

Visualize Me

Explore the quickly updated world

Zining Wang @ Northeastern University

2025-04-04

EECE5642 Project Proposal

1. Project Information	2
1.1 Project Information	3
1.2 Motivation	4
1.3 Current Situation	8
2. Develop Process	9
2.1 Developed Part	10
2.2 Live Example	11
2.3 Challenge	12
2.4 Current bugs & Undeveloped Part	13
3. Appendix	14
3.1 project Source Code	15
3.2 slides Source Code	28

1. Project Information

1.1 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/>



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.

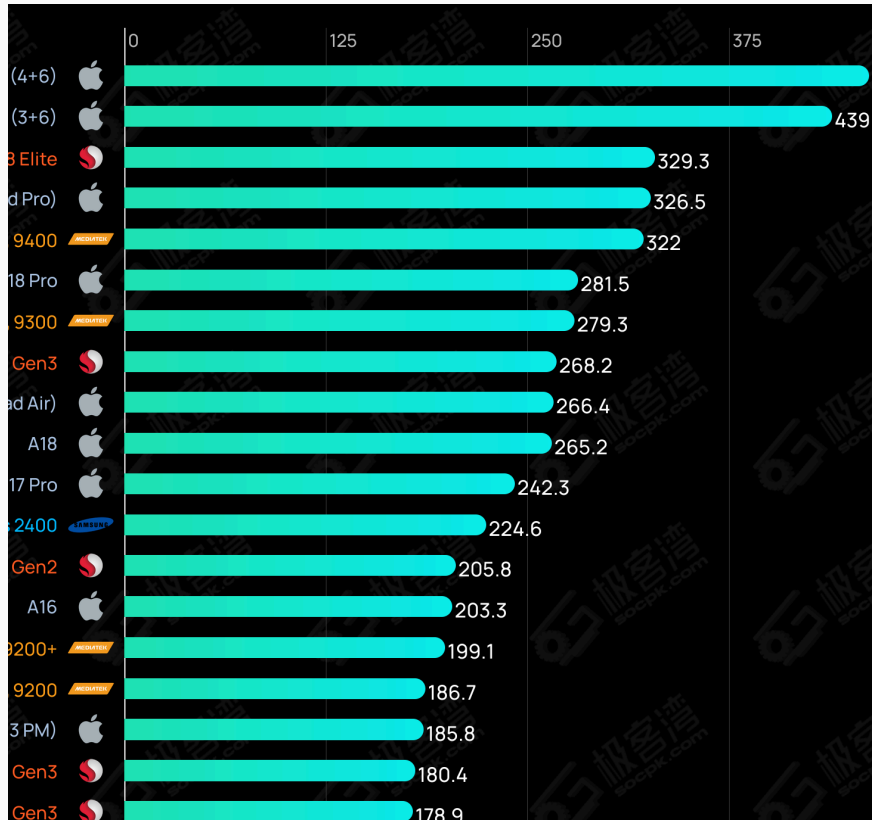
1.2 Motivation

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.

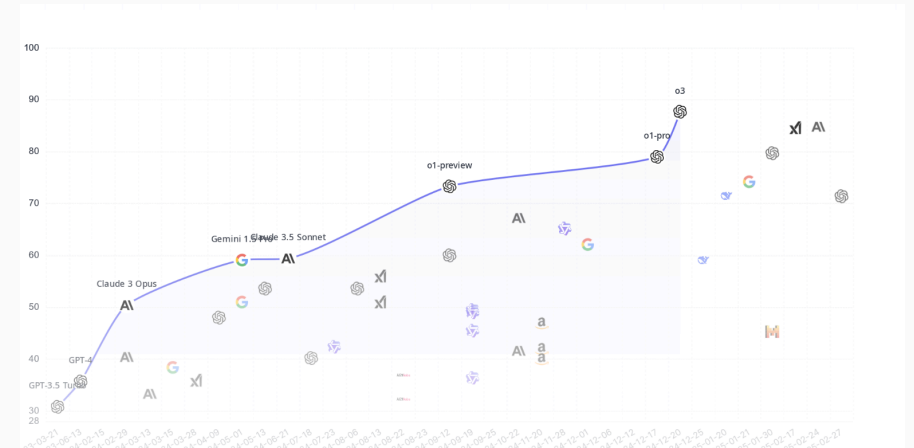
1.3 Current Situation

Hard to filter the desired parameters.

