Visualize Me

Explore the quickly updated world

Zining Wang @ Northeastern University

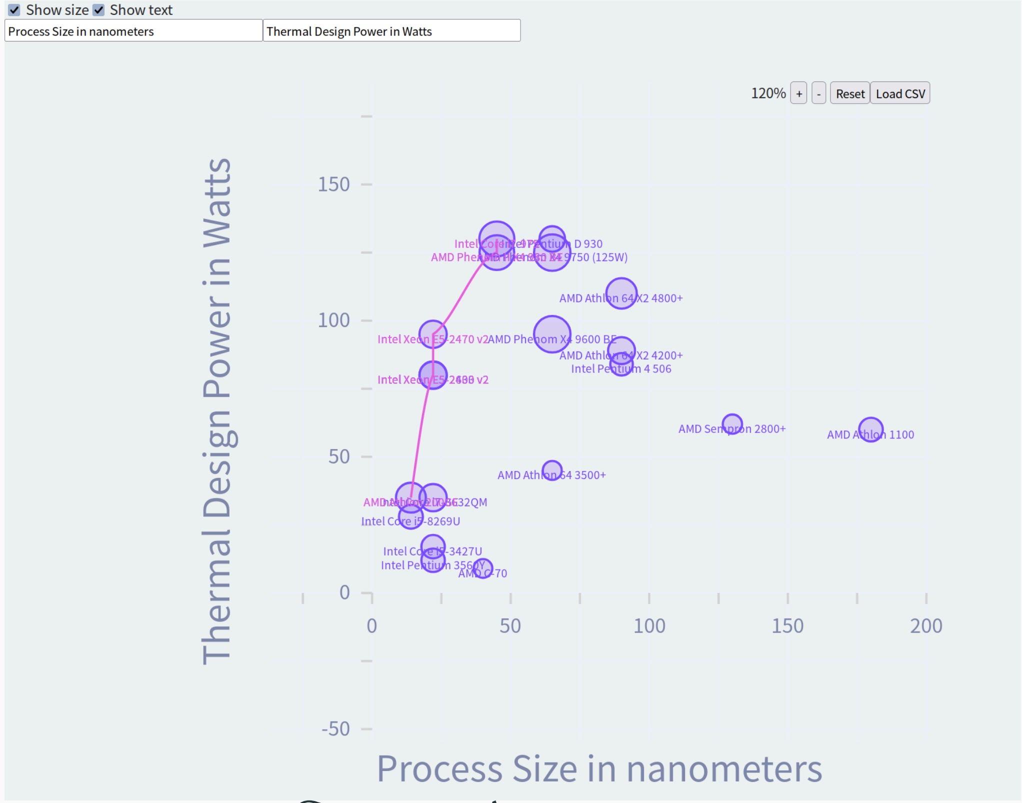
2025-04-04

EECE5642 Project Proposal

Outline

1. Project Information

- Team Members & contributions:
  - Zining Wang: Idea, Design, Implementation and Presentation
- Used Language: elm-lang
- Source Code: https://github.com/wznmickey/visualizeMe
- Online Demo: https://wznmickey.github.io/visualizeMe/
- Data Source:
  - https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data
  - https://llm-stats.com/



☑ Show size ☑ Show text

Process Size in nanometers

Thermal Design Power in Watts

Current Appearance

1.2 Motivation

In today's fast-paced technological world, advancements in hardware and artificial intelligence (AI) are occurring at breakneck speed. This creates a dilemma for consumers and professionals who seek to stay up-to-date with the best performing systems without breaking the bank. While tools for benchmarking hardware, particularly CPUs and GPUs, have become increasingly sophisticated, the problem lies in the fact that top-tier performance is not necessarily the most cost-effective solution for everyone.

When selecting hardware, it's important to recognize that performance should not be the only consideration. Several other factors play a crucial role in choosing the right setup for a given application.

