Modeling the Formation of Rain

in Hi-Performance Erlang

John Tyree
Universiteit van Amsterdam

Project Goals

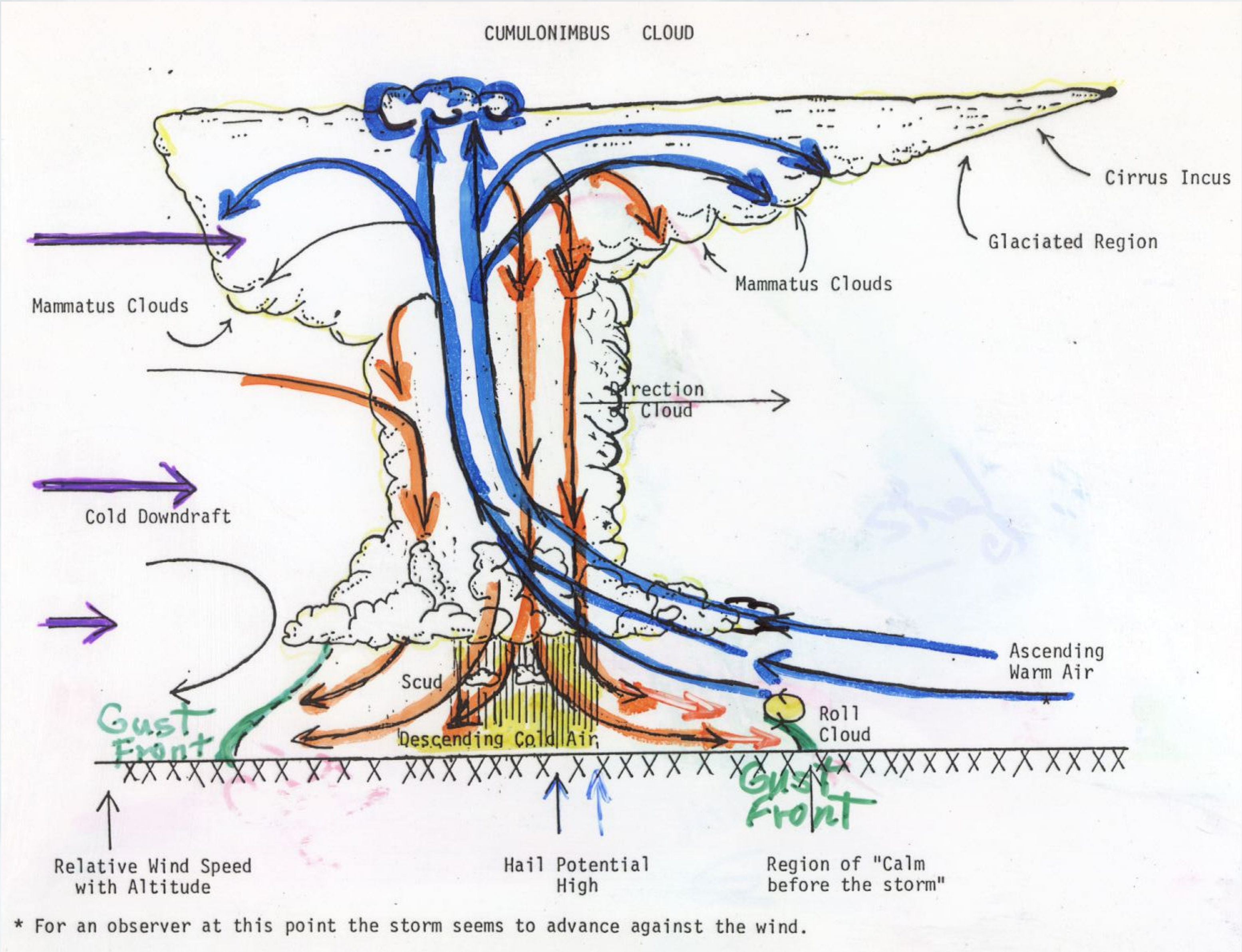
- Believably simulate rain formation
  - Does it require high humidity?
  - High wind?
  - Huge spaces?
- Gain experience in functional programming
  - Erlang and HiPE
  - Is it suited for what we do?

Nascent clouds

- Vapor
- Condensation nuclei
  - Air temp @ dewpoint.
  - Salts, pollutants, ash
  - 0.001 - 0.05 mm diameter[1]
  - Negligible gravitational effects

- [Coalescence -> Drop Formation] -> Rain
  - Upper limit: ~4mm

Source: http://ga.water.usgs.gov



Rain formation in the lower atmosphere.

http://www.ndsu.edu/ndsco/education/thunderstorms/images/CumulusCloud.jpg

How do we model this?

