

The Multi-seat Configuration for Wayland

Uea-angkun “Diskette” Khunpradith¹

¹The dailitation.xyz team; E-mail: diskette@dailitation.xyz

First published: 2025-09-12

compiled on 2025-11-17

Abstract

Multi-seat or a multi-terminal configuration is traditionally used over 40 years ago with networked main-frame computers. It is out of fashion today, but it has advantages in economical and ease-of-administration situations, as it requires less hardware and less administration. The project uses Linux on a low-power device to simulate the worst case scenario.

Keywords: Multi-seat, Linux, Shared Computing, X11, Wayland

1 Introduction

More than over 40 years or so, computers were tremendously expensive and terribly slow compared to today. Despite the fact that it was slow and expensive, it was considered to be the ultimate technology there is, and they made the most out of it by sharing resources simultaneously between each users of that particular computer with a time-sharing operating system¹. Now, we have computers for ourselves with over three hundred-thousand times faster than the computer we had back then² without the need to share computing resources as computers has gotten vastly affordable in the last decades.

Despite that fact, sharing computing resources via multi-seat configuration is still practical and has several advantages compared to a single-seat configuration. For example, economical advantages of having one powerful computer connected to multiple input and output devices to create a multi-seat system is less expensive than buying multiple computers, it also saves electricity, thus making it also environmentally friendly. Administration advantage, as the administrator would have to administrates less computers overall.

2 Methodology

The target user of multi-seat system are system administrators of a library, a school, or a small workplace in which requires affordable computer system. Traditionally, that is achieved by using one computer and multiple graphics adapters via the PCI-E or USB interface per one seat (user). However, while it is true that most graphics adapters are inherently cheaper than the

computer itself. But due to time constraint and cost-cutting measures, this research used a single graphics adapter built-in to the computer hardware. Keep in mind that some graphics card is able to do hardware level virtualization, effectively providing multiple virtual interfaces for configurations similar to this, whether that would be Intel GVT-g, NVIDIA vGPU, or generic SR-IOV devices. However, the researcher do not have the sufficient hardware to be able to demonstrate the said latter and would like to showcase a universal method that works with every hardware with some limitations.

2.1 Comparison Metrics

To measure the efficiency, a comparison was used between two scenarios:

1. **A single-seat computer** with exceeding the hardware requirement as a control group.
2. **A multi-seat computer** with the same hardware or the same machine, but with a multi-seat configuration to make comparisons.

A comparison was made by comparing use cases based on a set of tasks that a typical user would do. Currently, this article is just a manual-style documents of how it would have worked. Comparison information between systems that do utilize single-seat and multi-seat system would be updated in a revision and this revision would be obsoleted. However, the following paragraphs The CPU time and power consumption (wattage) was used as a comparison metric between single-seat and multi-seat system. CPU time and power consumption are captured every one second interval parallel to each other for 30 seconds from the

¹Time-sharing operating system is a type of operating system that is able to do concurrent sharing of computing resources by giving each task a small processing time.

²Motorola 68000 is able to do 1.4 MIPS at 8 MHz, while Intel Core i5-11600K (6-core) 346,350 MIPS at 4.92 GHz. See more at reference [2]

start of the benchmark. To calculate the efficiency we will use this following formula:

$$\frac{\text{CPU time}}{\text{Power consumption}} = \text{Performance per watt}$$

2.2 Suggested Hardware Requirements

The researcher have picked a set of criteria based on their personal self-observation with the specification of government supported school computers inside the computer laboratories. A multi-seat configuration would be most effective in an environment similar to the latter said. Currently, dailitation.xyz team lacks the funding to be able to find sufficient hardware.

- **The computer hardware** needs to be at least the worst case scenario of what the school computer laboratories administrator would have to deal with, or more or less the same thing. With an addition to specific specification to this experiment to simplify the comparison made.

ThinkPad T410 was picked. It's—as of writing this—14 years old reliable machine. It's the perfect device to simulate the worst case scenario. Any laptop would have a built-in keyboard, pointing device and a screen, perfect for the first seat. But this specific laptop, the ThinkPad T410, has a configuration of Intel Core i5-520M which is a homogeneous multi-core processor, 8 GiB of RAM and a single Intel-based graphics adapter.

HDMI capture card was used. This is not required for multi-seat and was quite an irregular choice of hardware. It was used instead of a screen monitor, because the all of the screen monitor that the researcher had access to was occupied at the time of the research. The capture card allows screen multiplexing which provides a “virtual” monitor, similar to that of a real monitor.

- **An operating system** needs to be free³ Unix-like operating system with elogind.

Alpine Linux was used. It's a simple and small Linux distribution, designed for embedded devices and/or general purpose.

- **A graphical user-interface**, either an X11 window manager or Wayland compositor, needs to have great usability, simple design with simple interworkings and lots of customization.

