System-level State Equality Detection for the Dynamic Verification of Legacy Distributed Applications

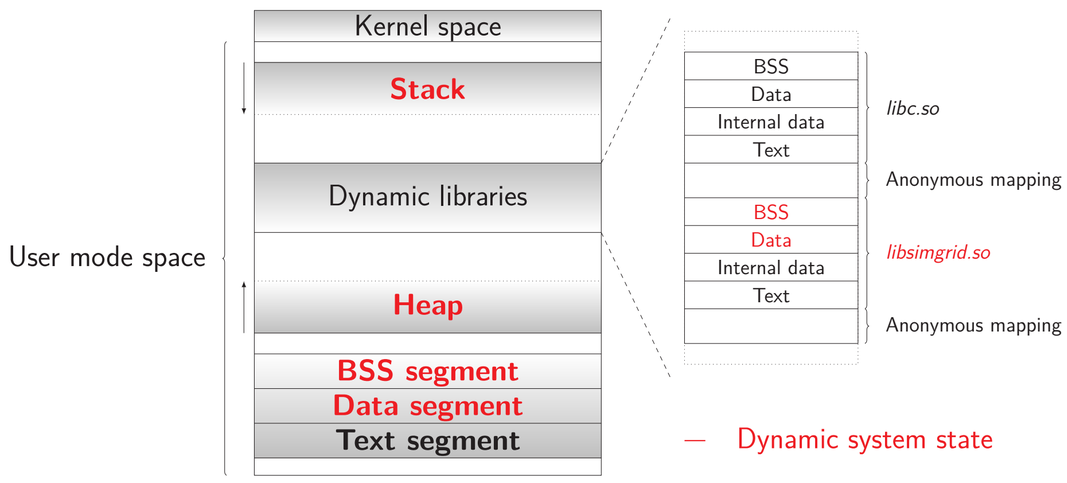
Marion Guthmuller and Martin Quinson
Université de Lorraine, France

Motivation: Study and verification of legacy distributed applications (C, C++)

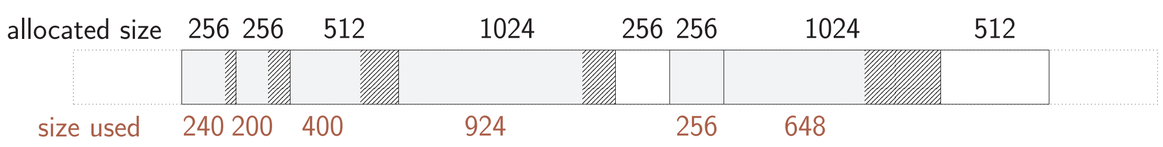
- Distributed systems are notoriously difficult to verify (non-deterministic, distributed memory, ...)
- Existing tools (Verisoft, MaceMC, JavaPathFinder, ISP, ...) are unsuitable/incomplete

System-level State Equality Detection

- **Use cases**:
  - **Stateful verification** (intermediate backtracking)
  - Dynamic verification of **liveness properties**
  - **Verification of (infinite-time) cyclic applications**
- **System state content**:

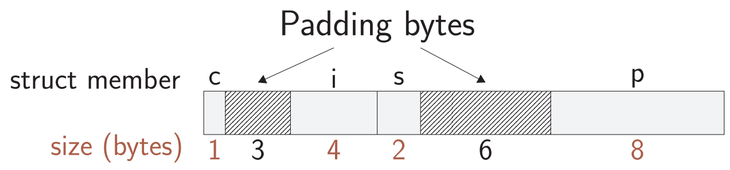


- **OS-level challenges**:
  - **Memory overprovisioning**:

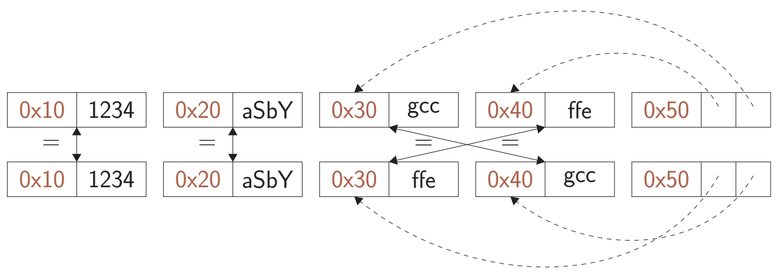


  - **Padding bytes**: Data structure alignment

```
struct foo {
    char c;
    int i;
    short s;
    void *p;
}
```



  - **Irrelevant differences**: system-level PID, fd, ...
  - **Syntactic differences / semantic equalities**:



- **Solutions**:

Issue	Heap solution	Stack solution
Overprovisioning	`memset` 0 (customized `mmalloc`)	Stack pointer detection
Padding bytes	`memset` 0 (customized `mmalloc`)	DWARF + libunwind
Irrelevant differences	Ignore explicit areas	DWARF + libunwind + ignore
Syntactic differences	Heuristic for semantic comparison	N/A (sequential access)

Experimentation and Results

- **Dynamic verification of a liveness property**
  - Buggy centralized mutual exclusion algorithm (infinite-time version)
  - Bug: one client never obtains the requested CS
  - Property: *Any process that requests the CS must get it*
  - Worst case considered (bug on the last process)
  - ≈ 100 LOCs – State snapshot size: ≈ 5MB

#P	# States	Time	Memory	Depth	Counter-example
3	64	1.9 s	2.3 GB	57	Found
5	1 112	14s	3.8 GB	1 009	Found
7	12 281	9m	45 GB	11 195	Found
9	> 100 000	> 1h	> 100 GB	-	Not found

- **Verification of some MPICH3 unit tests**
  - Looking for assertion failures, deadlocks and non-progressive cycles
  - Exhaustive exploration but no error found
  - ≈ 1300 LOCs (per test) – State snapshot size: ≈ 4MB

Application	#P	Stateless exploration		Stateful exploration		
		# States	Time	# States	Time	Memory
sendrecv2	2	> 55 millions	> 6h	936	13s	2GB
	5	-	-	2 284	43s	5.4GB
	10	-	-	3 882	2m	11GB
bcastzerotype	5	> 12 millions	> 1h	2 474	41s	3.1GB
	6	-	-	17 525	5m	19GB
coll4	4	> 100 millions	> 24h	29 973	20m	38GB
	5	-	-	> 150 000	> 4h	> 200GB
groupcreate	5	> 10 millions	> 1h30	2 217	38s	2.8GB
	7	-	-	71 280	24m	62GB
dup	4	> 57 millions	> 5h	4 827	1m20	6.5GB
	5	-	-	75 570	49m	87GB

Conclusion

- System-level state equality detection thanks to:
  - Debugging and profiling tools: DWARF and libunwind
  - Semantic heap comparison based on an heuristic
- Perspectives:
  - Reconcile the stateless and stateful approaches
  - Improve the memory performance

`http://simgrid.gforge.inria.fr/`