- Domain representation:
  - Synchronized "cellular semi-automata"
  - Multi-scale: Drops < - > Clouds
- Apply expert rules based on both large scale meteorology and small scale physics of water droplet interaction

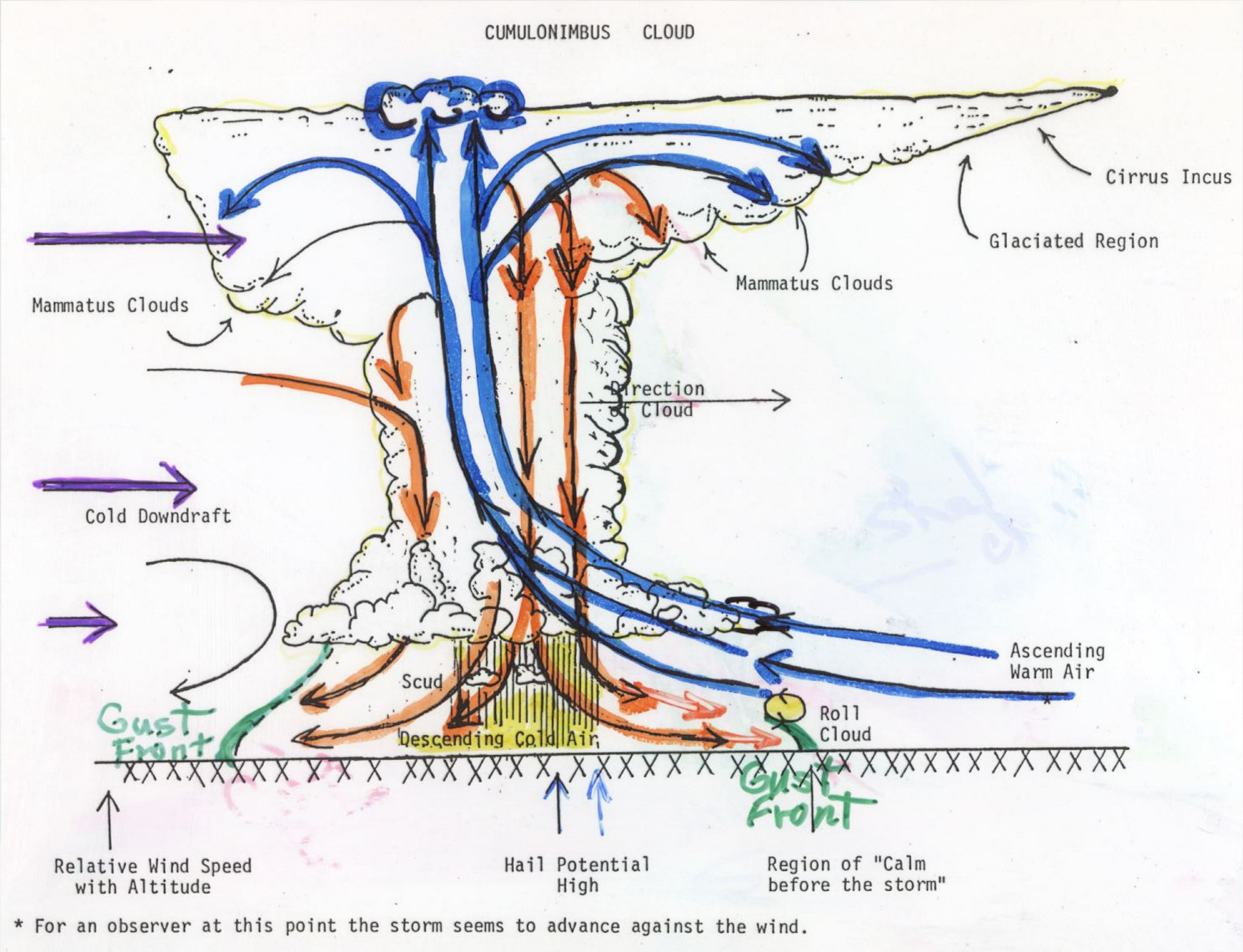
How do we model this?