1.2 Motivation

- Power Consumption: While higher performance often correlates with higher power requirements, a balance must be struck to avoid excessive energy costs or hardware overheating.
- Price: The most powerful hardware often comes with a premium price tag, but for many users, the best performance is not required for their daily tasks. For example, a developer or gamer may need just enough GPU power to run programs efficiently, but not necessarily the highest-end models available.

1.2 Motivation

- Specific Use Case: Different use cases demand different hardware characteristics.
  - Single-core performance might be essential for applications that rely heavily on sequential processing (e.g., certain older games or single-threaded applications).
  - Multi-core performance is critical for tasks like rendering, scientific computing, and modern gaming where parallel processing is leveraged.
  - For GPUs, some users prioritize high encoding and decoding ability for video production, while others might prefer high memory bandwidth for tasks such as gaming, machine learning, or 3D rendering.

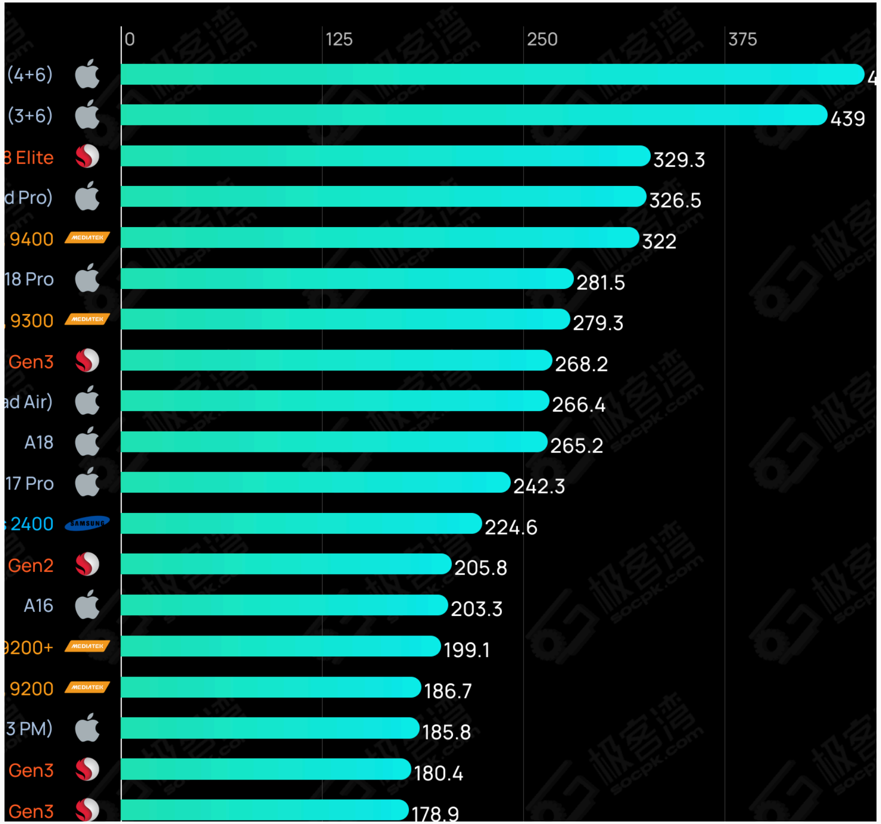
1.2 Motivation

In addition to hardware considerations, there are also other contexts in which performance must be evaluated. For instance, in the realm of Large Language Models (LLMs) and other AI models, the user’s needs must be balanced across several dimensions. As LLMs continue to evolve, model size, accuracy, and processing speed are important factors to weigh. Some users may prioritize faster inference times, while others may be more concerned with the accuracy of the results. Additionally, some may be restricted by computational resources, thus requiring smaller models that can still offer competitive performance.

