Building a Real Workflow

Thursday morning, 9:00 am

Lauren Michael <lmichael@wisc.edu>
Research Computing Facilitator
University of Wisconsin - Madison

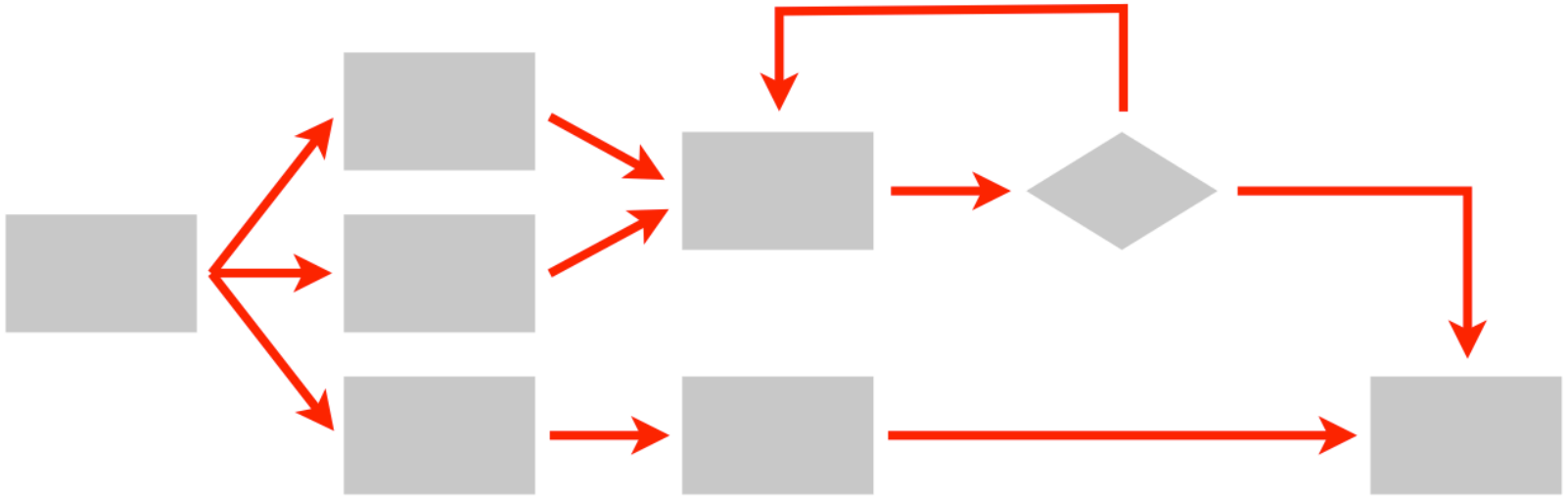
Workflows *Should* Make ~~Life~~ Science Easier

- non-computing "workflows" are all around you … especially in science
  - grading exams
  - instrument setup
  - experimental procedures

- when planned/documented, workflows help with:
  - organizing and managing processes
  - saving time with **automation**
  - objectivity, reliability, and reproducibility
    (THE TENENTS OF GOOD SCIENCE!)

Workflows are like Computing Algorithms

- Steps
- Connections
- (Metadata)