<https://socpk.com/allperf/>



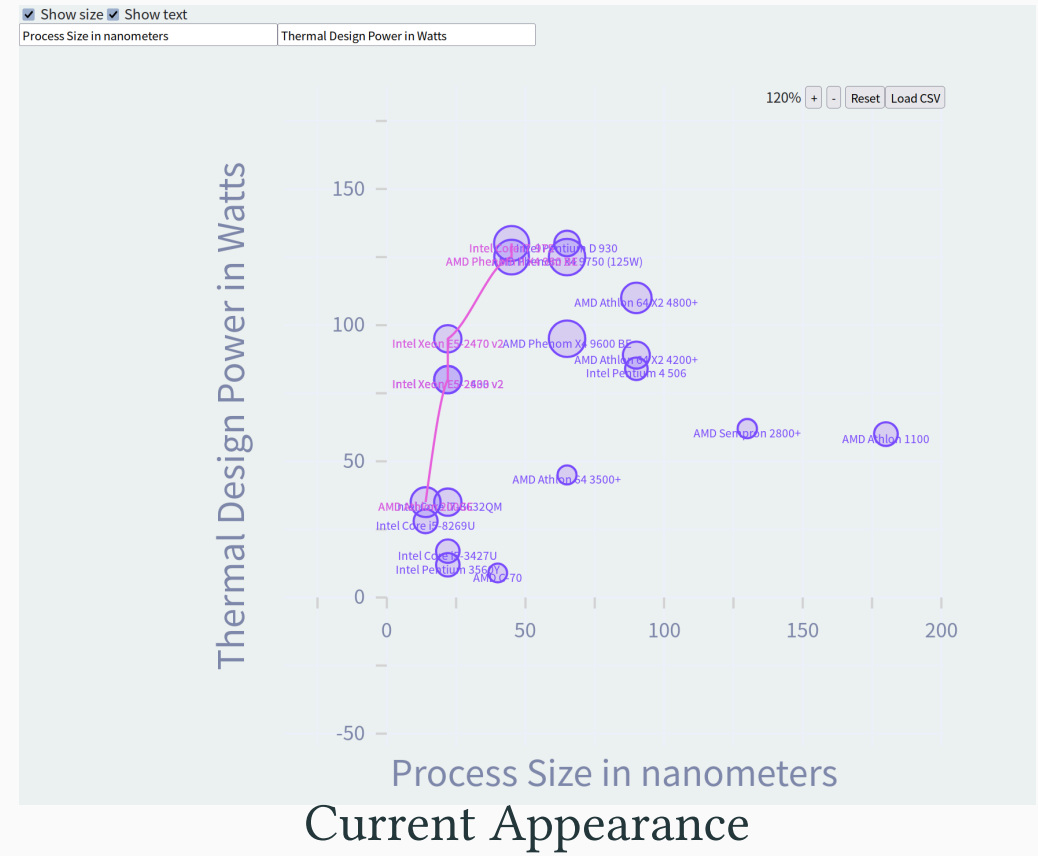
<https://llm-stats.com/>



2. Develop Process

2.1 Developed Part

To make the tool accessible 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.



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

```
1  module Main exposing (..)
2
3  import Browser
4  import Zoom exposing (..)
5
6
7  main =
8      Browser.element
9          { init = init
10            , view = view
11            , update = update
12            , subscriptions = \_ -> Sub.none
13            }
14
```

elm

Zoom.elm

```
1  module Zoom exposing (..)
2
3  import Chart as C
4  import Chart.Attributes as CA
5  import Chart.Events as CE
6  import Chart.Item as CI
7  import Chart.Svg as CS
8  import Csv.Decode as Decode exposing (Decoder, column, float, into, pipeline, string)
9  import File exposing (File)
10 import File.Select as Select
11 import Html as H exposing (Html, div)
12 import Html.Attributes as HA exposing (style)
13 import Html.Events as HE
```

elm

3.1 project Source Code

```
14 import Html.Events.Extra.Wheel as Wheel
15 import Svg as S
16 import Svg.Attributes as SA
17 import Task
18
19
20 type alias Point =
21   { x : Float
22     , y : Float
23     , s : Float
24     , w : String
25   }
26
27
28 getPareto : List Point -> List Point
29 getPareto points =
30   let
31     sorted =
32       List.sortBy .x points
33
34     pareto =
35       List.foldl
36         (\p acc ->
37           if
38             List.isEmpty acc
39             || (case List.head acc of
40                 Just headPoint ->
41                   headPoint.y < p.y
42                 Nothing ->
43                   True
44             )
45         then
46           p :: acc
```