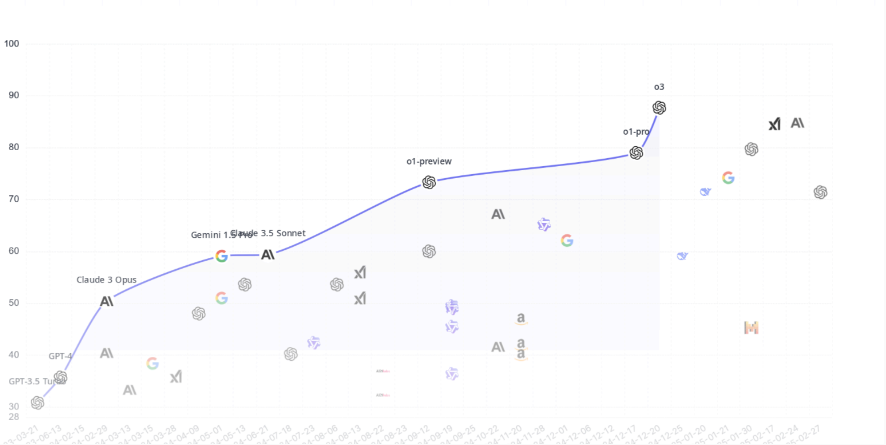
1.3 Current Situation

Hard to filter the desired parameters.

https://socpk.com/allperf/

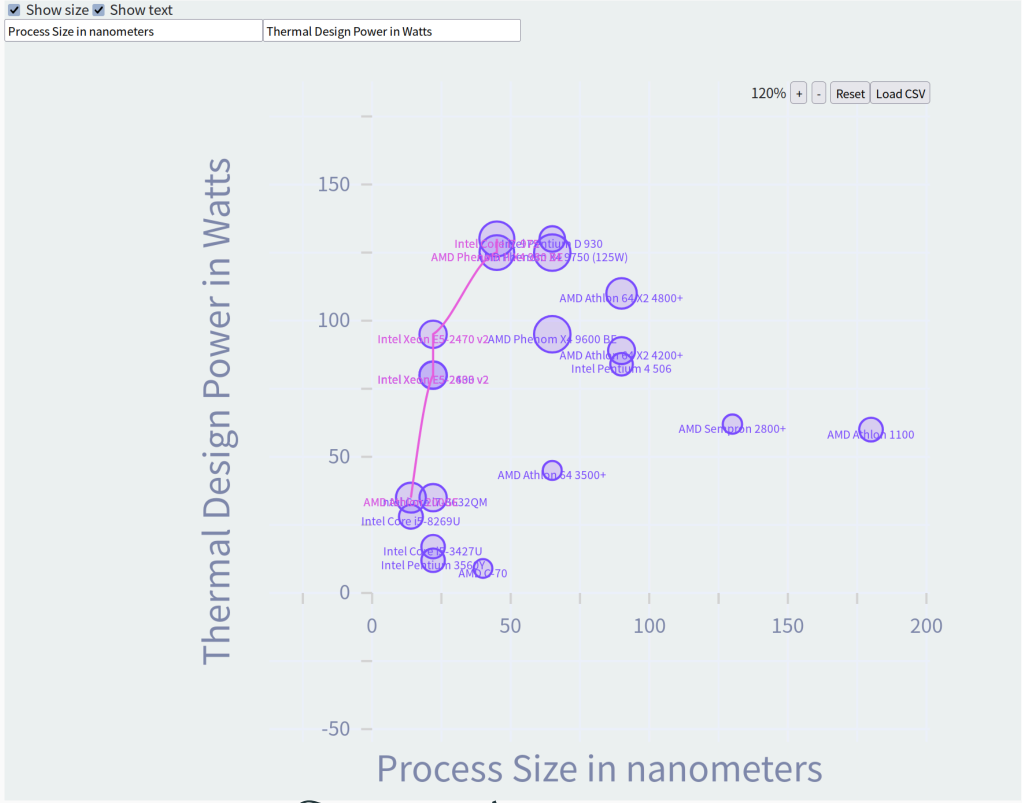


https://llm-stats.com/



2. Develop Process

To make the tool acessible to more people, I want to develop a web-based tool and I use elm-lang which is a functional programming language that could be compiled into a static HTML file with JavaScript containing SVG images which could be easily hosted on GitHub Pages service.



