Applications For High Speed Data Transfer

Joshua Miller
University of Chicago

Let’s talk about data

- **Technology yields high resolution data**
- Collaboration!
- **Two options:**
  - **Compute over local data**
  - **Transfer data to compute**

Bringing compute to data

- CSists love locality
- **It's a great idea!**
- **Not always feasible**

Bringing data to compute

- Just scp it!
- **Just download it in <favorite browser>!**
- 800 GB genomic binary file
- 2 Gbps
- 55 minutes
- Great!

Bringing data to compute

- Nope…
- That's from Chicago to Chicago
- It's already local
- How about from Chicago to Amsterdam?
- 0.148 Gbps
- 12 Hours!
- Again: 1 to 12 hours

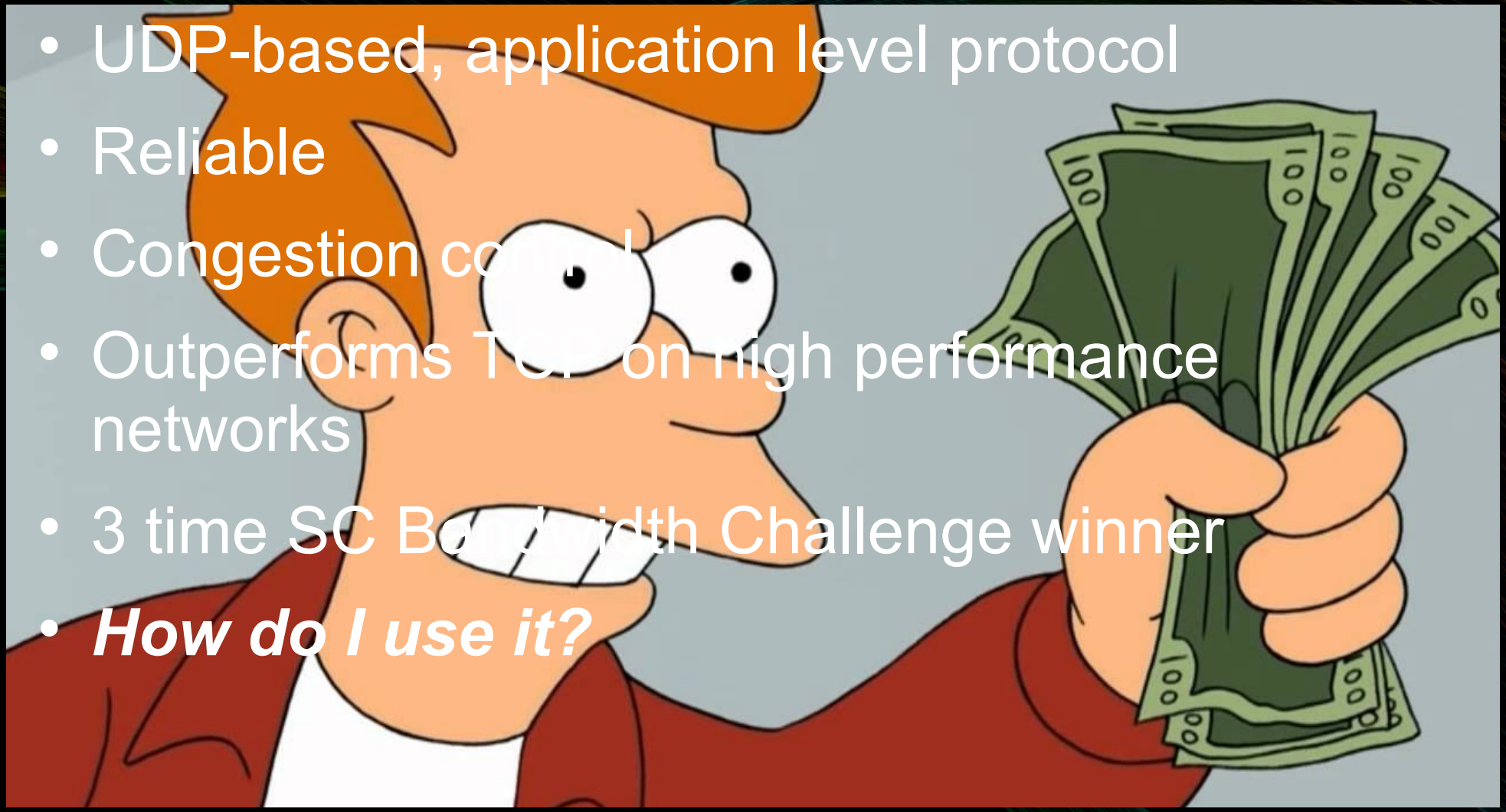
Bringing data to compute

- Why is it slow?
  - Is the network bad?
  - Do we need better networks?
  - Is the software bad?
  - Is the protocol bad?
- Let’s look at the protocol
- TCP, the *de facto* standard of the internet
- The web, scp, rsync, ssh, etc.