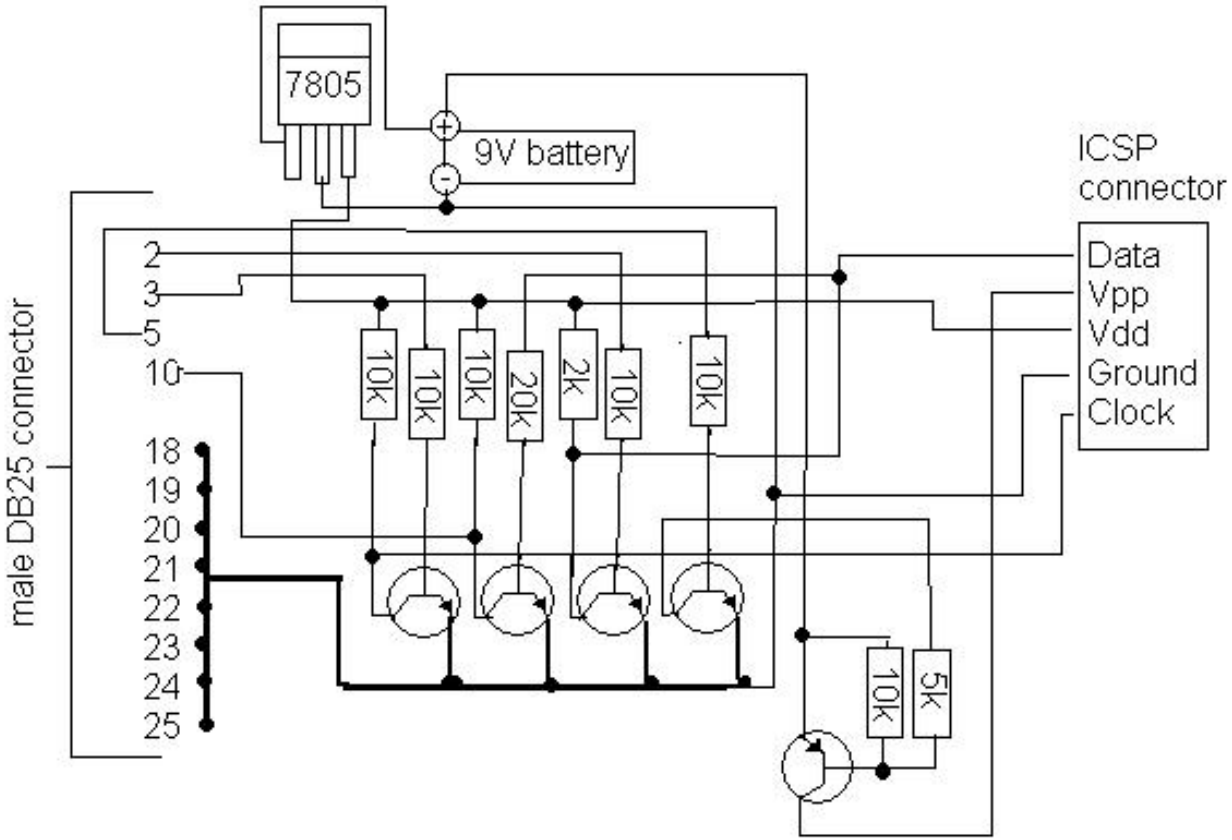


'Engineering' a Good Workflow

1. Draw out the *general* workflow
2. Define details (test 'pieces' with HTCondor jobs)
   - divide or consolidate 'pieces'
   - off-load file transfers and consider file transfer times
   - identify steps to be automated or checked
3. Build it piece-by-piece; test and optimize
4. Scale-up: data and computing resources
5. What more can you automate or error-check?