☑ Show size ☑ Show text

Process Size in nanometers | Thermal Design Power in Watts

Current Appearance

2.2 Live Example

https://wznmickey.github.io/visualizeMe/

1. Zoom in and out by pressing the button.
2. Reset the zoom by pressing the button.
3. Zoom in and out by the wheel. (Thanks to the suggestion that using the wheel to zoom in my proposal)
4. Move the plot by clicking the new center.
5. Drag the plot to move it. (Thanks to the suggestion that removing the + line at the center of the screen in my proposal)
6. Switch on the text, size and the pareto line.
7. Upload the data and visualize it.
8. Change the x-axis, y-axis text.

2.3 Challenge

1. Lack of documentation and packages for elm-lang. e.g. csv parsing,key binding
2. Lack of contributor. I work alone in this project.
3. Lack of experience in web frontend development. I am not familiar with tools like CSS in organizing different elements in a page.

2.4 Current bugs & Undeveloped Part

There are some bugs in the current version:

- The text would not disappear when it is out of the screen.
- The size of each point could not resize according to the relative size of the data.
- Text may of x-axis and y-axis may be overlapped by the data point features.
- The layer of the element to capture wheel is lower than that of the element to show the data which causes the wheel event not captured in the center.

To-do:

- Filter different data columns in the csv file.
- When the mouse is moved onto each point, the text of the point should be shown.
- Better design of the layout of the page.

Developed For

computer DIY players

LLM local deployment users & researchers

People who have many options to choose from

https://wznmickey.github.io/visualizeMe/

3. Appendix

3.1 project Source Code

Main.elm

```elm
module Main exposing (..)

import Browser
import Zoom exposing (..)


main =
    Browser.element
        { init = init
        , view = view
        , update = update
        , subscriptions = \_ -> Sub.none
        }

```

Zoom.elm

```elm
module Zoom exposing (..)

import Chart as C
import Chart.Attributes as CA
import Chart.Events as CE
import Chart.Item as CI
import Chart.Svg as CS
import Csv.Decode as Decode exposing (Decoder, column, float, into, pipeline, string)
import File exposing (File)
import File.Select as Select
import Html as H exposing (Html, div)
import Html.Attributes as HA exposing (style)
import Html.Events as HE
```

3.1 project Source Code

```elm
import Html.Events.Extra.Wheel as Wheel
import Svg as S
import Svg.Attributes as SA
import Task


type alias Point =
    { x : Float
    , y : Float
    , s : Float
    , w : String
    }


getPareto : List Point -> List Point
getPareto points =
    let
        sorted =
            List.sortBy .x points

        pareto =
            List.foldl
                (\p acc ->
                    if
                        List.isEmpty acc
                            || (case List.head acc of
                                    Just headPoint ->
                                        headPoint.y < p.y

                                    Nothing ->
                                        True
                               )
                    then
                        p :: acc
```

3.1 project Source Code

```

                    else
                        acc
                )
                []
                sorted
    in
    List.reverse pareto


decoder : Decoder Point
decoder =
    into Point
        |> pipeline (column 1 float)
        |> pipeline (column 2 float)
        |> pipeline (column 3 float)
        |> pipeline (column 0 string)


type alias Model =
    { center : CS.Point
    , dragging : Dragging
    , percentage : Float
    , data : List Point
    , showSize : Bool
    , showText : Bool
    , showPareto : Bool
    , textX : String
    , textY : String
    }


type Dragging
    = CouldStillBeClick CS.Point
```