TCP

- Underutilizes network bandwidth over high-speed connections with long delays
- TCP is additive increase/multiplicative decrease
- System noise, packet loss, concurrent streams
- More latency means less time to respond
- Solutions:
  - Larger increase ratio (HighSpeed TCP)
  - Hardware updates (Router feedback, etc.)
  - Timer-based acknowledgment (UDT)

UDT



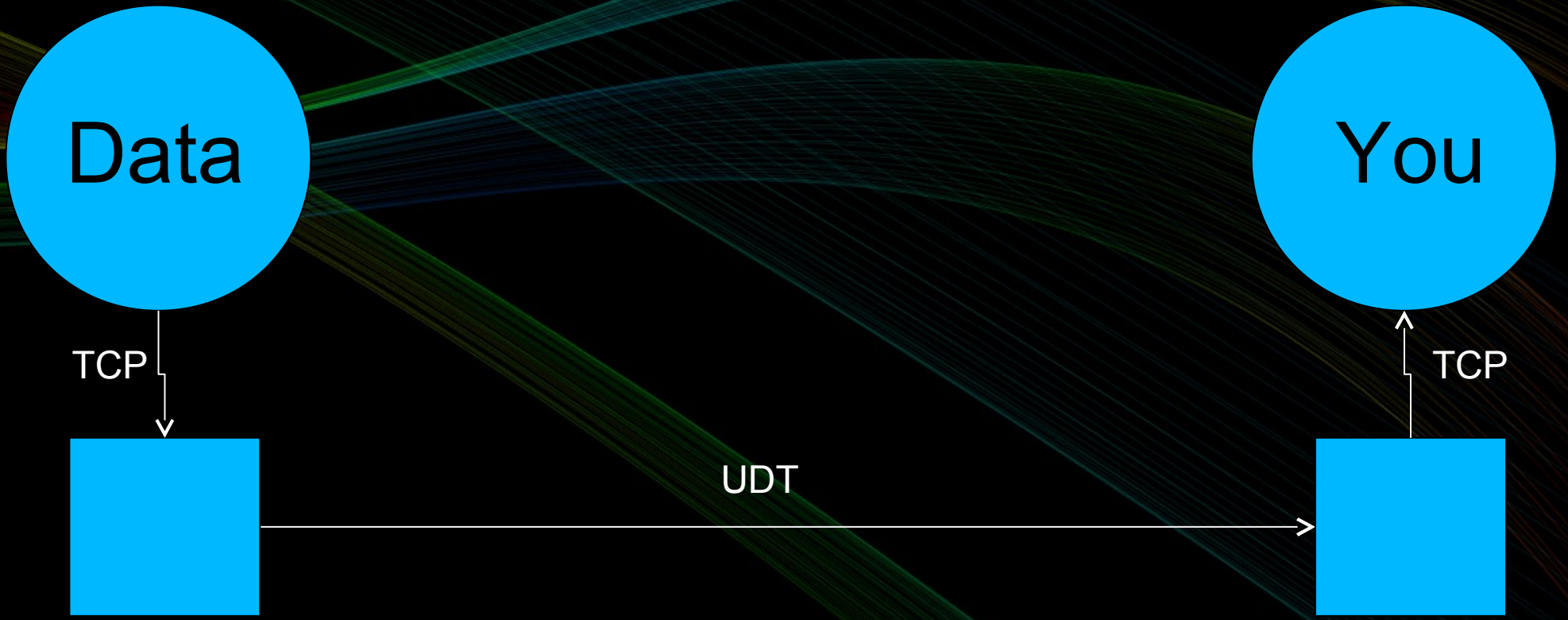
- UDP-based, application level protocol
- Reliable
- Congestion control
- Outperforms TCP on high performance networks
- 3 time SC Bandwidth Challenge winner
- ***How do I use it?***

UDT Applications

- UDR
  - Rsync ported to UDT
- Udpipe
  - netcat ported to udt
- https://github.com/LabAdvComp
- Parcel
  - UDT Proxy