3.1 project Source Code

```
48
49         else
50             acc
51     )
52     []
53     sorted
54     in
55     List.reverse pareto
56
57
58     decoder : Decoder Point
59     decoder =
60         into Point
61         |> pipeline (column 1 float)
62         |> pipeline (column 2 float)
63         |> pipeline (column 3 float)
64         |> pipeline (column 0 string)
65
66
67     type alias Model =
68     { center : CS.Point
69     , dragging : Dragging
70     , percentage : Float
71     , data : List Point
72     , showSize : Bool
73     , showText : Bool
74     , showPareto : Bool
75     , textX : String
76     , textY : String
77     }
78
79
80     type Dragging
81     = CouldStillBeClick CS.Point
```

3.1 project Source Code

```
82     | ForSureDragging CS.Point
83     | None
84
85
86   init : () -> ( Model, Cmd Msg )
87   init _ =
88     ( { center = { x = 0, y = 0 }
89       , dragging = None
90       , percentage = 100
91       , data =
92         [ { x = 65, y = 45, s = 77, w = "AMD Athlon 64 3500+" }
93           , { x = 14, y = 35, s = 192, w = "AMD Athlon 200GE" }
94           , { x = 22, y = 80, s = 160, w = "Intel Xeon E5-2603 v2" }
95           , { x = 45, y = 125, s = 258, w = "AMD Phenom II X4 980 BE" }
96           , { x = 22, y = 95, s = 160, w = "\tIntel Xeon E5-2470 v2" }
97         ]
98       , showSize = True
99       , showText = True
100       , showPareto = False
101       , textX = "Process Size in nanometers"
102       , textY = "Thermal Design Power in Watts"
103     }
104     , Cmd.none
105   )
106
107
108   type Msg
109   = OnMouseMove CS.Point
110   | OnMouseDown CS.Point
111   | OnMouseUp CS.Point CS.Point
112   | OnMouseLeave
113   | OnZoomIn
114   | OnZoomOut
115   | OnZoomReset
```

3.1 project Source Code

```
116 | FileRequested
117 | FileUpload File
118 | FileLoad String
119 | ToggleShowSize
120 | ToggleShowText
121 | ToggleShowPareto
122 | OnWheelEvent Float
123 | UpdateTextX String
124 | UpdateTextY String
125
126
127 update : Msg -> Model -> ( Model, Cmd Msg )
128 update msg model =
129   case msg of
130     OnMouseDown offset ->
131       ( { model | dragging = CouldStillBeClick offset }, Cmd.none )
132
133     OnMouseMove offset ->
134       case model.dragging of
135         CouldStillBeClick prevOffset ->
136           if prevOffset == offset then
137             ( model, Cmd.none )
138
139         else
140           ( { model
141             | center = updateCenter model.center prevOffset offset
142             , dragging = ForSureDragging offset
143           }
144           , Cmd.none
145           )
146
147     ForSureDragging prevOffset ->
148       ( { model
149         | center = updateCenter model.center prevOffset offset
```

3.1 project Source Code

```
150         , dragging = ForSureDragging offset
151     }
152     , Cmd.none
153 )
154
155 None ->
156     ( model, Cmd.none )
157
158 OnMouseUp offset coord ->
159     case model.dragging of
160         CouldStillBeClick prevOffset ->
161             ( { model | center = coord, dragging = None }, Cmd.none )
162
163         ForSureDragging prevOffset ->
164             ( { model
165                 | center = updateCenter model.center prevOffset offset
166                 , dragging = None
167             }
168             , Cmd.none
169             )
170
171 None ->
172     ( model, Cmd.none )
173
174 OnMouseLeave ->
175     ( { model | dragging = None }, Cmd.none )
176
177 OnZoomIn ->
178     ( { model | percentage = model.percentage + 20 }, Cmd.none )
179
180 OnZoomOut ->
181     ( { model | percentage = max 1 (model.percentage - 20) }, Cmd.none )
182
183 OnZoomReset ->
```

