Security and the transactional programming model

Emmett Witchel

University of Texas at Austin


University
FEDERAL CREDIT UNION
VISA
Welcome
PLEASE SWIPE CARD
FOR SERVICE
POR FAVOR PASE SU
TARJETA DONDE SE
INDIQUE PARA SERVICIO
Card

Transactional Memory: Architectural Support for Lock-Free Data Structures

Maurice Herlihy
Digital Equipment Corporation
Cambridge Research Laboratory
Cambridge, MA 02139
herlihy@crl.dec.com

J. Eliot B. Moss
Dept. of Computer Science
University of Massachusettes
Amherst, MA 01003
moss@cs.umass.edu

Transactional Memory: Architectural Support for Lock-Free Data Structures

Maurice Herlihy
Digital Equipment Corporation
Cambridge Research Laboratory
Cambridge, MA 02139
herlihy@crl.dec.com

J. Eliot B. Moss
Dept. of Computer Science
University of Massachusettes
Amherst, MA 01003
moss@cs.umass.edu

Transactional Memory: Architectural Support for Lock-Free Data Structures

Maurice Herlihy
Digital Equipment Corporation
Cambridge Research Laboratory
Cambridge, MA 02139
herlihy@crl.dec.com

J. Eliot B. Moss
Dept. of Computer Science
University of Massachusettes
Amherst, MA 01003
moss@cs.umass.edu

Transaction Interface

- Programmer transactions are consistent
  - Represents some coherent unit of work
  - Programmer associates metadata with code region
- Big opportunity to leverage programmer effort

Transactions for Security

- Transactions naturally benefit security
  - Time-of-check-to-time-of-use (TOCTTOU) bugs

```
xbegin;
if(access(argv[1], R_OK) != 0) exit(1);
int fd = open(argv[1], O_RDONLY);
xend;
```

- Requires strong isolation for system calls
  - And a transactional (or journaling) file system

Transaction Capabilities

- Transactions can take memory space capabilities
- Capabilities are OS created 64-bit hash values
  - Code regions are principles
- Capability associated with transaction, not pointer
  - Avoids many traditional capability problems

```
in_cap =
  cap_create(TX, stack,
    RW, heap);
get_input(in_cap);
```

```
get_input(void*input_cap){
  xbegin, trans_cap;
  char buf[128];
  gets(buf);
  xend;
}
```

Let Us Agree (Vote?)

- Transactionalize vs. Transactify
- Strong isolation vs. Strong atomicity
  - No weak isolation?