Модификация инструментария AWT в графической подсистеме X11 для удовлетворения требованиям проекта OpenJDK CRaC

Кузнецов Илья Александрович
Группа 19.Б11-ММ

Научный руководитель:
ассистент Антон Павлович Козлов

Санкт-Петербургский Государственный Университет

Кафедра Системного Программирования
Программная инженерия

Java и особенности программ, OpenJDK CRaC

- OpenJDK – эталонная реализация Java
- Проблемы при использовании: медленный старт, долгий разогрев
- CRaC – развивающийся проект в OpenJDK:
  - сохранение и восстановление полного образа состояния JVM
  - позволяет выполнять произвольный код перед сохранением (Checkpoint) и после восстановления (Restore) с помощью интерфейса ресурса (Resource)
  - требует отсутствия внешних связей: сетевые соединения, межпроцессные взаимодействия…
- Графические приложения подключались к графической подсистеме ОС → не поддерживались

Цели и задачи

Целью данной работы является добавление начальной поддержки графических приложений AWT/X11 в проект OpenJDK CRaC

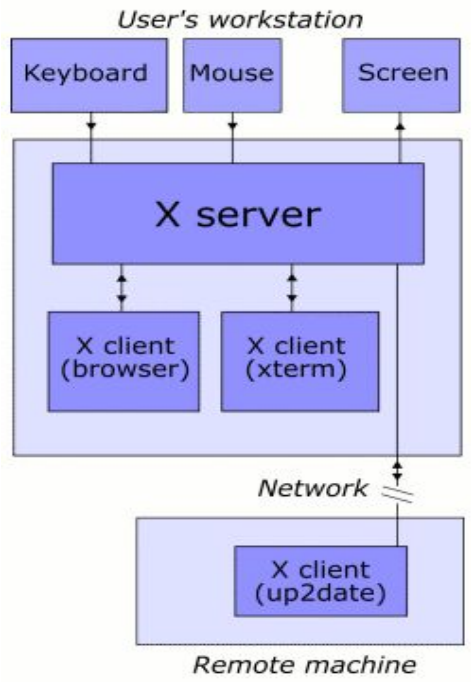
Задачи:

- Провести обзор предметной области:
  - графической подсистемы UNIX и протокола X11
  - реализации AWT для X11 в OpenJDK

- Реализовать начальную поддержку графических приложений AWT/X11, а именно: сохранение образа JVM, исполняющего графическое приложение, и его восстановление с возможностью создания графического интерфейса приложения

X Window System, протокол X11, библиотека Xlib

- X Window System – стандартная графическая подсистема UNIX
- X состоит из клиентов, сервера и графического (сетевого) протокола X11

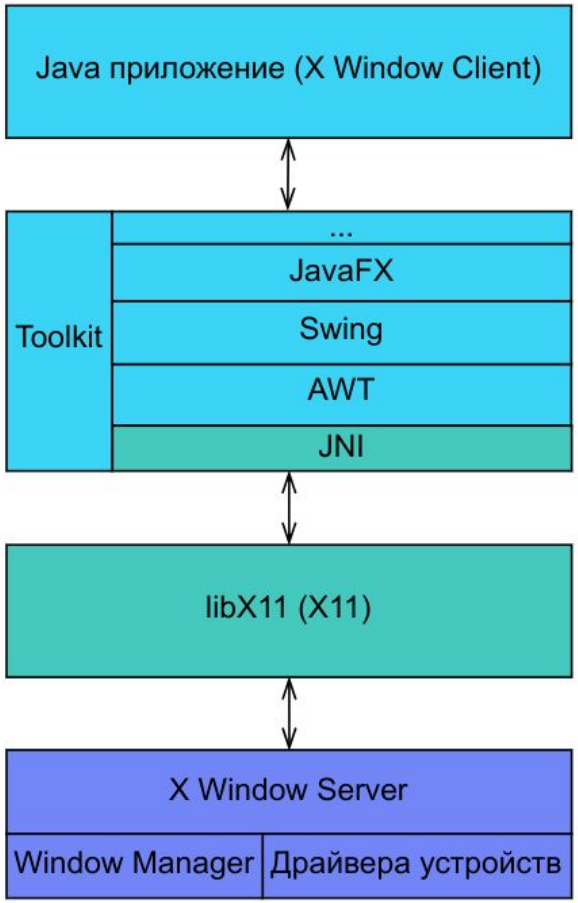


- Библиотека Xlib – низкоуровневая реализация клиентской части X11
- Window Manager - специальный клиент X11
- Клиент и сервер общаются по X11 с использованием запросно-событийного принципа