Large Scale

- Updraft wind force
- Lateral wind force
- Gravitational force
- Relative humidity effects

Small Scale

- Coalescence frequency
- Collision detection
- Terminal Velocity
- Splitting



Rain formation in the lower atmosphere.

http://www.ndsu.edu/ndsco/education/thunderstorms/images/CumulusCloud.jpg

Large Scale

Wind conditions:

- DY = Gravitational Movement - Partially stochastic updraft
- DX = Partially stochastic wind speed.

Humidity Conditions:

- "Vapor" continuously added
  - Relative humidity determines density
  - Size range 0.001mm, 0.05mm. ("Rain" at 0.5mm)

Source: Intro. To Computational Science

Small Scale

Coalescence frequency:

- Difficult to solve numerically
- Used values from literature:
  - Small drops have high rates
  - High size ratios have high rates
  - Very large drops often disintegrate

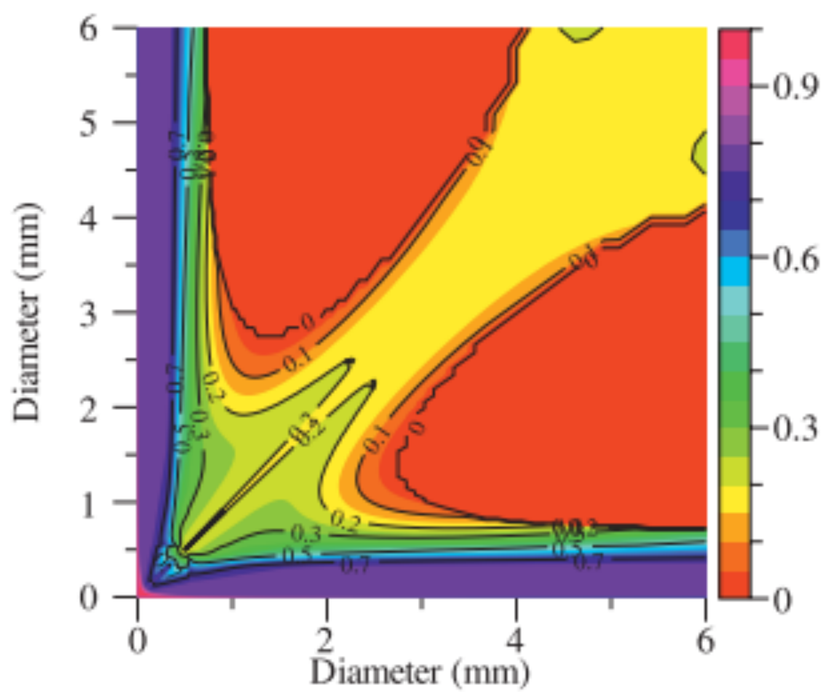


FIG. 1. Coalescence efficiencies determined from Eq. (16), which relies on efficiencies from BO95 for $d_s < 0.3$ mm, from LL82a for $d_s > 0.5$ mm, and an interpolation formula from S05 between the two. Efficiencies were obtained at a temperature of 20°C and pressure of 1013 hPa.

Source: Jacobson (2010)

Small Scale

Collision detection

Euclidean distance:

$$E = \frac{d^2}{(r_1 + r_2)^2}$$

where

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

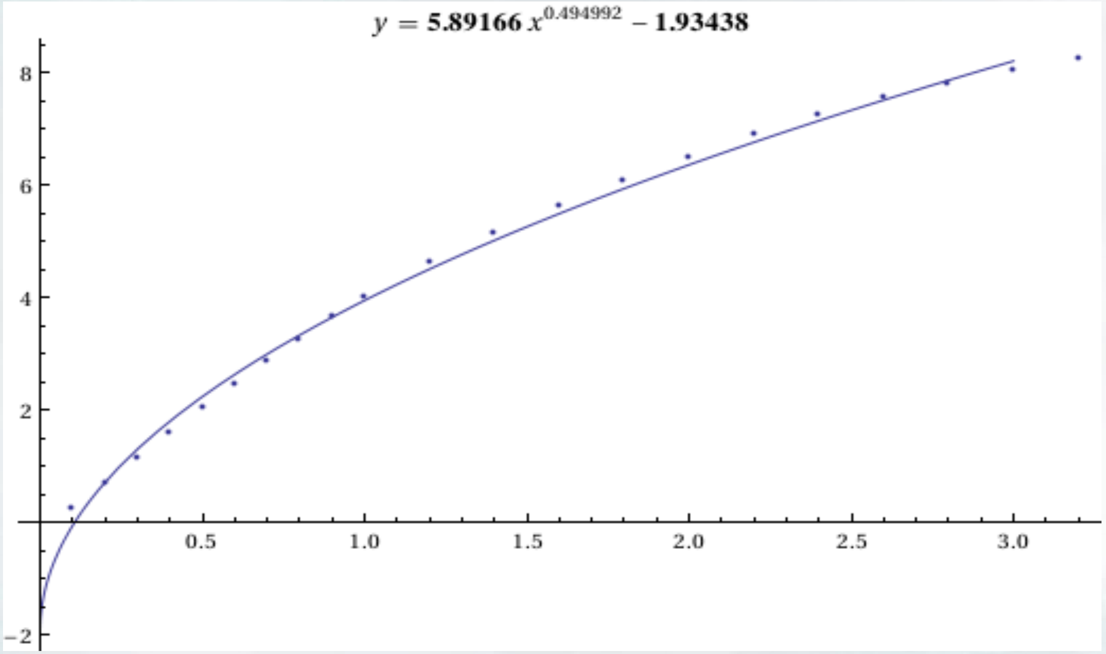
Source: Intro. To Computational Science

Small Scale

Diameter (mm) Velocity (m/s) 0.1 0.27 0.2 0.72 0.3 1.17 0.4 1.62 0.5 2.06 0.6 2.47 0.7 2.87 0.8 3.27 0.9 3.67 1.0 4.03 1.2 4.64 1.4 5.17 1.6 5.65 1.8 6.09 2.0 6.49 2.2 6.90 2.4 7.27 2.6 7.57 2.8 7.82 3.0 8.06 3.2 8.26

Terminal Velocity:

- From literature:
  - 0.05 mm -> 0.07 m/s
  - 0.2 mm -> 0.7 m/s
  - 1 mm -> 4.0 m/s
- Interpolation between values



Source: Gunn and Kinzer (1949)

Raindrops != Teardrops

Air Resistance vs Cohesion

Source: http://ga.water.usgs.gov/edu/raindropshape.html

Simulation implementation

Environment:

- Determine values from literature:
    - Relative Humidity: 1
    - Updraft Speed:
        - Minimum: 0
        - Maximum: >100 km/h
    - Coalescence Efficiency (previously shown)
    - Split frequency distribution:
        - 1.5mm -> approx. 0.5
        - 3mm -> approx. 1
    - Vapor drop size
        - Normally distributed
        - Mean -> 0.025
        - Stddev -> 0.006

Source: http://ncar.ucar.edu/learn-more-about/weather

So is it going to rain or not?