Sway, the Wayland Compositor, was picked as a background compositor. It's a lightweight compositor meant to be compatible with i3 window manager; thus is also customizable.

Wayfire, the Wayland Compositor, was picked. It's a simple tiling Wayland Compositor without sacrificing user usability and with simple interworkings.

2.3 Benchmark

These following set of tasks would've been chosen and performed as a benchmark for the next revision.

- **Web browser tasks** Most internet websites use the JavaScript programming language, a high-level interpreted language to provide dynamic content to the world wide web. It came with a cost as it's a high-level interpreted language, it's quite CPU time-intensive.
 - Document authoring: use two instances of Google Docs concurrently.
 - Reading: access two instances of Wikipedia articles concurrently.
 - Social media: access an instance of Twitter.
 - Instant messaging: access an instance of FaceBook messenger web-page.
 - Video multimedia: play an instance of “WING IT! - Blender Open Movie” on YouTube.
- **Native software tasks** Software compiled native to the machine ABI that aren't using JavaScript in the backend are most likely using a library as an intermediary to the system calls. These native libraries are fast compared to using a high-level interpreted language like JavaScript.
 - Document authoring: use three instances of OpenOffice Document files using LibreOffice Writer concurrently.
 - Reading: access three instances of PDF documents using Okular concurrently.
 - Social media: access an instance of Tokodon.
 - Instant messaging: access an instance of NeoChat.
 - Video multimedia: play an instance of “WING IT! - Blender Open Movie” locally using VLC.

3 Linux Graphics Architecture

3.1 Display communication protocols

Linux—as of writing this research—has two main display server communications protocols, **X11** (via X.Org) and **Wayland** protocol.

X11 or **X Window System** is an extensible architecture-independent client-server window system. A display server called **X server** runs in the background and is a man-in-the-middle between the computer input and output hardware (via evdev and DDX DRI modesetting drivers) and the X clients. X clients are applications that are displayed on an X server, it can be your typical graphic application like an internet browser, but it could also be a window manager. X

³Free as in freedom, without any restrictions; licensed under a free license.

window manager is an X client that controls the appearance and behavior of windows—how graphical applications are drawn—the look and feel of the graphical user-interface “frames”. The X compositor is an X11 extension that composite special inanimate or animated 2D or 3D effects into an off-screen buffer, and automatically merge into the graphical user-interface. When a user input is sent from the hardware to the kernel, the kernel sends it to X server through the evdev input driver, then the X server pass the input to the intended X client, whether that would be the window manager itself, or other X client. When the X client received the input, it often will change the user-interface as a response to the input, such as selecting a text, X client will send a rendering request back to the X server to render at the display server itself.

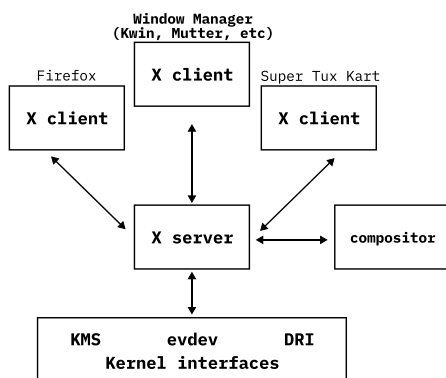


Figure 1: Interworking of X11

The limitations of X11 is simply to reliant on the X server itself—as of today, we now have a potential⁴ successor for X11, Wayland.

Wayland is a communication protocol specification, destined to be the successor to X11. Unlike X11, Wayland is just a specification and requires no man-in-the-middle server software. Instead, an implementation of the communication protocol called, Wayland compositor, is its own display server, it does both window management and composition. The Wayland compositor has a direct control of kernel mod- esetting and evdev without a middleman. When a user input is sent from the hardware to the kernel and to the Wayland compositor, the compositor decides which client that should receive the event by looking at its scenegraph, as the scenegraph is directly correlated to what’s being displayed even if the window frame are transformed, rotated, or even wobbly—since the display server, window manager, and the compositor is in an all-in-one package. The rendering happens in the client itself, although the client still needs to send the request to the compositor to indicate which graphical region was updated.[1]

⁴Although Wayland is the supposed successor to X11, there are still reasons to prefer X11 over Wayland.