JNI, инструментарий AWT

- Со стороны Java библиотеку Xlib через Java Native Interface реализует инструментарий AWT

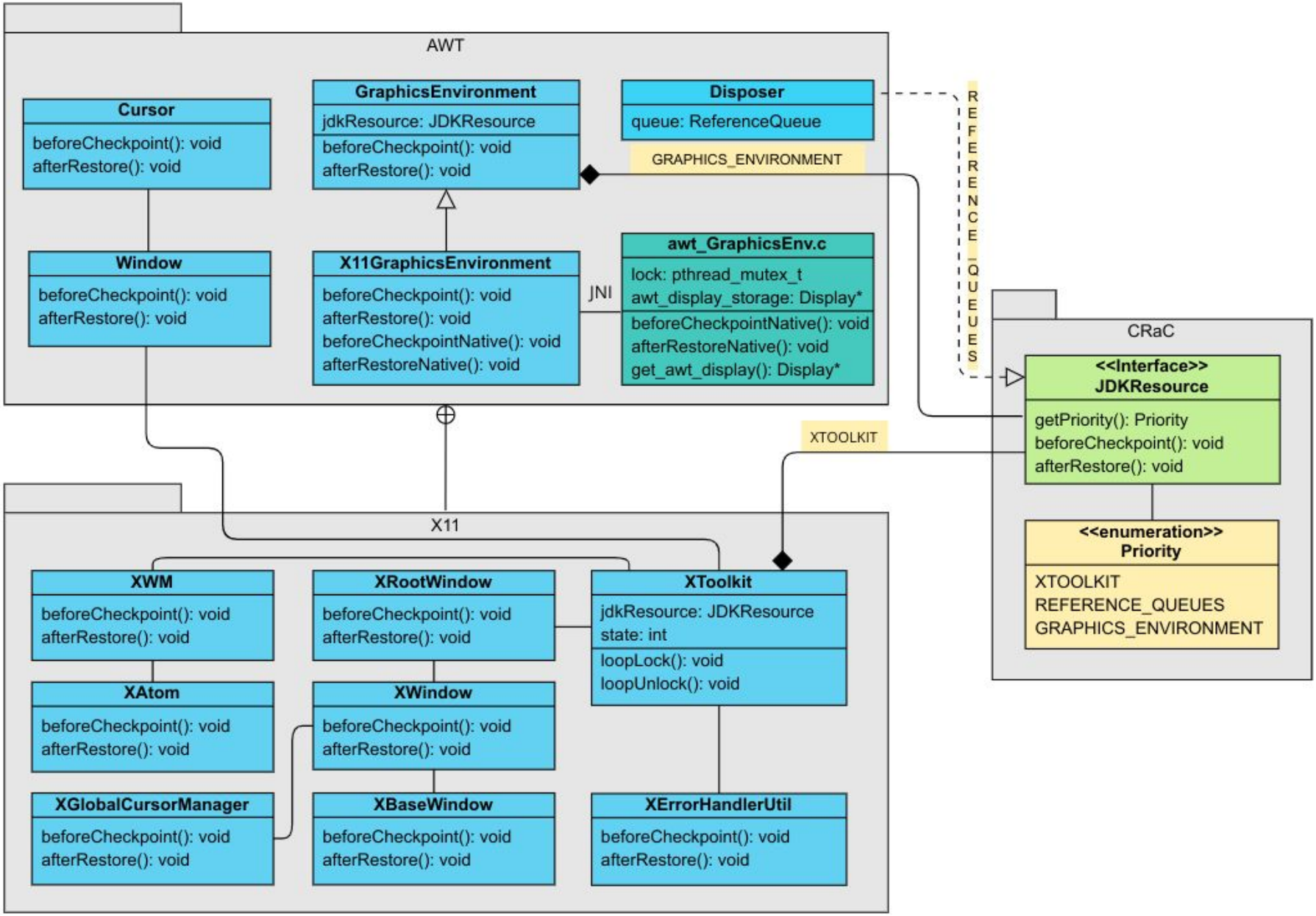
- AWT предоставляет интерфейс к X11, набор компонентов, нативных peer и функции графического ядра, оборачивая Xlib
- От AWT зависят более продвинутые инструментарии Swing и JavaFX
- Модификация AWT/X11 – это важный шаг для поддержки графических приложений в OpenJDK CRaC



