

# Tools Seminar

## Week 9 - Visualization

Hongzheng Chen

Mar 23, 2020

1 Overview

2 Matplotlib

3 Draw.io

4 TikZ

5 Summary

1

# Overview

# Visualization\*

Visualization is used to **gain or show insights through data**

- Information visualization
  - not only statistical charts; various visualization forms help to show multi-attributes, topological structure, and complex relationships
  - actually it is a sub-topic of human-computer interaction (HCI) with top-tier conference [CHI](#)
  - good visualizations help your paper to be accepted!
- Scientific visualization
  - a sub-topic of CG
  - emphasizing on realistic renderings of volumes, surfaces, illumination sources, etc.

We will focus on information visualization

---

\*Ref: [https:](https://shellywhen.github.io/Visualization/Outline-Visualization.html#slide=3)

[//shellywhen.github.io/Visualization/Outline-Visualization.html#slide=3](https://shellywhen.github.io/Visualization/Outline-Visualization.html#slide=3) 🔍 ↻

# Catalog of Information visualization

- Tables
- Bar charts
- Flow charts
- Functions
- Graphs / Networks
- Time series
- Text
- Geo-spacial
- ...

# Tables

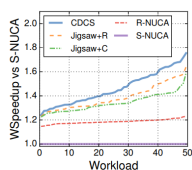
The most commonly seen data type

	Col 1	Col 2	Col 3
Row 1			
Row 2			
Row 3			

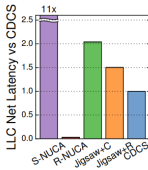
Tools: Excel, [Tableau](#),  $\text{\LaTeX}$

- One of the [2019 Turing Award](#) Winners: Pat Hanrahan (Tableau cofounder)

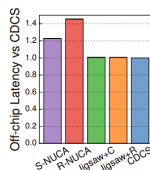
# Bar Charts



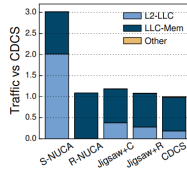
(a) Distribution of weighted speedups over S-NUCA.



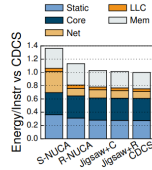
(b) Avg. on-chip latency on LLC accesses.



(c) Avg. off-chip latency (due to LLC misses).



(d) Breakdown of avg. on-chip network traffic per instruction.



(e) Breakdown of avg. energy per instruction.

Fig source: Nathan Beckmann, Po-An Tsai, Daniel Sanchez, *Scaling Distributed Cache Hierarchies through Computation and Data Co-Scheduling*, HPCA, 2015

Tools: [Matplotlib](#), [Plotly](#)

\* Pay attention to the figures when you read papers. There exists lots of details!

# Flow charts

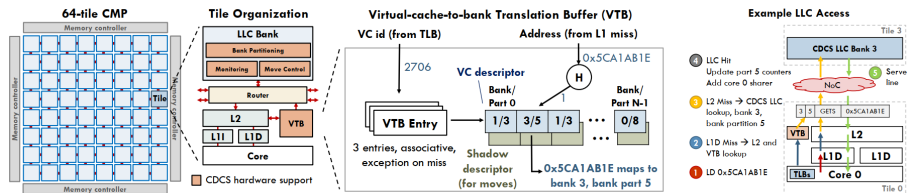
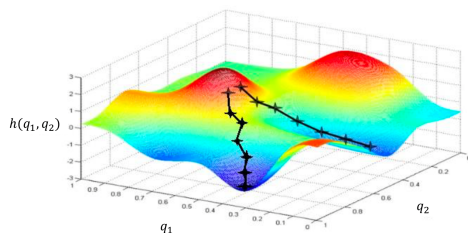
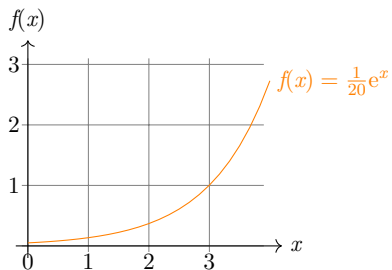


Fig source: Nathan Beckmann, Po-An Tsai, Daniel Sanchez, *Scaling Distributed Cache Hierarchies through Computation and Data Co-Scheduling*, HPCA, 2015

Tools: [Microsoft Visio](#) (enterprise version), [draw.io](#)



# Functions (2D & 3D)



Tools: [Matplotlib](#), [Mathematica](#),  $\text{\LaTeX}$  [TikZ](#)

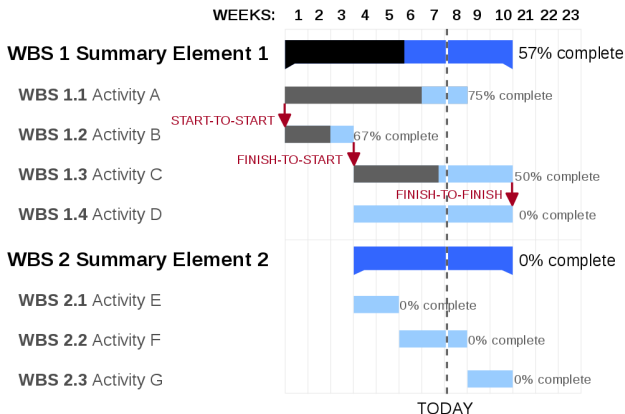
# Graphs/Networks



Fig source: <https://digi.uga.edu/network-graphs/>

- Node-link diagram, tree map, bubble chart
- Tools: [networkx](#),  $\text{\LaTeX}$  [Tikzcd](#),  $\text{\LaTeX}$  [forest](#), [Plotly](#)

# Time Series



- Line graph / Bar charts
- Gantt Chart
- Heat Map: Check your [Github](#) contribution (

## Text



Tools: [WordCloud](#), [Wordle](#), ...