3.1 project Source Code

```
    | ForSureDragging CS.Point
    | None


init : () -> ( Model, Cmd Msg )
init _ =
    ( { center = { x = 0, y = 0 }
      , dragging = None
      , percentage = 100
      , data =
            [ { x = 65, y = 45, s = 77, w = "AMD Athlon 64 3500+" }
            , { x = 14, y = 35, s = 192, w = "AMD Athlon 200GE" }
            , { x = 22, y = 80, s = 160, w = "Intel Xeon E5-2603 v2" }
            , { x = 45, y = 125, s = 258, w = "AMD Phenom II X4 980 BE" }
            , { x = 22, y = 95, s = 160, w = "\tIntel Xeon E5-2470 v2" }
            ]
      , showSize = True
      , showText = True
      , showPareto = False
      , textX = "Process Size in nanometers"
      , textY = "Thermal Design Power in Watts"
      }
    , Cmd.none
    )


type Msg
    = OnMouseMove CS.Point
    | OnMouseDown CS.Point
    | OnMouseUp CS.Point CS.Point
    | OnMouseLeave
    | OnZoomIn
    | OnZoomOut
    | OnZoomReset
```

3.1 project Source Code

```elm
    | FileRequested
    | FileUpload File
    | FileLoad String
    | ToggleShowSize
    | ToggleShowText
    | ToggleShowPareto
    | OnWheelEvent Float
    | UpdateTextX String
    | UpdateTextY String


update : Msg -> Model -> ( Model, Cmd Msg )
update msg model =
    case msg of
        OnMouseDown offset ->
            ( { model | dragging = CouldStillBeClick offset }, Cmd.none )

        OnMouseMove offset ->
            case model.dragging of
                CouldStillBeClick prevOffset ->
                    if prevOffset == offset then
                        ( model, Cmd.none )

                    else
                        ( { model
                            | center = updateCenter model.center prevOffset offset
                            , dragging = ForSureDragging offset
                          }
                        , Cmd.none
                        )

                ForSureDragging prevOffset ->
                    ( { model
                        | center = updateCenter model.center prevOffset offset
```

3.1 project Source Code

```
                        , dragging = ForSureDragging offset
                      }
                    , Cmd.none
                    )

                None ->
                    ( model, Cmd.none )

        OnMouseUp offset coord ->
            case model.dragging of
                CouldStillBeClick prevOffset ->
                    ( { model | center = coord, dragging = None }, Cmd.none )

                ForSureDragging prevOffset ->
                    ( { model
                        | center = updateCenter model.center prevOffset offset
                        , dragging = None
                      }
                    , Cmd.none
                    )

                None ->
                    ( model, Cmd.none )

        OnMouseLeave ->
            ( { model | dragging = None }, Cmd.none )

        OnZoomIn ->
            ( { model | percentage = model.percentage + 20 }, Cmd.none )

        OnZoomOut ->
            ( { model | percentage = max 1 (model.percentage - 20) }, Cmd.none )

        OnZoomReset ->
```

3.1 project Source Code

```
            ( { model | percentage = 100, center = { x = 0, y = 0 } }, Cmd.none )

        FileRequested ->
            ( model
            , Select.file [ "text/csv" ] FileUpload
            )

        FileUpload file ->
            ( model, Task.perform FileLoad (File.toString file) )

        FileLoad str ->
            ( { model
                | data =
                    case
                        Decode.decodeCsv Decode.NoFieldNames decoder str
                    of
                        Ok points ->
                            points

                        Err err ->
                            []
              }
            , Cmd.none
            )

        ToggleShowSize ->
            ( { model | showSize = not model.showSize }, Cmd.none )

        ToggleShowText ->
            ( { model | showText = not model.showText }, Cmd.none )

        ToggleShowPareto ->
            ( { model | showPareto = not model.showPareto }, Cmd.none )

