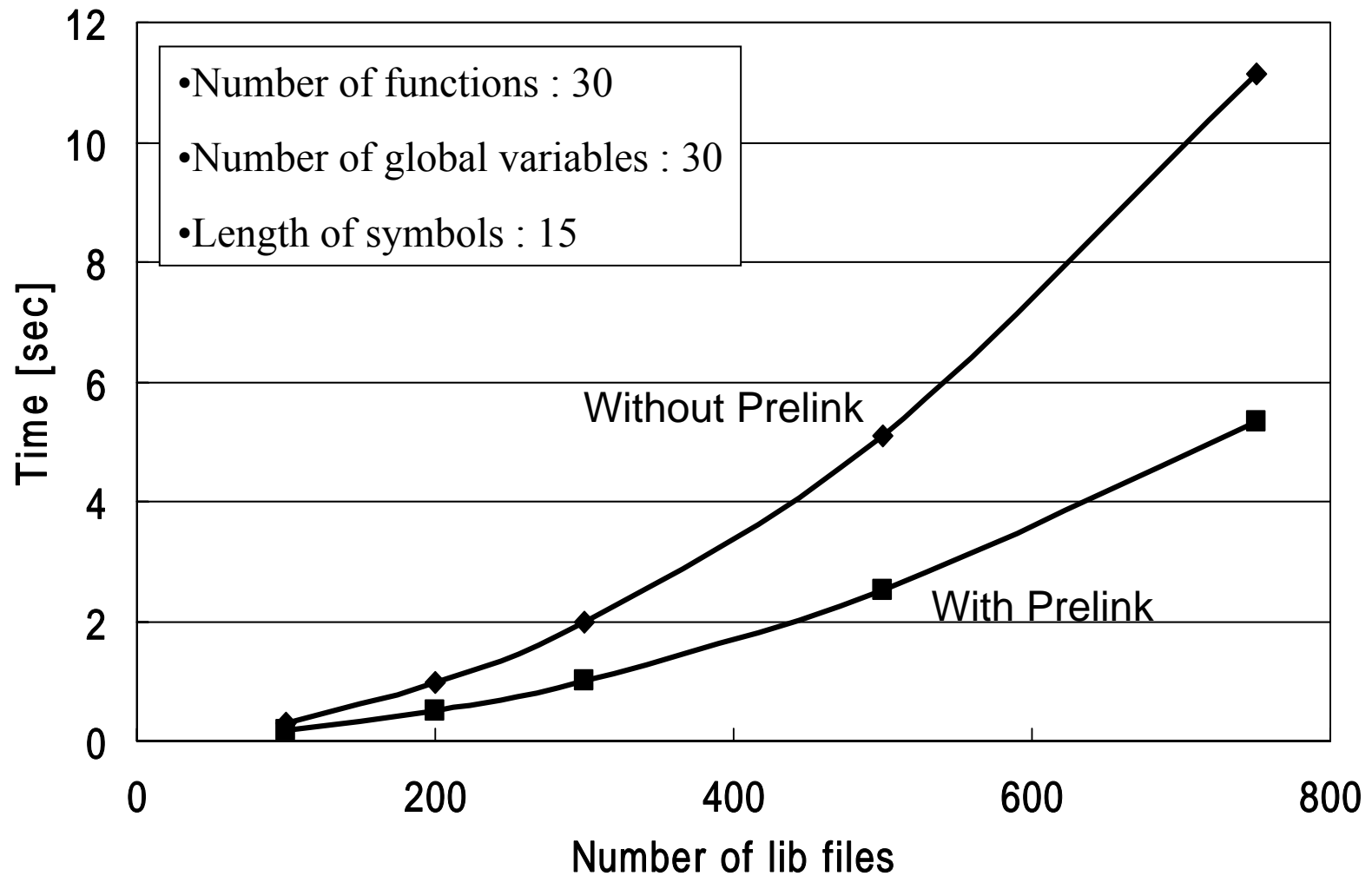
```
extern int ival00001_00001;
extern int ival00001_00002;
:
extern int ival00001_00010;

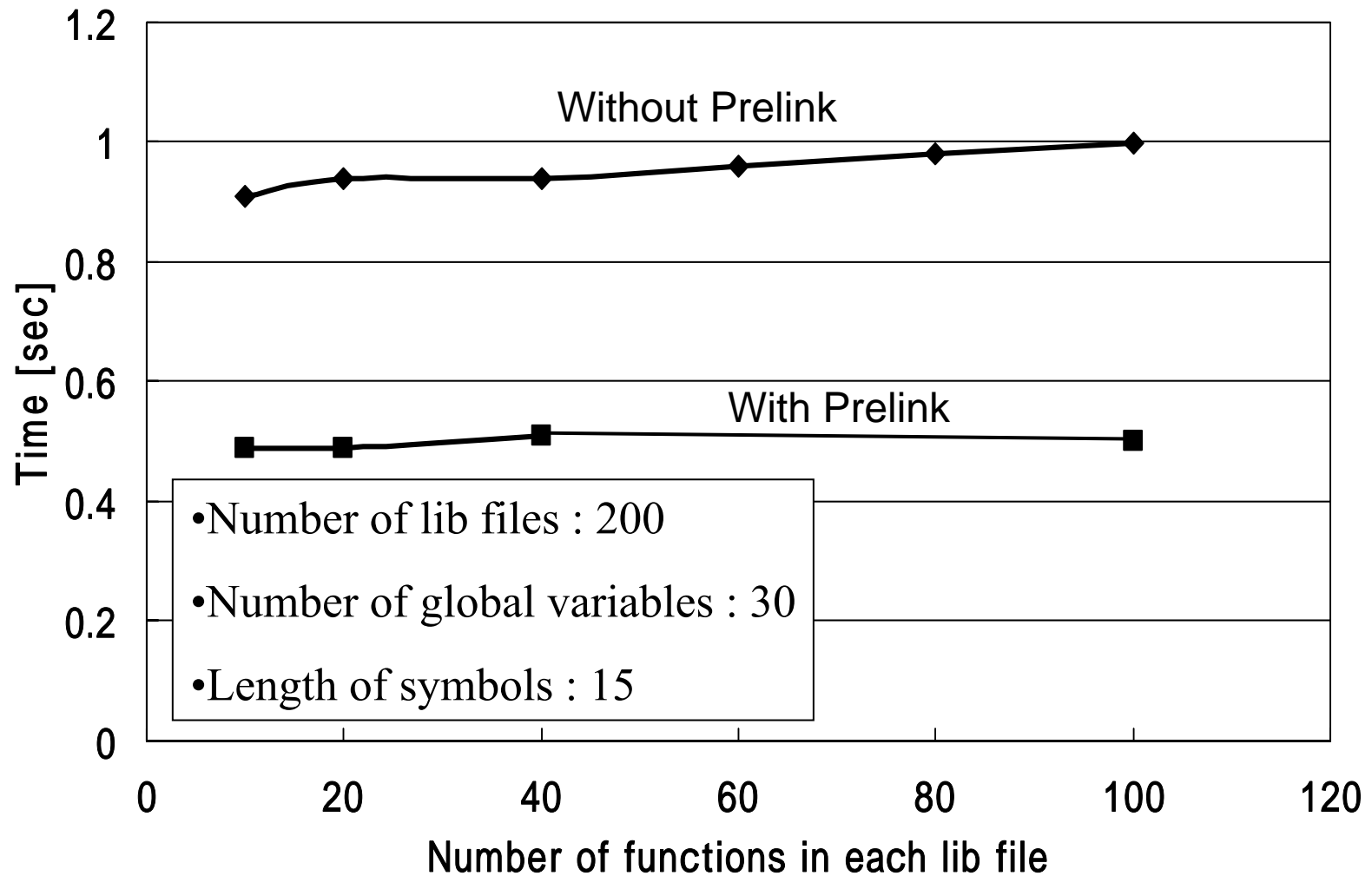
extern void func00001_00001();
:
extern void func00001_00002();
extern void func00001_00010();