⁵Do not confuse screen and monitor with display

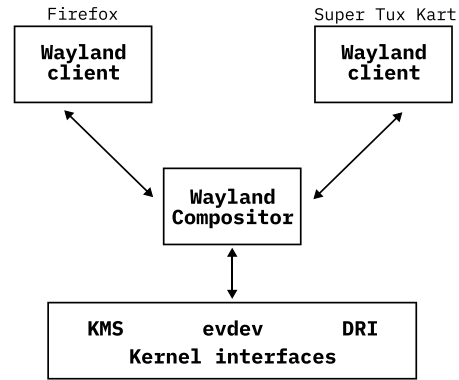


Figure 2: Interworking of Wayland

3.2 The DISPLAY, Screen and Input Devices

Since both X11 and Wayland is a **display server**, it works similarly to that of a normal typical servers on the internet. A client connects to the server via socket—but display server usually connects to the socket from the DISPLAY variable. The DISPLAY variable is used to refer to a collection of monitors, keyboard and mouse of that particular “display” and has this following syntax:

`hostname:display.screen`

The `hostname` is rather self-explanatory.

The `display` is an arbitrary numeric value assigned to each display.

The `screen` is used in case if that particular display has 2 or more screens or monitors⁵.

The following figure (figure 3) is an example of how those correlate: in this imaginary room, there is one computer named “haus” and two seats (displays). One of the seats have two monitors (screen0 & screen1), a keyboard (keyboard0) and a mouse (mouse0). The other seat only has a single monitor (screen2), also a keyboard (keyboard1) and a mouse (mouse1).

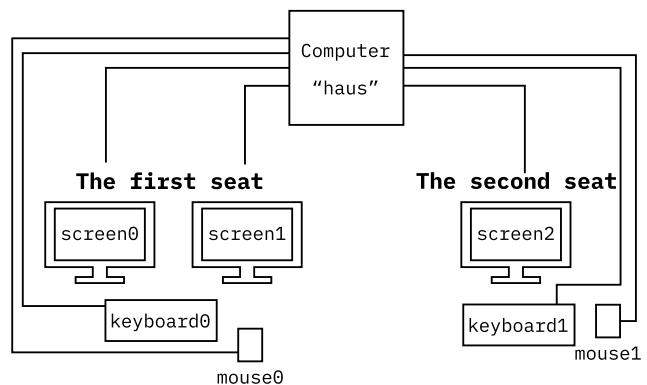


Figure 3: An imaginary diagram based on the example

The system would’ve been configured to have the first seat, seat0 to use DISPLAY number 0 which got assigned with screen0 and screen1, keyboard0 and

mouse0. The second seat uses DISPLAY number 1 which got assigned with screen2, keyboard1 and mouse1—see figure 4 for visualization.

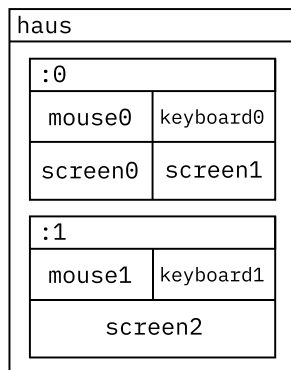


Figure 4: A visualization of DISPLAY seats

3.3 Nested Display Server

The X display server and the Wayland compositor are able to have a nested session inside of the display server environment itself.

Although single multiseat system does not necessarily require a nested display server, but for the sake of administration, the researcher decided to use one.

4 Configuration Setup

The researcher has configured a basic installation of unto the ThinkPad T410 with Sway and Wayfire compositor. Sway was used as an underlying compositor

for Wayfire to nest inside. A start script was launched to launch Wayfire on each nested DISPLAY and moves the corresponding window to their assigned monitor.

4.1 Limitations

Software-level virtualization do not meet the standards of which one could use it as their daily workstation. The performance overhead caused by multi-seat system could be mitigated by a more powerful hardware-level multiplexing graphics adapter.

5 Results

The multi-seat configuration is able to simultaneously support two users using one graphics card. Although, with limitations as the experiment was using one graphics card without hardware-level virtualization.

6 Acknowledgment

This project was conceived by the initiator of dailiation.xyz team, Uea-angkun “Diskette” Khunpradith due to their curiosity of the interworkings of Linux graphics architecture. Also a slight pettiness towards research documents that are closed behind a paywall. No large language model (LLM) or generative artificial intelligence (AI) was used during the entirety of this research article, but rather, pure anthropoid intelligence.

Special thanks to Miss Lori Michelle Hershberger, the proofreader of this article.

References

- [1] The wayland.freedesktop.org contributors. *Chapter 3. Wayland Architecture*. freedesktop.org, 2025. URL: <https://web.archive.org/web/20250802143517/https://wayland.freedesktop.org/docs/html/ch03.html>.
- [2] Ludovic Drolez. *Lud’s Retro Computing Ressources*. 2025. URL: <https://web.archive.org/web/20250515161718/https://drolez.com/retro/>.

License

This article is dual-licensed under Creative Commons Attribution ShareAlike 4.0 International (CC BY-SA 4.0) and GNU Free Document License 1.3 (GFDL-1.3).

You may copy, modify, and redistribute this file and its source codes, if and only if you redistribute the copy under the same license(s).