```

3.1 project Source Code

```
    OnWheelEvent delta ->
      ( { model
        | percentage =
          if delta > 0 then
            model.percentage + 20

          else
            max 1 (model.percentage - 20)
       }
      , Cmd.none
      )

    UpdateTextX text ->
      ( { model | textX = text }, Cmd.none )

    UpdateTextY text ->
      ( { model | textY = text }, Cmd.none )


updateCenter : CS.Point -> CS.Point -> CS.Point -> CS.Point
updateCenter center prevOffset offset =
  { x = center.x + (prevOffset.x - offset.x)
  , y = center.y + (prevOffset.y - offset.y)
  }


view : Model -> Html Msg
view model =
  div []
    [ div
      [ style "position" "absolute" ]
      [ div []
        [ H.label []
          [ H.input [ HA.type_ "checkbox", HA.checked model.showSize, HE.onClick ToggleShowSize ] []
```

3.1 project Source Code

```
                    , H.text " Show size"
                    ]
                , H.label []
                    [ H.input [ HA.type_ "checkbox", HA.checked model.showText, HE.onClick ToggleShowText ] []
                    , H.text " Show text"
                    ]
                , H.label []
                    [ H.input [ HA.type_ "checkbox", HA.checked model.showPareto, HE.onClick ToggleShowPareto ] []
                    , H.text " Show pareto line"
                    ]
                ]
            , div []
                [ H.input
                    [ HA.value model.textX
                    , HE.onInput UpdateTextX
                    ]
                    []
                , H.input
                    [ HA.value model.textY
                    , HE.onInput UpdateTextY
                    ]
                    []
                ]
            ]
        , div
            [ style "width" "100vw", style "height" "100vh", style "overflow" "hidden", Wheel.onWheel chooseZoom ]
            []
        , div
            [ style "position" "absolute"
            , style "top" "calc(50vh - 40vh)"
            , style "left" "calc(50vw - 40vh)"
            , style "width" "80vh"
            , style "height" "80vh"
            ]
```

3.1 project Source Code

```
            [ C.chart
                [ CA.height 300
                , CA.width 300
                , CA.range [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.x ]
                , CA.domain [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.y ]
                , CE.onMouseDown OnMouseDown CE.getOffset
                , CE.onMouseMove OnMouseMove CE.getOffset
                , CE.on "mouseup" (CE.map2 OnMouseUp CE.getOffset CE.getCoords)
                , CE.onMouseLeave OnMouseLeave
                , CA.htmlAttrs
                    [ HA.style "user-select" "none"
                    , HA.style "cursor" <|
                        case model.dragging of
                            CouldStillBeClick _ ->
                                "grabbing"

                            ForSureDragging _ ->
                                "grabbing"

                            None ->
                                "grab"
                    ]
                ]
                [ C.xLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
                , C.yLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
                , C.xTicks [ CA.amount 10, CA.ints ]
                , C.yTicks [ CA.amount 10, CA.ints ]
                , C.labelAt CA.middle
                    .min
                    [ CA.moveDown 18 ]
                    [ S.text model.textX ]
                , C.labelAt .min
                    CA.middle
                    [ CA.moveLeft 18, CA.rotate 90 ]
```

3.1 project Source Code

```
                    [ S.text model.textY ]
                , C.series .x
                    [ C.scatter .y [ CA.opacity 0.2, CA.borderWidth 1 ]
                        |> C.variation
                            (\_ d ->
                                [ CA.size
                                    (if model.showSize then
                                        d.s * model.percentage / 100 / 10

                                     else
                                        1
                                    )
                                , CA.hideOverflow
                                ]
                            )
                    ]
                    model.data
                , if model.showPareto then
                    C.series .x
                        [ C.interpolated .y [ CA.monotone ] []
                        ]
                        (getPareto model.data)

                  else
                    C.series .x
                        [ C.interpolated .y [ CA.monotone ] []
                        ]
                        []
                , if model.showText then
                    C.eachDot <|
                        \p dot ->
                            [ C.label
                                [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
                                [ S.text (CI.getData dot).w ]
```

