

Visualize Me

Explore the quickly updated world

Zining Wang @ Northeastern University

2025-03-21

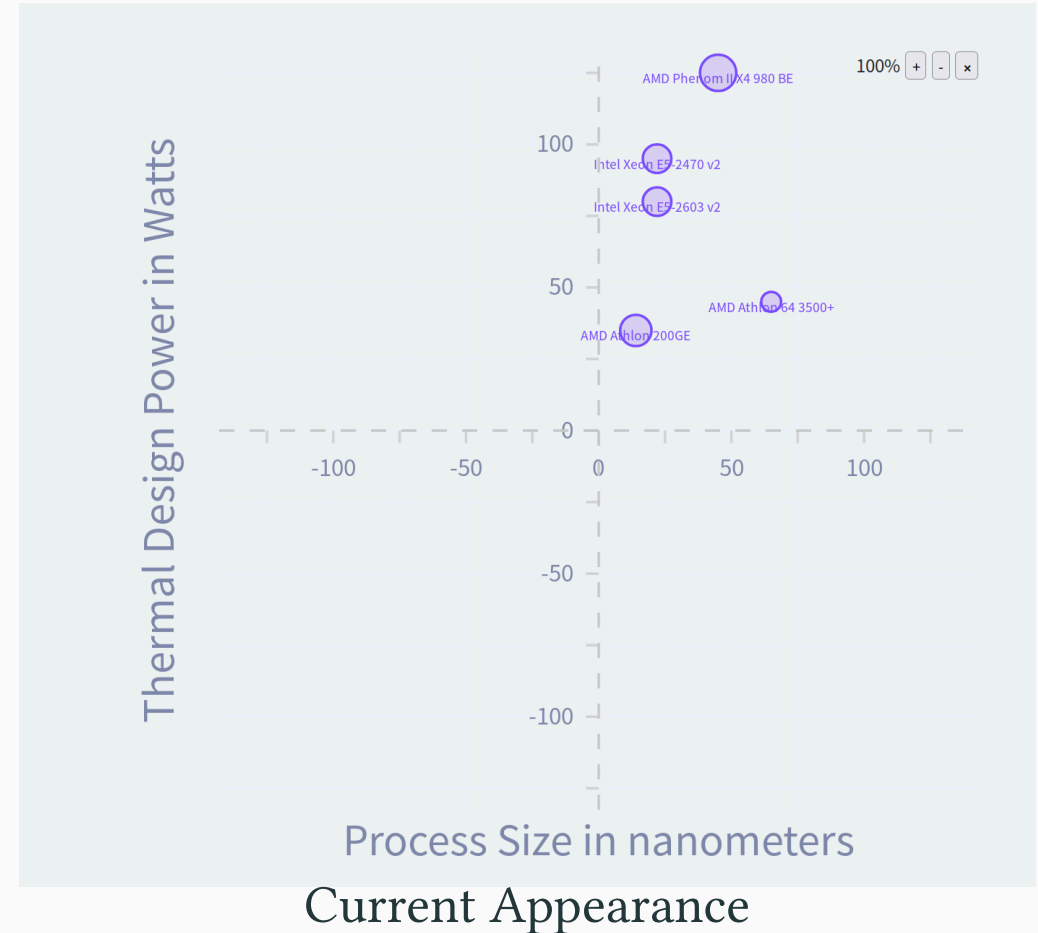
EECE5642 Project Proposal

1. Project Information	2
1.1 Project Information	3
1.2 Motivation	4
2. Develop Process	5
2.1 Developed Part	6
2.2 Developing Plan	7
3. Appendix	8
3.1 project Source Code	9
3.2 slides Source Code	16

1. Project Information

1.1 Project Information

- Team Members & responsibilities
 - Zining Wang: Full software development
- Used Language: elm-lang
- Source Code: <https://github.com/wznmickey/visualizeMe>
- Online Demo: <https://wznmickey.github.io/visualizeMe/>
- Data Source (Current): <https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data>