Parcel

Parcel



Using the UDT proxy

Server @ host1 port 9000

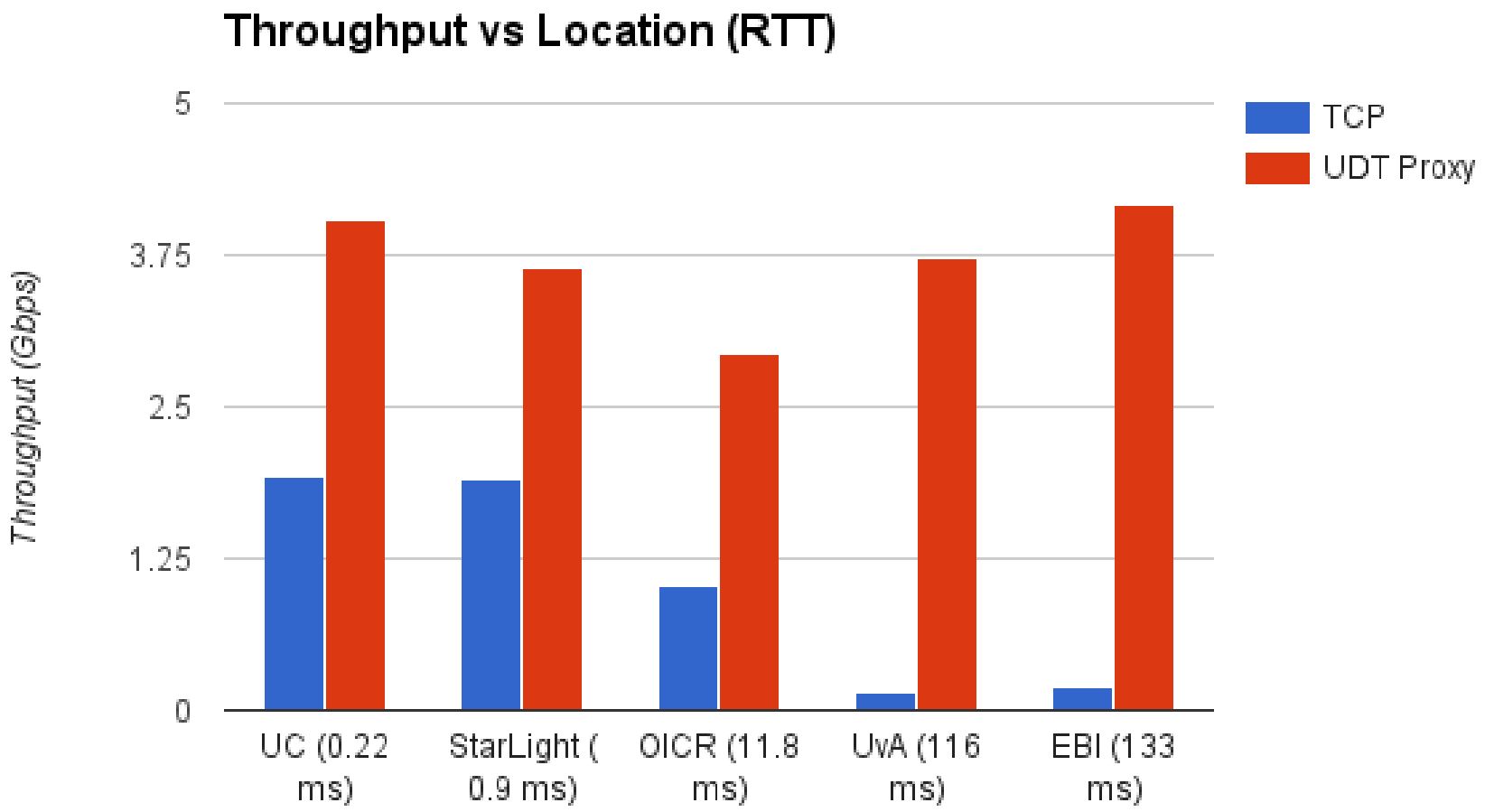
parcel-udt2tcp host1:9000 # @ host2 port 9000

parcel-tcp2udt host2:9000 # @ host2 port 9000

Client pointed toward localhost, port 9000

Bringing data to compute

- *Nope…*
- *That's from Chicago to Chicago*
- *It's already local*
- *How about from Chicago to Amsterdam?*
- *0.148 Gbps* → 3.72 Gbps
- *12 Hours!* → 28 minutes
- *Again: 1 to 12 hours*


Throughput vs Location (RTT)
Throughput (Gbps)
5
3.75
2.5
1.25
0
TCP
UDT Proxy
UC (0.22 ms)
StarLight (0.9 ms)
OICR (11.8 ms)
UvA (116 ms)
EBI (133 ms)

Bringing data to compute

- What can you do with these tools?,
  - Use UDR
  - Use Parcel proxy as a stand alone layer
  - Integrate Parcel, it's python (bound to C++)

Applications For High Speed Data Transfer

Joshua Miller
University of Chicago