2

# Matplotlib

# Matplotlib

**Matplotlib:** A Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms

```
pip install matplotlib
```

- Use Anaconda to install since the figure will pop out as a graphical window
- If you use WSL, you need to install graphical support
- **Highly recommend** to use Jupyter Notebook no matter which OS you use

\* See `matplotlib.ipynb` for demos

3

Draw.io

# Draw.io

[Draw.io](#) is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams

- Web-based
- Open-sourced
- Support  $\text{\LaTeX}$  formulas
- Export as pdf files



4

TikZ

# TikZ

**TikZ** and PGF are  $\text{\TeX}$  packages for creating graphics programmatically

- If you have installed TeXLive completely, TikZ must have been installed
- Called by `\usepackage{tikz}`
- Manual: <http://www.texample.net/media/pgf/builds/pgfmanualCVS2012-11-04.pdf>
- English tutorial from Overleaf:  
[https://www.overleaf.com/learn/latex/TikZ\\_package](https://www.overleaf.com/learn/latex/TikZ_package)
- Chinese tutorial: <https://www.latexstudio.net/archives/9774>
- Wiki book: <https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>

\* **Tikz-cd**: commutative diagrams

# TikZ Basis

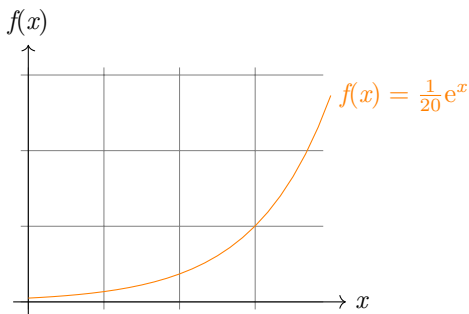
- `\begin{tikzpicture} ... \end{tikzpicture}`
- `\node [attribute] (nodelabel) at (coordinate) {texts};`
- `\draw (nodelabel) to (nodelabel);`
- Every command ends with `;`

# TikZ Example



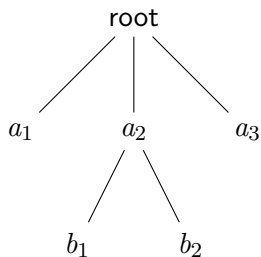
```
\begin{tikzpicture}
\node [left] (a) at (0,0) {$A$};
\node [right] (b) at (1,1) {$B$};
\draw [color=red!50,->] (a) -- (b);
\end{tikzpicture}
```

\* Use cycle to draw closed curves

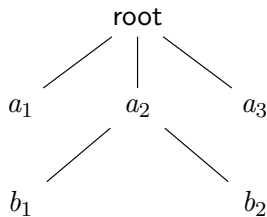


```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-0.1) grid (3.9,3.1);
\draw[>-] (-0.2,0) -- (4.2,0) node[right] {$x$};
\draw[>-] (0,-0.2) -- (0,3.4) node[above] {$f(x)$};
\draw[color=orange] plot (\x,{0.05*exp(\x)});
\inode[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}
```

# Tree



```
\begin{tikzpicture}
\node {root}
  child {node {$a_1$}}
  child {node {$a_2$}}
    child {node {$b_1$}}
    child {node {$b_2$}}
  child {node {$a_3$}};
\end{tikzpicture}
```



```
\begin{tikzcd}[column sep=scriptsize]
& \text{root} \arrow[dash,dl] \\
& \arrow[dash,d]\arrow[dash,dr] & \& \& \\
a_1 & a_2 \arrow[dash,dl] \arrow[dash,dr] & a_3 \\
b_1 & & b_2
\end{tikzcd}
```


# Control Command (Loop)



$$\int_{\Omega} x_1$$



$$\int_{\Omega} x_2$$



$$\int_{\Omega} x_3$$



$$\int_{\Omega} x_4$$



$$\int_{\Omega} x_5$$

```
\begin{tikzpicture}
[L1Node/.style={circle,
  draw=blue!50, fill=blue!20, very thick,
  minimum size=10mm},
L2Node/.style={rectangle,
  draw=green!50, fill=green!20, very thick,
  minimum size=10mm}]
\foreach \x in {1,...,5}
  \node[L1Node] (w1_\x) at (2*\x, 0){\int_{\Omega} x_\x};
\end{tikzpicture}
```

5

# Summary

# Summary

- Overview
- Matplotlib
- Draw.io
- TikZ