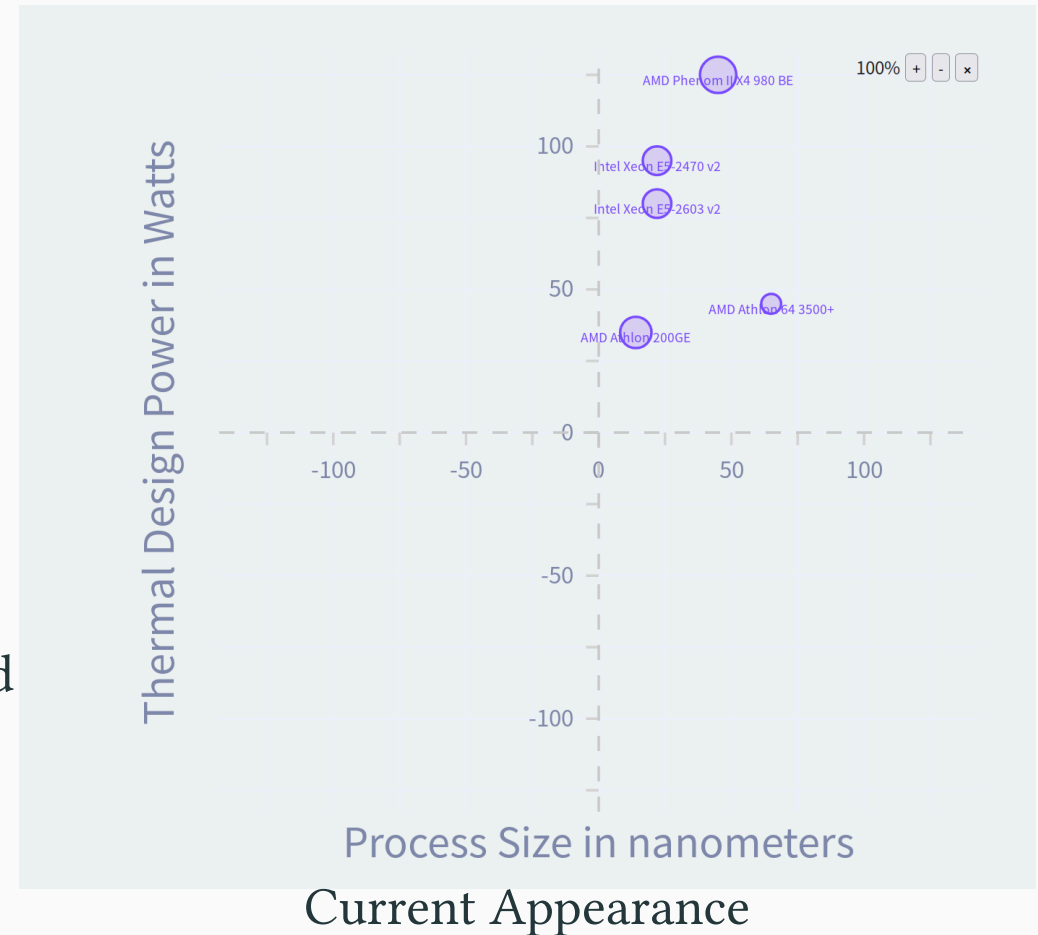
1.2 Motivation

Nowadays, technology is developing rapidly and the price is higher and higher if you want to get the best performance. While there are many tools showing the performance of the hardware like CPUs, the top performance is not always the best choice for everyone due to the price. Buying a suitable hardware is not focusing on one aspect but the overall balance between performance, power, price and so on. Different people may have different performance requirements preference on maybe single-core performance or multi-core performance. For GPU, one may want high encoding ability while another may want high bandwidth. Besides, in some other areas like LLM, when we want to choose a model, we would need to balance the speed, size, and accuracy of the model. And as the LLM is under the development, it is unfair to compare today's LLM with the model developed one year ago. So I want to develop a tool that could let people choose the measurement dimensions they care about and compare them in a 2D and 3D plot.

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. Now I have developed the basic structure of the tool. In the right image, the x axis is the process Size in Nanomeeters, the y axis is the TDP power and the area of each point is the die size. The user could also zoom in and zoom out to see the details of the data.



2.2 Developing Plan

I plan to do the following things later:

1. Support the feature that the user could choose how x-axis, y-axis, area are bind to the data and the user could filter some data out.
2. Support the feature that the user could upload any data and visualize it.
3. Support the feature that the user could choose the 3D mode and the user could choose one more axis and rotate the 3D plot. (If time allows)

Thanks for listening

<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 Html as H exposing (Html, div)
9  import Html.Attributes as HA exposing (style)
10 import Html.Events as HE
11 import Svg as S
12 import Svg.Attributes as SA
13
```

elm

3.1 project Source Code