Рабочее окружение

- Отладочная сборка OpenJDK 17 с CRaC и Xlib
- Анализаторы протокола X11: Xscope и Xtrace
- Скрипты
- Тестовые приложения с использованием AWT и Swing

Архитектура решения



Реинициализация: GraphicsEnvironment, X11GraphicsEnvironment

- Изначально соединение по X11 не закрывалось совсем
- Реинициализирован X11GraphicsEnvironment, инкапсулирующий нативные структуры для соединения с графической подсистемой ОС
- Сразу создан общий интерфейс для реинициализации на других ОС

Реинициализация: XErrorHandlerUtil

- За дисплеем закрепляется обработчик ошибок, управляемый XErrorHandlerUtil
- Его тоже необходимо реинициализировать, чтобы связать с новым дисплеем и очистить сохраненные ошибки

Реинициализация: Window, XWindow, XBaseWindow, XRootWindow

- Компоненты AWT и окна X11 продолжали использоваться на новом соединении, а нативные peer препятствовали остановке JVM
- Чтобы это исправить, было реализовано закрытие всех этих объектов и сброс статистических данных до Checkpoint
- XRootWindow реинициализируется отдельно, так как все остальные окна зависят от него

Реинициализация: Cursor, XGlobalCursorManager

- Объекты Cursor тоже кешируются приложением, ими управляет XGlobalCursorManager
- Для Cursor реализована очистка всех закешированных данных
- Для XGlobalCursorManager – реинициализация объекта

Реинициализация: WM, XAtom

- Окна приложений управляются с помощью системного WM, общение с которым происходит с помощью XAtom; некоторые из них кешируются
- Все XAtom было необходимо очистить до Checkpoint и восстановить необходимые после Restore; другие будут восстановлены в ходе работы с приложением
- Для WM была выполнена логическая деинициализация и очистка до Checkpoint и стандартная инициализация после Restore

Реинициализация: XToolkit

- XToolkit - инструментарий X11, лежащий в основе AWT
- Он управляет реинициализацией всех рассмотренных частей, а также:
  - блокирует свой Event loop на время CheckpointRestore
  - контролирует очистку всех окон и нативных peer, удаляет закешированные данные и статистику перед Checkpoint
  - реинициализирует устройства ввода-вывода и другие нативные объекты во время CheckpointRestore

Реинициализация: Disposer

- На переоткрытом соединении по X11 после Restore иногда отправлялись старые запросы на очистку
- AWT Disposer не был синхронизирован с Checkpoint и потому не успевал закончить работу
- Проблема была решена в общем случае через ожидание получения ссылок от GC и синхронизированной очистки нужных ReferenceQueue до Checkpoint

Особенности модификации AWT/X11

- Протокол X11 требует вначале договориться о более легковесных примитивах общения
- AWT/X11 использует кеширование
- Некоторые объекты в AWT/X11 были предназначены только для единственной инициализации
- Большие иерархии наследования, нетривиальные зависимости в коде и многопоточность в AWT/X11

Результаты

Для достижения **цели** работы были выполнены следующие **задачи:**

- Проведён обзор предметной области (X11 и AWT)
- Подготовлено рабочее окружение (OpenJDK, CRaC, Xlib, Xscope, Xtrace, скрипты, приложения)
- Реализована начальная поддержка графических приложений AWT/X11 в OpenJDK CRaC:
    - реинициализация образа JVM и рассмотренных частей AWT/X11 (выполнено)
    - переработка и минимизация кода (выполнено)
- Интеграция изменений в OpenJDK CRaC (в процессе из-за OCA, блокирующего PR)