2025

Address Binding in Cloud Financial Trading Platform: ITE240_Operating System (3)

LECTURER: AJARN THINZAR AUNG WIN

GROUP 5: ZWE PYI PHYO (2406120005)

Contents

Part 1: Address Binding Analysis & Mapping

1. Address Binding Plan

Process	Recommended Binding Type	Explanation
Order Execution Engine	Run-Time	This process requires trade to happen in real time. Therefore, it must be able to dynamically adapt to order changes. By using run-time binding, it will allow OS to adjust memory references after migration only. This is essential since it ensure no trade duplication or loss occur during the migration. Additionally, run-time binding provides flexibility and precision needed for security and accuracy in a live trading environment.
Market Data Aggregator	Run-Time	Since the global exchange is continuously changing, binding a Run-time is also ideal for this process. Since it handles constant high-throughput from exchanges, run-time binding will be able to support on-the-fly allocation without having to restart, preserve performance or data integrity.
Risk Analyzer	Load-Time	Risky analyzer is a large and memory intensive process but is less sensitive to real-time accuracy. Load-time binding will allow efficient memory setup while permitting migration without having to fully re-compiled.
Compliance Monitor	Compile-Time	Due to legal regulations, the logic must not be altered during or after migration. Therefore, compile time ensures fixed addresses and control of the process.
UI Dashboard	Load-Time	Unlike other processes, UI can tolerate being restarted and does not require dynamic relocation. By using load-time binding, it will simplify the deployment while ensuring responsiveness when restarted also.

2. Scenario

Market Data Aggregator streams real-time data from global exchanges and handles high throughput of feeds. Using compile-time or load-time address binding can cause significant issue when migrating one server to another, especially when the layouts are not the same. Since compile-time and load-time binding resolve physical memory addresses early in the process, if the physical memory layout in Frankfurt is not same with Singapore, these fixed addresses may no longer be

valid. This mismatch between memory address can cause risk such as data corruption, system crash or even preventing data aggregator to function properly.

Moreover, lack of dynamic rebinding poses additional security and performance risks to the system. For instance, the process could access unauthorized memory regions with a mismatched memory address. This could lead to potential data leaks or interfering with other critical applications in the system. Additionally, faulty addresses translation can cause a loss of data continuity that can cause real-time feeds to become outdated or incomplete. In an industry where every split second is crucial, having an unreliable data can cause serious of issues to the business. Therefore, to avoid these risks, run-time address binding is a preferable option for this type of process. It allows OS to reassign memory dynamically during migration and ensure both performance and security.

Part 2: Secure Dynamic Relocation & Memory Protection

1. Mapping Diagram

Before Migration (Singapore)	
Logical Address	Physical Address (X)
0x0001	0xA101
0x0002	0xA102
0x0003	0xA103

After Migration (Frankfurt)	
Logical Address	Physical Address (Y)
0x0001	0xB501
0x0002	0xB502
0x0003	0xB503

The logical address space remains the same, only the mapping to physical address is different. This allows processes to resume seamlessly even in the different layout.

2. Explanation

Using a fixed addresses can cause serious issue such as corrupted trade queues or crash the system. Since the Order Execution engines maintains sensitive and active queues of trades, having a fixed address can cause memory pointers to refer to an invalid or an occupied region. This can ultimately lead to trade duplication or loss in the system. Likewise, system can come to a halt if the exact address space is not available at the destination. Additionally, if confidential data like order books are redirected to unsecured or unauthorized memory region, hackers could potentially intercept or manipulate the data.