```
14
15 type alias Model =
16     { center : CS.Point
17     , dragging : Dragging
18     , percentage : Float
19     }
20
21
22 type Dragging
23     = CouldStillBeClick CS.Point
24     | ForSureDragging CS.Point
25     | None
26
27
28 init : () -> ( Model, Cmd Msg )
29 init _ =
30     ( { center = { x = 0, y = 0 }
31     , dragging = None
32     , percentage = 100
33     }
34     , Cmd.none
35     )
36
37
38 type Msg
39     = OnMouseMove CS.Point
40     | OnMouseDown CS.Point
41     | OnMouseUp CS.Point CS.Point
42     | OnMouseLeave
43     | OnZoomIn
44     | OnZoomOut
45     | OnZoomReset
46
47
```

3.1 project Source Code

```
48 update : Msg -> Model -> ( Model, Cmd Msg )
49 update msg model =
50     case msg of
51         OnMouseDown offset ->
52             ( { model | dragging = CouldStillBeClick offset }, Cmd.none )
53
54         OnMouseMove offset ->
55             case model.dragging of
56                 CouldStillBeClick prevOffset ->
57                     if prevOffset == offset then
58                         ( model, Cmd.none )
59
60                 else
61                     ( { model
62                         | center = updateCenter model.center prevOffset offset
63                         , dragging = ForSureDragging offset
64                     }
65                     , Cmd.none
66                     )
67
68                 ForSureDragging prevOffset ->
69                     ( { model
70                         | center = updateCenter model.center prevOffset offset
71                         , dragging = ForSureDragging offset
72                     }
73                     , Cmd.none
74                     )
75
76         None ->
77             ( model, Cmd.none )
78
79     OnMouseUp offset coord ->
80         case model.dragging of
81             CouldStillBeClick prevOffset ->
```

3.1 project Source Code

```
82         ( { model | center = coord, dragging = None }, Cmd.none )
83
84     ForSureDragging prevOffset ->
85         ( { model
86             | center = updateCenter model.center prevOffset offset
87             , dragging = None
88         }
89         , Cmd.none
90         )
91
92     None ->
93         ( model, Cmd.none )
94
95     OnMouseLeave ->
96         ( { model | dragging = None }, Cmd.none )
97
98     OnZoomIn ->
99         ( { model | percentage = model.percentage + 20 }, Cmd.none )
100
101     OnZoomOut ->
102         ( { model | percentage = max 1 (model.percentage - 20) }, Cmd.none )
103
104     OnZoomReset ->
105         ( { model | percentage = 100, center = { x = 0, y = 0 } }, Cmd.none )
106
107
108 updateCenter : CS.Point -> CS.Point -> CS.Point -> CS.Point
109 updateCenter center prevOffset offset =
110     { x = center.x + (prevOffset.x - offset.x)
111     , y = center.y + (prevOffset.y - offset.y)
112     }
113
114
115 view : Model -> Html Msg
```

3.1 project Source Code

```
116 view model =
117   div
118     [ style "position" "absolute"
119       , style "top" "50%"
120       , style "left" "50%"
121       , style "width" "700px"
122       , style "height" "700px"
123       , style "transform" "translate(-50%, -50%)"
124     ]
125     [ C.chart
126       [ CA.height 300
127         , CA.width 300
128         , CA.range [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.x ]
129         , CA.domain [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.y ]
130         , CE.onMouseDown OnMouseDown CE.getOffset
131         , CE.onMouseMove OnMouseMove CE.getOffset
132         , CE.on "mouseup" (CE.map2 OnMouseUp CE.getOffset CE.getCoords)
133         , CE.onMouseLeave OnMouseLeave
134         , CA.htmlAttrs
135           [ HA.style "user-select" "none"
136             , HA.style "cursor" <|
137               case model.dragging of
138                 CouldStillBeClick _ ->
139                   "grabbing"
140
141                 ForSureDragging _ ->
142                   "grabbing"
143
144                 None ->
145                   "grab"
146           ]
147       ]
148     [ C.xLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
149       , C.yLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
```

3.1 project Source Code