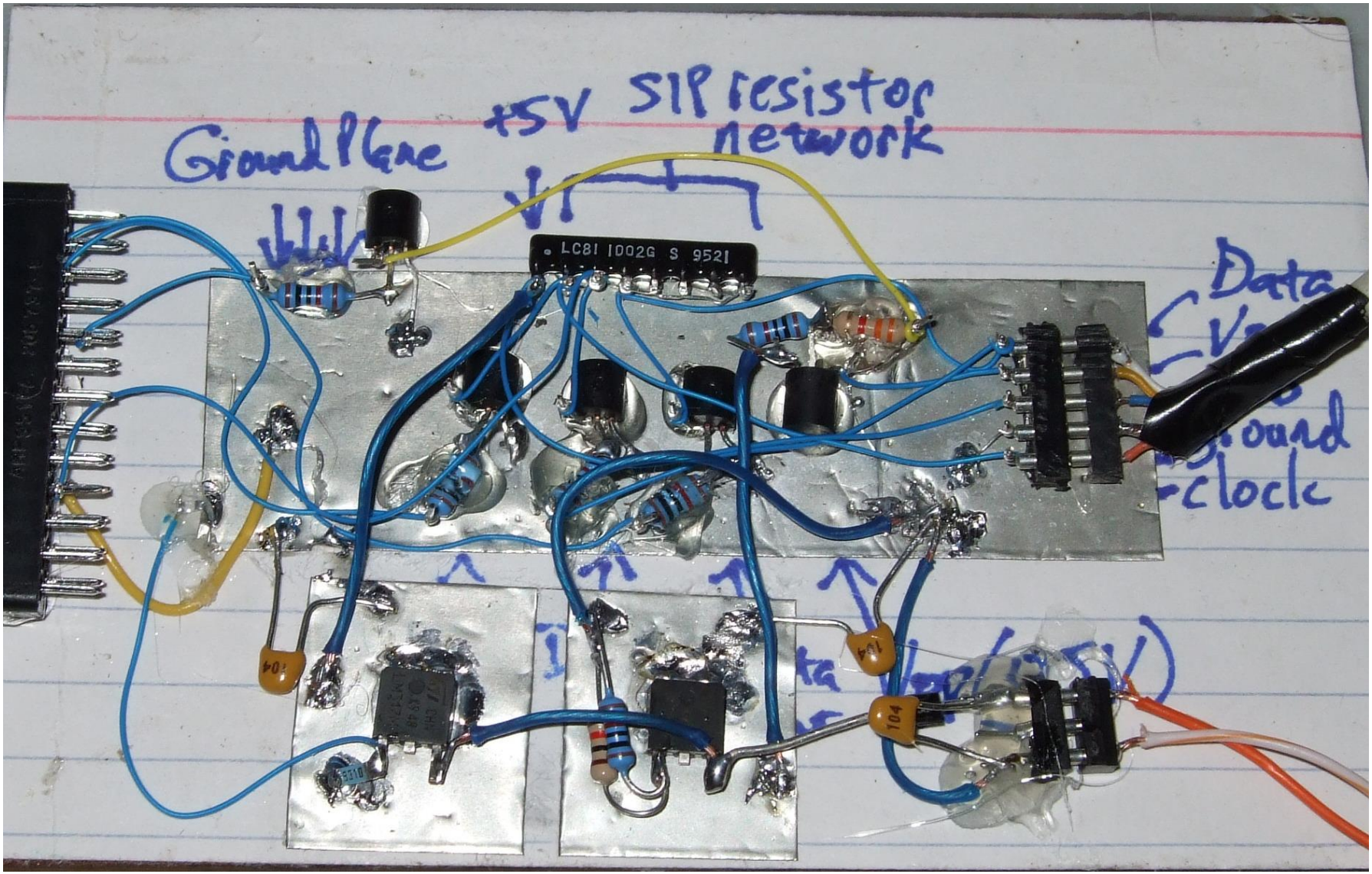
(And remember to document)