3.1 project Source Code

```
184      ( { model | percentage = 100, center = { x = 0, y = 0 } }, Cmd.none )
185
186      FileRequested ->
187      ( model
188      , Select.file [ "text/csv" ] FileUpload
189      )
190
191      FileUpload file ->
192      ( model, Task.perform FileLoad (File.toString file) )
193
194      FileLoad str ->
195      ( { model
196      | data =
197      case
198      Decode.decodeCsv Decode.NoFieldNames decoder str
199      of
200      Ok points ->
201      points
202
203      Err err ->
204      []
205      }
206      , Cmd.none
207      )
208
209      ToggleShowSize ->
210      ( { model | showSize = not model.showSize }, Cmd.none )
211
212      ToggleShowText ->
213      ( { model | showText = not model.showText }, Cmd.none )
214
215      ToggleShowPareto ->
216      ( { model | showPareto = not model.showPareto }, Cmd.none )
217
```

3.1 project Source Code

```
218     OnWheelEvent delta ->
219     ( { model
220       | percentage =
221         if delta > 0 then
222           model.percentage + 20
223
224         else
225           max 1 (model.percentage - 20)
226       }
227     , Cmd.none
228     )
229
230     UpdateTextX text ->
231     ( { model | textX = text }, Cmd.none )
232
233     UpdateTextY text ->
234     ( { model | textY = text }, Cmd.none )
235
236
237     updateCenter : CS.Point -> CS.Point -> CS.Point -> CS.Point
238     updateCenter center prevOffset offset =
239     { x = center.x + (prevOffset.x - offset.x)
240     , y = center.y + (prevOffset.y - offset.y)
241     }
242
243
244     view : Model -> Html Msg
245     view model =
246     div []
247     [ div
248       [ style "position" "absolute" ]
249       [ div []
250         [ H.label []
251           [ H.input [ HA.type_ "checkbox", HA.checked model.showSize, HE.onClick ToggleShowSize ] []
```


3.1 project Source Code

```
252         , H.text " Show size"
253     ]
254     , H.label []
255     [ H.input [ HA.type_ "checkbox", HA.checked model.showText, HE.onClick ToggleShowText ] []
256       , H.text " Show text"
257     ]
258     , H.label []
259     [ H.input [ HA.type_ "checkbox", HA.checked model.showPareto, HE.onClick ToggleShowPareto ] []
260       , H.text " Show pareto line"
261     ]
262 ]
263 , div []
264   [ H.input
265     [ HA.value model.textX
266       , HE.onInput UpdateTextX
267     ]
268     []
269     , H.input
270     [ HA.value model.textY
271       , HE.onInput UpdateTextY
272     ]
273     []
274   ]
275 ]
276 , div
277   [ style "width" "100vw", style "height" "100vh", style "overflow" "hidden", Wheel.onWheel chooseZoom ]
278   []
279 , div
280   [ style "position" "absolute"
281     , style "top" "calc(50vh - 40vh)"
282     , style "left" "calc(50vw - 40vw)"
283     , style "width" "80vh"
284     , style "height" "80vh"
285   ]
```

3.1 project Source Code

```
286      [ C.chart
287        [ CA.height 300
288          , CA.width 300
289          , CA.range [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.x ]
290          , CA.domain [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.y ]
291          , CE.onMouseDown OnMouseDown CE.getOffset
292          , CE.onMouseMove OnMouseMove CE.getOffset
293          , CE.on "mouseup" (CE.map2 OnMouseUp CE.getOffset CE.getCoords)
294          , CE.onMouseLeave OnMouseLeave
295          , CA.htmlAttrs
296            [ HA.style "user-select" "none"
297              , HA.style "cursor" <|
298                case model.dragging of
299                  CouldStillBeClick _ ->
300                    "grabbing"
301
302                  ForSureDragging _ ->
303                    "grabbing"
304
305                  None ->
306                    "grab"
307            ]
308      ]
309      [ C.xLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
310        , C.yLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
311        , C.xTicks [ CA.amount 10, CA.ints ]
312        , C.yTicks [ CA.amount 10, CA.ints ]
313        , C.labelAt CA.middle
314          .min
315          [ CA.moveDown 18 ]
316          [ S.text model.textX ]
317        , C.labelAt .min
318          CA.middle
319          [ CA.moveLeft 18, CA.rotate 90 ]
```