```
150      , C.xTicks [ CA.amount 10, CA.ints ]
151      , C.yTicks [ CA.amount 10, CA.ints ]
152      , C.labelAt CA.middle
153      .min
154      [ CA.moveDown 18 ]
155      [ S.text "Process Size in nanometers" ]
156      , C.labelAt .min
157      CA.middle
158      [ CA.moveLeft 18, CA.rotate 90 ]
159      [ S.text "Thermal Design Power in Watts" ]
160      , C.series .x
161      [ C.scatter .y [ CA.opacity 0.2, CA.borderWidth 1 ]
162      |> C.variation (\_ d -> [ CA.size (d.s * model.percentage / 100 / 10), CA.hideOverflow ])
163      ]
164      [ { x = 65, y = 45, s = 77, w = "AMD Athlon 64 3500+" }
165      , { x = 14, y = 35, s = 192, w = "AMD Athlon 200GE" }
166      , { x = 22, y = 80, s = 160, w = "Intel Xeon E5-2603 v2" }
167      , { x = 45, y = 125, s = 258, w = "AMD Phenom II X4 980 BE" }
168      , { x = 22, y = 95, s = 160, w = "\tIntel Xeon E5-2470 v2" }
169      ]
170      , C.eachDot <|
171      \p dot ->
172      [ C.label
173      [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
174      [ S.text (CI.getData dot).w ]
175      (CI.getCenter p dot)
176      ]
177      , C.withPlane <|
178      \p ->
179      [ C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.y1 (CA.middle p.y) ]
180      , C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.x1 (CA.middle p.x) ]
181      ]
182      , C.htmlAt .max
183      .max
```


3.1 project Source Code

```
184         0
185         0
186         [ HA.style "transform" "translateX(-100%)" ]
187         [ H.span
188           [ HA.style "margin-right" "5px" ]
189           [ H.text (String.fromFloat model.percentage ++ "%") ]
190         , H.button
191           [ HE.onClick OnZoomIn
192             , HA.style "margin-right" "5px"
193           ]
194           [ H.text "+" ]
195         , H.button
196           [ HE.onClick OnZoomOut
197             , HA.style "margin-right" "5px"
198           ]
199           [ H.text "-" ]
200         , H.button
201           [ HE.onClick OnZoomReset ]
202           [ H.text "x" ]
203       ]
204   ]
205 ]
206
207
208 meta =
209   { category = "Interactivity"
210     , categoryOrder = 5
211     , name = "Zoom"
212     , description = "Add zoom effect."
213     , order = 20
214   }
215
```

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 & responsibilities
28   - Zining Wang: Full software development
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 (Current): https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data
37 ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
38     Current Appearance]]
39
40 == Motivation
41
42 Nowadays, technology is developing rapidly and the price is higher and higher if you want to get the best performance. While there are many tools showing the performance of the hardware like CPUs, the top performance is not always the best choice for everyone due to the price. Buying a suitable hardware is not focusing on one aspect but the overall balance between performance, power, price and so on. Different people may have different performance requirements preference on maybe single-core performance or multi-core performance. For GPU, one may want high encoding ability while another may want high bandwidth.
43 Besides, in some other areas like LLM, when we want to choose a model, we would need to balance the speed, size, and accuracy of the model. And as the LLM is under the development, it is unfair to compare today's LLM with the model developed one year ago. So I want to develop a tool that could let people choose the measurement dimensions they care about and compare them in a 2D and 3D plot.
44
45 == Develop Process
46
47 == Developed Part
48 #grid(columns: 2)[
49     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. Now I have developed the basic structure of the tool. In the right image, the x axis is the process Size in Nanometers, the y axis is the TDP power and the area of each point is the die size. The user could also zoom in and zoom out to see the details of the data.
50 ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
51     Current Appearance]]
52
53 == Developing Plan
54 I plan to do the following things later:
55
56 1. Support the feature that the user could choose how x-axis, y-axis, area are bind to the data and the user could filter some data out.
57 2. Support the feature that the user could upload any data and visualize it.
58 3. Support the feature that the user could choose the 3D mode and the user could choose one more axis and rotate the 3D plot. (If time allows)
59
60 #focus-slide[
```

3.2 slides Source Code

```
61 Thanks for listening
62
63 https://wznmickey.github.io/visualizeMe/
64 ]
65
66 == Appendix
67
68 == project Source Code
69 `Main.elm`
70 #{
71   set text(size: 8pt)
72   let x = read("../src/Main.elm")
73   raw(x, block: true, lang: "elm")
74 }
75 `Zoom.elm`
76 #{
77   set text(size: 8pt)
78   let x = read("../src/Zoom.elm")
79   raw(x, block: true, lang: "elm")
80 }
81
82
83 == slides Source Code
84 The typst code generating the slides.
85 `main.typ`
86 #{
87   set text(size: 8pt)
88   let x = read("../main.typ")
89   raw(x, block: true, lang: "typ")
90 }
91
```