Simulated Clouds

Ratio of lateral wind speed to relative humidity

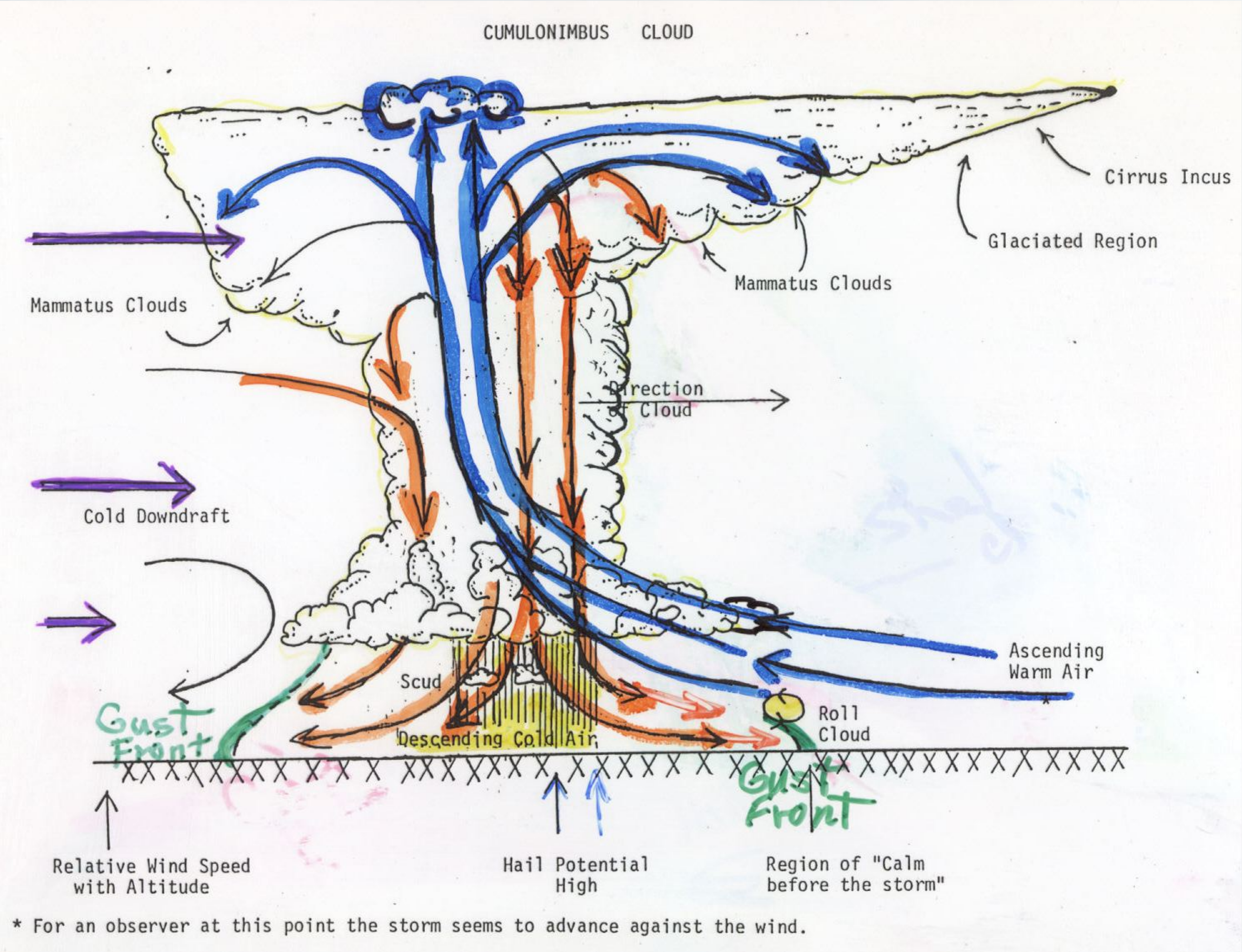
High　　　　　　　　　　　　　　　　Low

No rain -> Light rain -> Medium rain -> Cats and Dogs

Low
(or too high)　　　　　　　　　　　　High

Strength of updraft

Of course dependent on domain size.



Rain formation in the lower atmosphere.

http://www.ndsu.edu/ndsco/education/thunderstorms/images/CumulusCloud.jpg

Simulated Clouds

Increasing Storm Strength

Lateral Wind (m/s)	5	1	1	1	0.5	0
Relative Humidity	1	0.1	0.5	1	1	1
Updraft (m/s)	2	2	2	2	2	5
Max Drop Radius (mm)	0.4	0.54	0.9	1.2	1.5	2
Avg Max Drop Radius (mm)	0.25	0.3	0.65	0.9	1.1	1.5

Not Rain

Rain

Simulation implementation

Problems and work-arounds:

- Clouds are absolutely enormous compared to drops
    - Scale movement down (this was a challenge)
    - Possibly "overlapping" drops to increase drop count
    - Scale lateral wind down significantly
        - What about updraft cycles?

- Erlang doesn't use enormous O(1) access arrays
    - Store drops as a hash table.
    - Aggravates size problem even further.

References

- National Center for Atmospheric Research (2008). "Hail". University Corporation for Atmospheric Research.
- "Numerical Solution to Drop Coalescence/Breakup with a Volume-Conserving, Positive-Definite, and Unconditionally Stable Scheme", M. Z. JACOBSON, Jour. of Atmos. Science 2010
- "The Development of the HiPE system: design and experience report", http://user.it.uu.se/~kostis/Papers/hipe-sttt.pdf

What is Erlang?

- Functional
  - Declarative, not Imperative
- Distributed
  - Abstract processes
  - Message Passing
- High Availability
  - Process monitoring
  - Supervision trees
- Massively Scalable
  - Architecture agnostic
  - VM -> Nodes -> LWPs

Sounds great! Right?

What is Erlang?

- Cross-Platform
  - BEAM VM
- Light Weight Processes (LWP)
  - New processes cost approx. 300 words
  - Can easily number in the tens of thousands
  - Cheap messaging
  - Single context

Sounds great! Right?

What is Erlang?

Answer:

SLOW

But wait!

What is HiPE?

Hi-performance Erlang!

- Native compilation.
- 5-10 x speedup for number intensive workloads.
- Can beat python in some cases

But wait again!

What is HiPE?

Hi-performance Erlang!

STILL PRETTY SLOW

What is HiPE?

Hi-performance Erlang!

Avoid

QUESTIONS?