3.1 project Source Code

```
320         [ S.text model.textY ]
321     , C.series .x
322         [ C.scatter .y [ CA.opacity 0.2, CA.borderWidth 1 ]
323           |> C.variation
324             (\_ d ->
325               [ CA.size
326                 (if model.showSize then
327                   d.s * model.percentage / 100 / 10
328
329                 else
330                   1
331               )
332             , CA.hideOverflow
333           ]
334         )
335     ]
336     model.data
337     , if model.showPareto then
338       C.series .x
339         [ C.interpolated .y [ CA.monotone ] []
340       ]
341       (getPareto model.data)
342
343     else
344       C.series .x
345         [ C.interpolated .y [ CA.monotone ] []
346       ]
347       []
348     , if model.showText then
349       C.eachDot <|
350         \p dot ->
351           [ C.label
352             [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
353             [ S.text (CI.getData dot).w ]
```

3.1 project Source Code

```
354         (CI.getCenter p dot)
355     ]
356
357     else
358         C.eachDot <|
359             \_ _ ->
360                 []
361
362         -- C.eachDot <|
363             -- \p dot ->
364                 -- [ C.label
365                     --     [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
366                     --     [ S.text (CI.getData dot).w ]
367                     --     (CI.getCenter p dot)
368                     -- ]
369             -- , C.withPlane <|
370                 -- \p ->
371                 -- [ C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.y1 (CA.middle p.y) ]
372                 --     , C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.x1 (CA.middle p.x) ]
373                 -- ]
374         , C.htmlAt .max
375             .max
376             0
377             0
378             [ HA.style "transform" "translateX(-100%)" ]
379             [ H.span
380                 [ HA.style "margin-right" "5px" ]
381                 [ H.text (String.fromFloat model.percentage ++ "%") ]
382             , H.button
383                 [ HE.onClick OnZoomIn
384                     , HA.style "margin-right" "5px"
385                 ]
386                 [ H.text "+" ]
387             , H.button
```

3.1 project Source Code

```
388         [ HE.onClick OnZoomOut
389         , HA.style "margin-right" "5px"
390         ]
391         [ H.text "-" ]
392     , H.button
393         [ HE.onClick OnZoomReset ]
394         [ H.text "Reset" ]
395     , H.button [ HE.onClick FileRequested ] [ H.text "Load CSV" ]
396     ]
397 ]
398 ]
399 ]
400
401
402 chooseZoom : Wheel.Event -> Msg
403 chooseZoom wheelEvent =
404     case wheelEvent of
405         event ->
406             OnWheelEvent event.deltaY
407
408
409 meta =
410     { category = "Interactivity"
411     , categoryOrder = 5
412     , name = "Zoom"
413     , description = "Add zoom effect."
414     , order = 20
415     }
416
```

3.2 slides Source Code

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

```
1  #import "@preview/touying:0.6.1": *
2  #import "@preview/pinit:0.2.0": *
3  #import themes.metropolis: *
4  #import "@preview/numbly:0.1.0": numbly
5  #import "@preview/codly:1.0.0": *
6
7  #show: codly-init.with()
8  #show: metropolis-theme.with(
9    aspect-ratio: "16-9",
10    footer: self => self.info.institution,
11    config-info(
12      title: [Visualize Me],
13      subtitle: [Explore the quickly updated world],
14      author: [Zining Wang \@ Northeastern University],
15      date: datetime.today(),
16      institution: [EECE5642 Project Proposal ],
17    ),
18  )
19  #set heading(numbering: numbly("{1}.", default: "1.1"))
20  #title-slide()
21  = Outline <touying:hidden>
22
23  #outline(title: none, indent: 1em, depth: 2)
24  = Project Information
25  == Project Information
26  #grid(columns: 2)[
27    - Team Members & contributions:
28      - Zining Wang: Idea, Design, Implementation and Presentation
29
30    - Used Language: elm-lang
31
```

