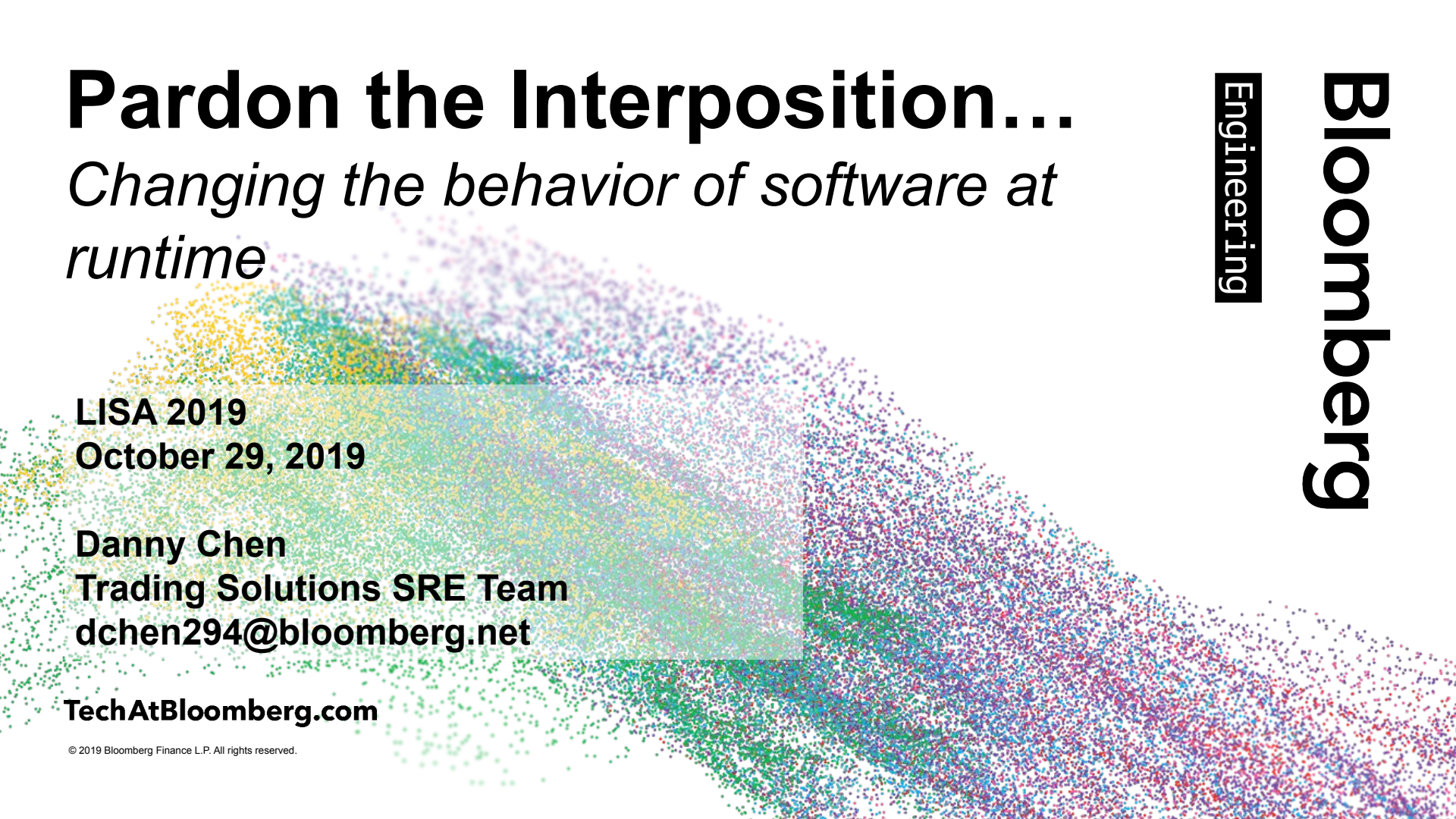
Pardon the Interposition…

Changing the behavior of software at runtime

Bloomberg

Engineering



LISA 2019
October 29, 2019

Danny Chen
Trading Solutions SRE Team
dchen294@bloomberg.net

TechAtBloomberg.com

My Background

- UNIX Performance Engineering
  - App and kernel performance
  - Capacity Planning
  - Enterprise Systems Monitoring
- Market data
- Messaging
- Distributed transaction management
- Reliability Engineering