3.1 project Source Code

```
                            (CI.getCenter p dot)
                        ]

              else
                C.eachDot <|
                    \_ _ ->
                        []

            -- C.eachDot <|
            --     \p dot ->
            --         [ C.label
            --             [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
            --             [ S.text (CI.getData dot).w ]
            --             (CI.getCenter p dot)
            --         ]
            -- , C.withPlane <|
            --     \p ->
            --         [ C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.y1 (CA.middle p.y) ]
            --         , C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.x1 (CA.middle p.x) ]
            --         ]
            , C.htmlAt .max
                .max
                0
                0
                [ HA.style "transform" "translateX(-100%)" ]
                [ H.span
                    [ HA.style "margin-right" "5px" ]
                    [ H.text (String.fromFloat model.percentage ++ "%") ]
                , H.button
                    [ HE.onClick OnZoomIn
                    , HA.style "margin-right" "5px"
                    ]
                    [ H.text "+" ]
                , H.button
```

3.1 project Source Code

```
                        [ HE.onClick OnZoomOut
                        , HA.style "margin-right" "5px"
                        ]
                        [ H.text "-" ]
                    , H.button
                        [ HE.onClick OnZoomReset ]
                        [ H.text "Reset" ]
                    , H.button [ HE.onClick FileRequested ] [ H.text "Load CSV" ]
                    ]
                ]
            ]
        ]


chooseZoom : Wheel.Event -> Msg
chooseZoom wheelEvent =
    case wheelEvent of
        event ->
            OnWheelEvent event.deltaY


meta =
    { category = "Interactivity"
    , categoryOrder = 5
    , name = "Zoom"
    , description = "Add zoom effect."
    , order = 20
    }

```

3.2 slides Source Code

The typst code generating the slides. `main.typ`

typ

```
#import "@preview/touying:0.6.1": *
#import "@preview/pinit:0.2.0": *
#import themes.metropolis: *
#import "@preview/numbly:0.1.0": numbly
#import "@preview/codly:1.0.0": *

#show: codly-init.with()
#show: metropolis-theme.with(
  aspect-ratio: "16-9",
  footer: self => self.info.institution,
  config-info(
    title: [Visualize Me],
    subtitle: [Explore the quickly updated world],
    author: [Zining Wang \@ Northeastern University],
    date: datetime.today(),
    institution: [EECE5642 Project Proposal ],
  ),
)
#set heading(numbering: numbly("{1}.", default: "1.1"))
#title-slide()
= Outline <touying:hidden>

#outline(title: none, indent: 1em, depth: 2)
= Project Information
== Project Information
#grid(columns: 2)[
  - Team Members & contributions:
    - Zining Wang: Idea, Design, Implementation and Presentation

  - Used Language: elm-lang