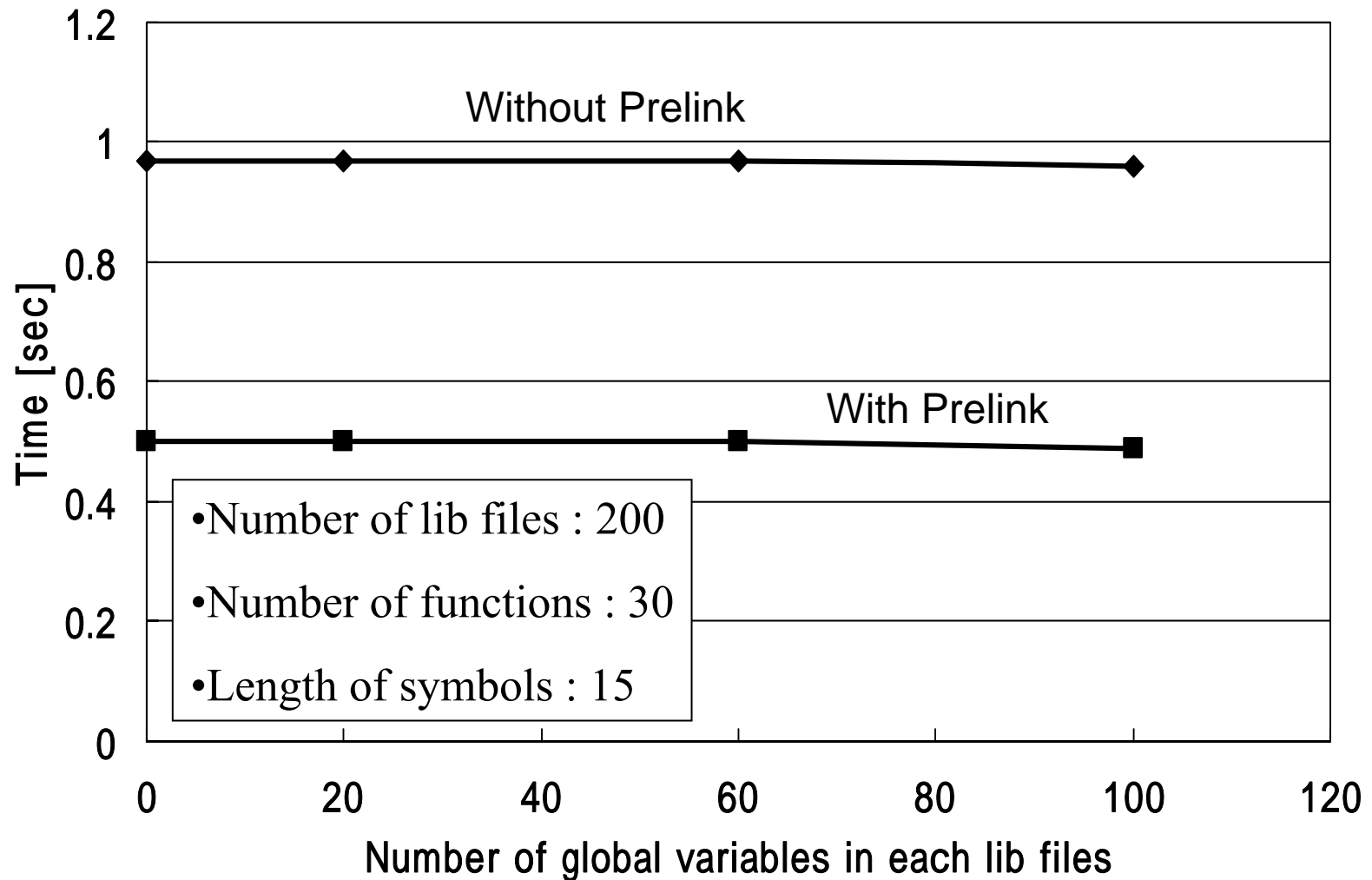
extern int ival00002_00001;
:
```

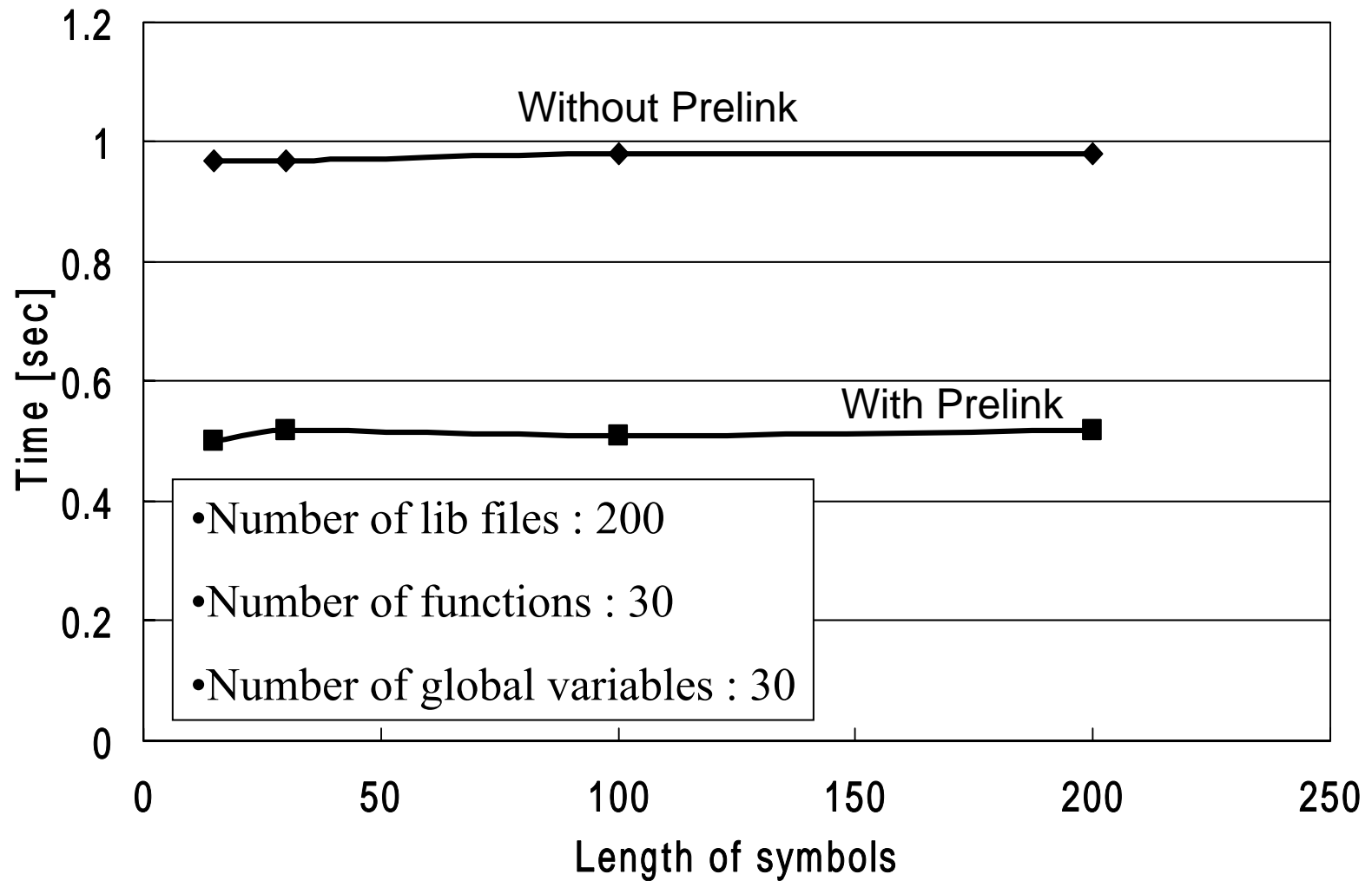
Length of
symbols



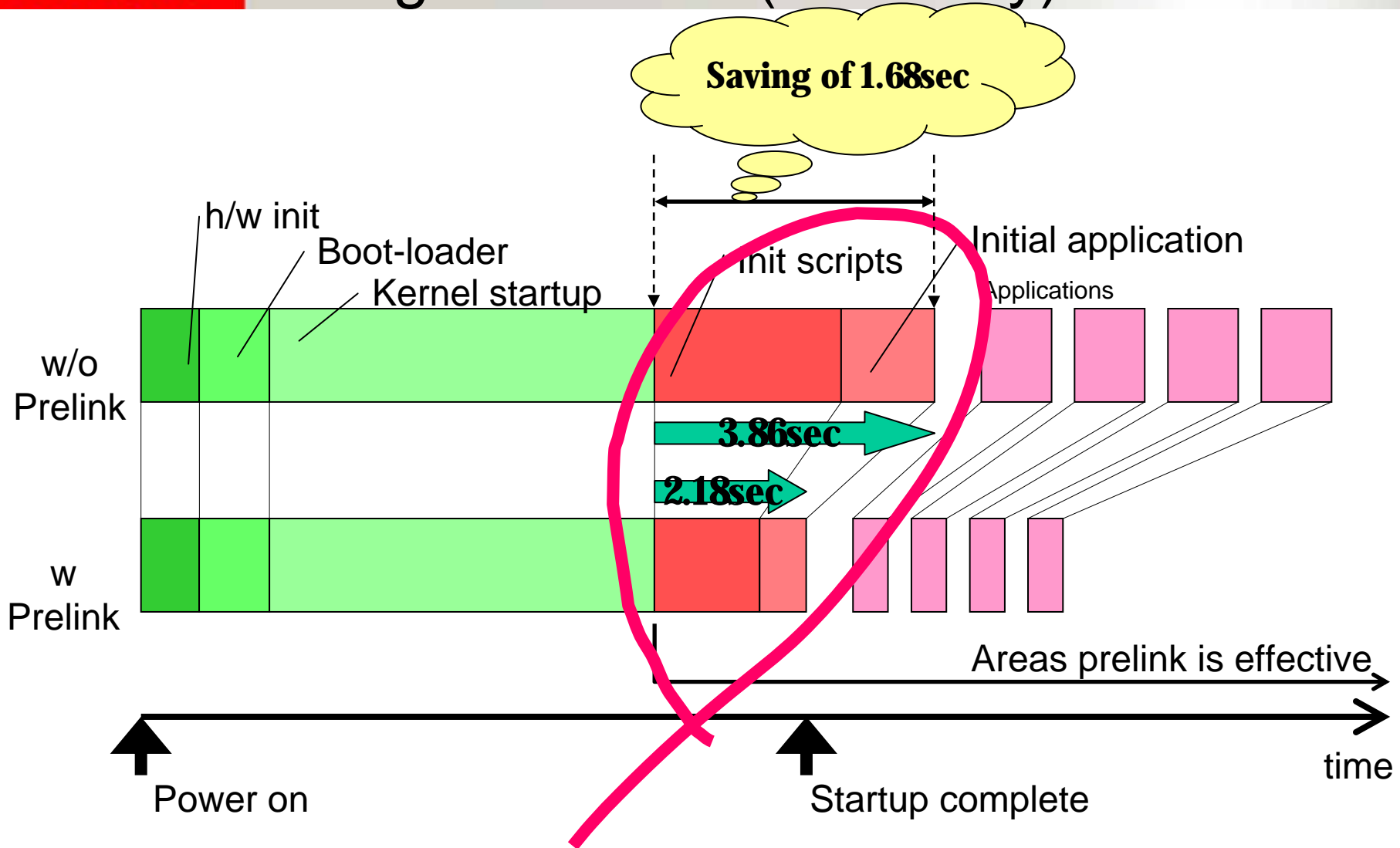








Target of test 3 (C++ only)



- ◆ At the top of the init script, just after mounting the procs, write out contents of /proc/uptime onto a temporary file
- ◆ At the beginning of first invoked application, just after the completion of init script, got the value of /proc/uptime; measured the time elapsed from init script to the invocation of the first application by comparing the obtained value and the saved value
- ◆ Time saving of **1.68 sec** observed for startup time it run through busybox 8 times (3.86sec -> 2.18sec)

- ◆ We verified Prelinker for MIPS
- ◆ We observed the improvement of 0.92sec -> 0.49 sec (C) and 1.88 sec -> 0.70 sec (C++) with 200 links
- ◆ Application startup time increases exponentially with number of links
- ◆ More improvement of application startup time was observed for C++ than for C

- ◆ MIPS Technologies
- ◆ CodeSourcery
- ◆ MIPS-SIG Members (CE-Linux Forum)

Thank You !!!