USENIX

- Attended first USENIX conference (1982 or 1983)
- USENIX talk: User/kernel tracing package (1988)
- USENIX talk: Virtual memory performance improvements (1990)
- SRECon talk: Visibility into logging performance (2019)

Outline

- The need to modify software behavior
  - Sometimes an interposer is just what is needed
- Overview of interposers
- Examples of interposers
  - “Fixing” atoi
  - Debugging malloc
  - Modifying JIRA behavior
- Summary

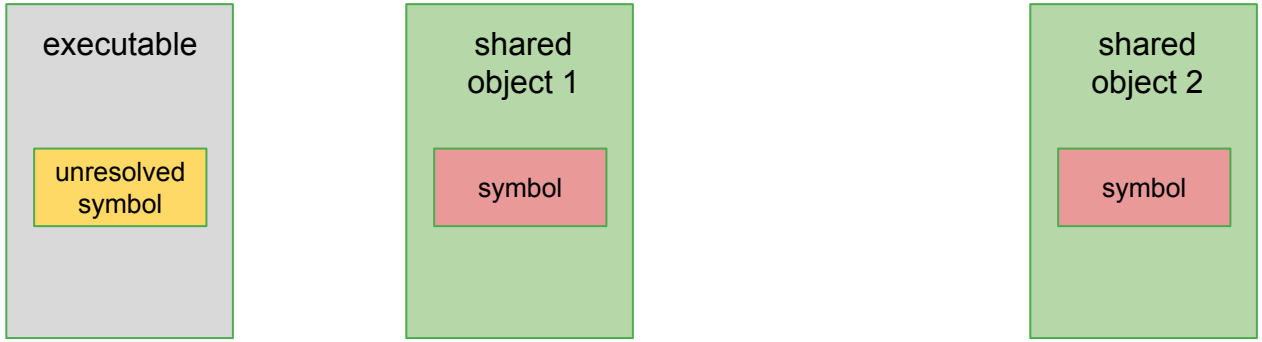
Software often needs customization

- Adapt to local “enterprise” conventions and mechanisms
  - Visibility
    - Metrics
    - Logging
  - Error handling and resiliency
- Bugs

- But we don’t have the source code…
- Or we have the source code but want to isolate the changes

Interposers (on UNIX)

- Leverages the late binding features of shared objects and the runtime linker
- Replacing one function for another at runtime



- We often still want to invoke the original function...

Example 1: atoi()

int I = atoi(“-1234”)	I gets value -1234	okay
int I = atoi(“123 skiddo”)	I gets value 123	maybe
int I = atoi(“0x377”)	I gets value 0	probably bad
int I = atoi(“not a number”)	I gets value 0	likely bad

Example 1: atoi()

```
#include <stdlib.h>
#include <ctype.h>

int atoi(const char *nptr)
{
      char *ptr = nptr;
      while (*ptr)
      {
            if (!isdigit(*ptr))
                  abort(); // fail fast
            ptr++;
      }
      return the_real_atoi(nptr); // how do we call the real one?
}
$ gcc -shared -fPIC -o myatoi.so atoi.c
$ LD_PRELOAD=./myatoi.so command args
```

TechAtBloomberg.com

Bloomberg

Engineering

Example 2: Debugging malloc

- Recording allocations and frees (for playback)
- Collect latency metrics

Bloomberg

Engineering

Example 2: Debugging malloc

```
#include <stdlib.h>
#include <dlfcn.h>

static void *(malloc_ptr)(size_t size);
static void (free_ptr)(void *ptr);

void
maalox_init()
{
     malloc_ptr = dlsym(RTLD_NEXT,
“malloc”);
     free_ptr = dlsym(RTLD_NEXT, “free”);
}
```

```
void *malloc(size_t size)
{
     ptr = (*malloc_ptr)(size);
     // record malloc, ptr, size;
     return ptr;
}

void free(void *ptr)
{
     // record free(ptr);
     (*free_ptr)(ptr);
}
```

```
$ gcc -fPIC -shared -o maalox.so maalox.c -ldl
$ LD_PRELOAD=./maalox.so command args
```