```

3.2 slides Source Code

```
  - Source Code: https://github.com/wznmickey/visualizeMe

  - Online Demo: https://wznmickey.github.io/visualizeMe/

  - Data Source:
    - https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data
    - https://llm-stats.com/
][#grid(columns: 1, align: center)[#image("tempResult.png") ][
    Current Appearance]]

== Motivation

In today’s fast-paced technological world, advancements in hardware and artificial intelligence (AI) are occurring at breakneck speed. This creates a dilemma for consumers and professionals who seek to stay up-to-date with the best performing systems without breaking the bank. While tools for benchmarking hardware, particularly CPUs and GPUs, have become increasingly sophisticated, the problem lies in the fact that top-tier performance is not necessarily the most cost-effective solution for everyone.

When selecting hardware, it’s important to recognize that performance should not be the only consideration. Several other factors play a crucial role in choosing the right setup for a given application.
#pagebreak()
- Power Consumption: While higher performance often correlates with higher power requirements, a balance must be struck to avoid excessive energy costs or hardware overheating.

- Price: The most powerful hardware often comes with a premium price tag, but for many users, the best performance is not required for their daily tasks. For example, a developer or gamer may need just enough GPU power to run programs efficiently, but not necessarily the highest-end models available.
#pagebreak()
- Specific Use Case: Different use cases demand different hardware characteristics.
  - Single-core performance might be essential for applications that rely heavily on sequential processing (e.g., certain older games or single-threaded applications).

  - Multi-core performance is critical for tasks like rendering, scientific computing, and modern gaming where parallel processing is leveraged.

  - For GPUs, some users prioritize high encoding and decoding ability for video production, while others might prefer high memory bandwidth for tasks such as gaming, machine learning, or 3D rendering.
#pagebreak()
In addition to hardware considerations, there are also other contexts in which performance must be evaluated. For instance, in the realm of Large Language Models (LLMs) and other AI models, the user’s needs must be balanced across several dimensions. As LLMs continue to evolve, model size, accuracy, and processing speed are important factors to weigh. Some users may prioritize faster inference times, while others may be more concerned with the accuracy of the results. Additionally, some may be restricted by computational resources, thus requiring smaller models that can still offer competitive performance.
```

3.2 slides Source Code

```

== Current Situation
Hard to filter the desired parameters.
#grid(columns: (1fr, 0.3fr, 1fr))[https://socpk.com/allperf/
  #image("socpk.png", height: 80%)][][
  https://llm-stats.com/
  #image("LLM.png")
]



= Develop Process

== Developed Part
#grid(columns: 2)[
  To make the tool acessible to more people, I want to develop a web-based tool and I use elm-lang which is a functional programming language that could be compiled into a static HTML file with JavaScript containing SVG images which could be easily hosted on GitHub Pages service.
][#grid(columns: 1, align: center)[#image("tempResult.png") ][
    Current Appearance]]

== Live Example

https://wznmickey.github.io/visualizeMe/


1. Zoom in and out by pressing the button.
2. Reset the zoom by pressing the button.
3. Zoom in and out by the wheel. (Thanks to the suggestion that using the wheel to zoom in my proposal)
4. Move the plot by clicking the new center.
5. Drag the plot to move it. (Thanks to the suggestion that removing the + line at the center of the screen in my proposal)
6. Switch on the text, size and the pareto line.
7. Upload the data and visualize it.
8. Change the x-axis, y-axis text.

```

3.2 slides Source Code

```
== Challenge

1. Lack of documentation and packages for elm-lang. e.g. csv parsing,key binding
2. Lack of contributor. I work alone in this project.
3. Lack of experience in web frontend development. I am not familiar with tools like CSS in organizing different elements in a page.


== Current bugs & Undeveloped Part

There are some bugs in the current version:
- The text would not disappear when it is out of the screen.
- The size of each point could not resize according to the relative size of the data.
- Text may of x-axis and y-axis may be overlapped by the data point features.
- The layer of the element to capture wheel is lower than that of the element to show the data which causes the wheel event not captured in the center.

To-do:
- Filter different data columns in the csv file.
- When the mouse is moved onto each point, the text of the point should be shown.
- Better design of the layout of the page.

#focus-slide[

  *Developed For*

  computer DIY players

  LLM local deployment users & researchers

  People who have many options to choose from

  https://wznmickey.github.io/visualizeMe/
]

= Appendix
```

3.2 slides Source Code

```

== project Source Code
`Main.elm`
#{
  set text(size: 8pt)
  let x = read("../src/Main.elm")
  raw(x, block: true, lang: "elm")
}
`Zoom.elm`
#{
  set text(size: 8pt)
  let x = read("../src/Zoom.elm")
  raw(x, block: true, lang: "elm")
}


== slides Source Code
The typst code generating the slides.
`main.typ`
#{
  set text(size: 8pt)
  let x = read("./main.typ")
  raw(x, block: true, lang: "typ")
}

```