typ

3.2 slides Source Code

```
32 - Source Code: https://github.com/wznmickey/visualizeMe
33
34 - Online Demo: https://wznmickey.github.io/visualizeMe/
35
36 - Data Source:
37   - https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data
38   - https://llm-stats.com/
39 ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
40   Current Appearance]]
41
42 == Motivation
43
44 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
45 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
46 sophisticated, the problem lies in the fact that top-tier performance is not necessarily the most cost-effective solution for everyone.
47
48 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
49 application.
50 #pagebreak()
51 - Power Consumption: While higher performance often correlates with higher power requirements, a balance must be struck to avoid excessive energy costs or hardware overheating.
52
53 - 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
54 need just enough GPU power to run programs efficiently, but not necessarily the highest-end models available.
55 #pagebreak()
56 - Specific Use Case: Different use cases demand different hardware characteristics.
57   - Single-core performance might be essential for applications that rely heavily on sequential processing (e.g., certain older games or single-threaded applications).
58
59   - Multi-core performance is critical for tasks like rendering, scientific computing, and modern gaming where parallel processing is leveraged.
60
61   - 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
62 rendering.
63 #pagebreak()
64 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,
65 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
66 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
67 that can still offer competitive performance.
```

3.2 slides Source Code

```
60
61 == Current Situation
62 Hard to filter the desired parameters.
63 #grid(columns: (1fr, 0.3fr, 1fr))[https://socpk.com/allperf/
64   #image("socpk.png", height: 80%)][]
65   https://llm-stats.com/
66   #image("LLM.png")
67 ]
68
69
70
71 == Develop Process
72
73 == Developed Part
74 #grid(columns: 2)[
75   To make the tool accessible 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
76   with JavaScript containing SVG images which could be easily hosted on GitHub Pages service.
77   ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
78     Current Appearance]]
79
80
81 == Live Example
82
83
84 1. Zoom in and out by pressing the button.
85 2. Reset the zoom by pressing the button.
86 3. Zoom in and out by the wheel. (Thanks to the suggestion that using the wheel to zoom in my proposal)
87 4. Move the plot by clicking the new center.
88 5. Drag the plot to move it. (Thanks to the suggestion that removing the + line at the center of the screen in my proposal)
89 6. Switch on the text, size and the pareto line.
90 7. Upload the data and visualize it.
91 8. Change the x-axis, y-axis text.
92
```


3.2 slides Source Code

```
93 == Challenge
94
95 1. Lack of documentation and packages for elm-lang. e.g. csv parsing, key binding
96 2. Lack of contributor. I work alone in this project.
97 3. Lack of experience in web frontend development. I am not familiar with tools like CSS in organizing different elements in a page.
98
99
100 == Current bugs & Undeveloped Part
101
102 There are some bugs in the current version:
103 - The text would not disappear when it is out of the screen.
104 - The size of each point could not resize according to the relative size of the data.
105 - Text may of x-axis and y-axis may be overlapped by the data point features.
106 - 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.
107
108 To-do:
109 - Filter different data columns in the csv file.
110 - When the mouse is moved onto each point, the text of the point should be shown.
111 - Better design of the layout of the page.
112
113 #focus-slide[
114
115 *Developed For*
116
117 computer DIY players
118
119 LLM local deployment users & researchers
120
121 People who have many options to choose from
122
123 https://wznmickey.github.io/visualizeMe/
124 ]
125
126 == Appendix
```

3.2 slides Source Code

```
127
128 == project Source Code
129 `Main.elm`
130 #{
131   set text(size: 8pt)
132   let x = read("../src/Main.elm")
133   raw(x, block: true, lang: "elm")
134 }
135 `Zoom.elm`
136 #{
137   set text(size: 8pt)
138   let x = read("../src/Zoom.elm")
139   raw(x, block: true, lang: "elm")
140 }
141
142
143 == slides Source Code
144 The typst code generating the slides.
145 `main.typ`
146 #{
147   set text(size: 8pt)
148   let x = read("../main.typ")
149   raw(x, block: true, lang: "typ")
150 }
151
```