TechAtBloomberg.com

Bloomberg

Engineering

Example 3: JIRA attachments

- JIRA stores attachments (associated with tickets) on the filesystem
  - Atlassian recommends the attachment filesystem to be NFS
  - Our infrastructure division does not support NFS between datacenters
  - Storing attachments on local filesystem makes JIRA less resilient

- Like JIRA tickets, attachments are forever
  - Difficult to provision infinite filesystem store
  - We tried to limited attachment sizes

Ideal JIRA attachment behavior

- Filesystem as a cache for a cloud store
  - Write through?

- Attachments can be removed from the filesystem and re-fetched from the cloud as needed

- JIRA installation can be moved to a “fresh” host (e.g., failover)

Options for Changing JIRA behavior

- Adding functionality in the issue UI
  - New “cloud attach” button
- Leverage JIRA’s plugin architecture
- Modifying source code
- **Changing behavior of the “underlying system”**

Bloomberg

Engineering

When you can’t go high...then go low!

- Taking a step back
  - JIRA is a Java web app that runs in a Tomcat container
  - Both JIRA and Tomcat code runs in a JVM instance
  - The JVM itself is just a native app that uses underlying OS services

- Let’s take a look at how the JVM behaves for JIRA attachments...

Bloomberg

Engineering

Uploading an attachment...

```
…
open("/bb/jira-data/caches/tmp_attachments/temp13402685724047448 42",
...) = 205
fstat(205, ...}) = 0
…
write(205, …) = 1317
…
close(205) = 0
…
stat("/bb/jira-data/caches/tmp_attachments/temp13402685724047448 42", ...}) = 0
…
lstat("/bb/jira-data/caches/tmp_attachments/temp13402685724047448 42", ...}) = 0
…
stat("/bb/jira-data/caches/tmp_attachments/temp13402685724047448 42", ...}) = 0
…
rename("/bb/jira-data/caches/tmp_attachments/temp13402685724047448 42",
“/bb/jira-data/data/attachments/RDTSBIDI/10000/RDTSBIDI-778/47502") =
0
```

TechAtBloomberg.com

Bloomberg

Engineering

Changing rename() behavior...

In addition, if oldpath is under the tmp attachment directory, and newpath is under the attachment directory:

- If **actual rename()** succeeds, then also transfer the file to the cloud store

i.e., For rename() calls related to attachments, do the rename(), but also upload to the cloud store. For all other rename() calls, behave normally.

Downloading an attachment...

```
…
stat("/bb/jira-data/data/attachments/RDTSBIDI/10000/RDTSBIDI-778/47502",...}) = 0
…
open("/bb/jira-data/data/attachments/RDTSBIDI/10000/RDTSBIDI-778/47502", O_RDONLY) = 205
```

Changing stat() behavior...

In addition, if stat() fails, then try looking for the corresponding object in the cloud store

- If found in the cloud store, then download the object into the specified path and run stat() on the newly downloaded file

i.e., for stat() calls to attachments, if the file is not found, then check the cloud store and download it as needed. This effectively makes the filesystem a cache for the cloud store.

Many many other uses

- Instrumentation
- Changing “time”
- Adding delays to I/O
- Post processing after “close”
- Re-routing network connections
- Converting local tcp connections to UNIX domain sockets

…

TechAtBloomberg.com

Bloomberg

Engineering

Summary

- Late binding (and dynamic languages) allow for runtime modification of behavior
- Interposers can be very useful
  - Adding visibility (metrics and logging)
  - Augmenting functionality
  - No special permissions needed
  - Localized to single processes (LD_PRELOAD on launch)

Thanks!

We are hiring: bloomberg.com/careers

Bloomberg

Engineering

Questions?

TechAtBloomberg.com