From schematics…



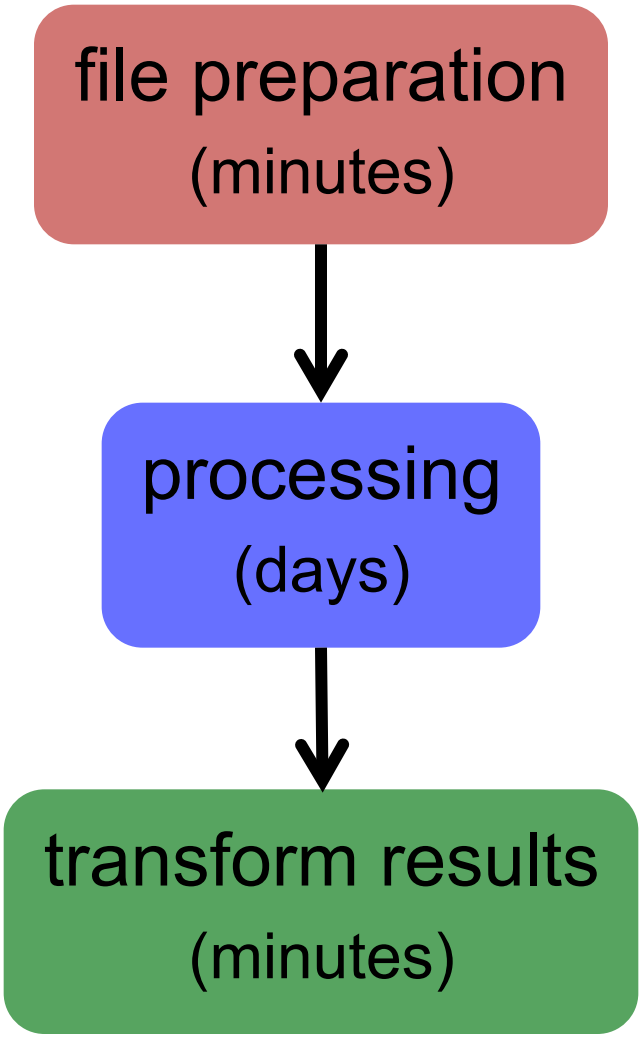
Open Science Grid

… to the real world

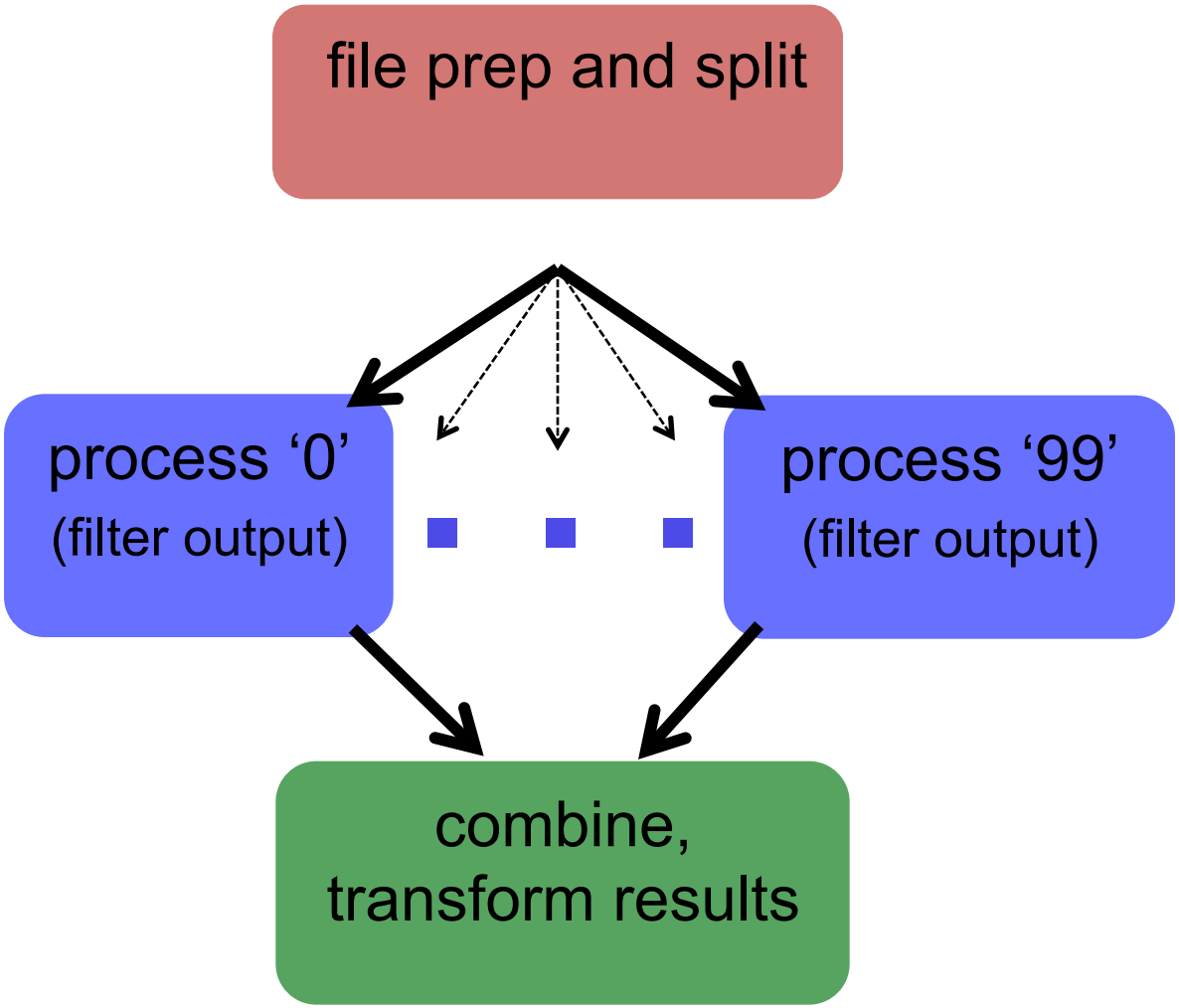


Open Science Grid

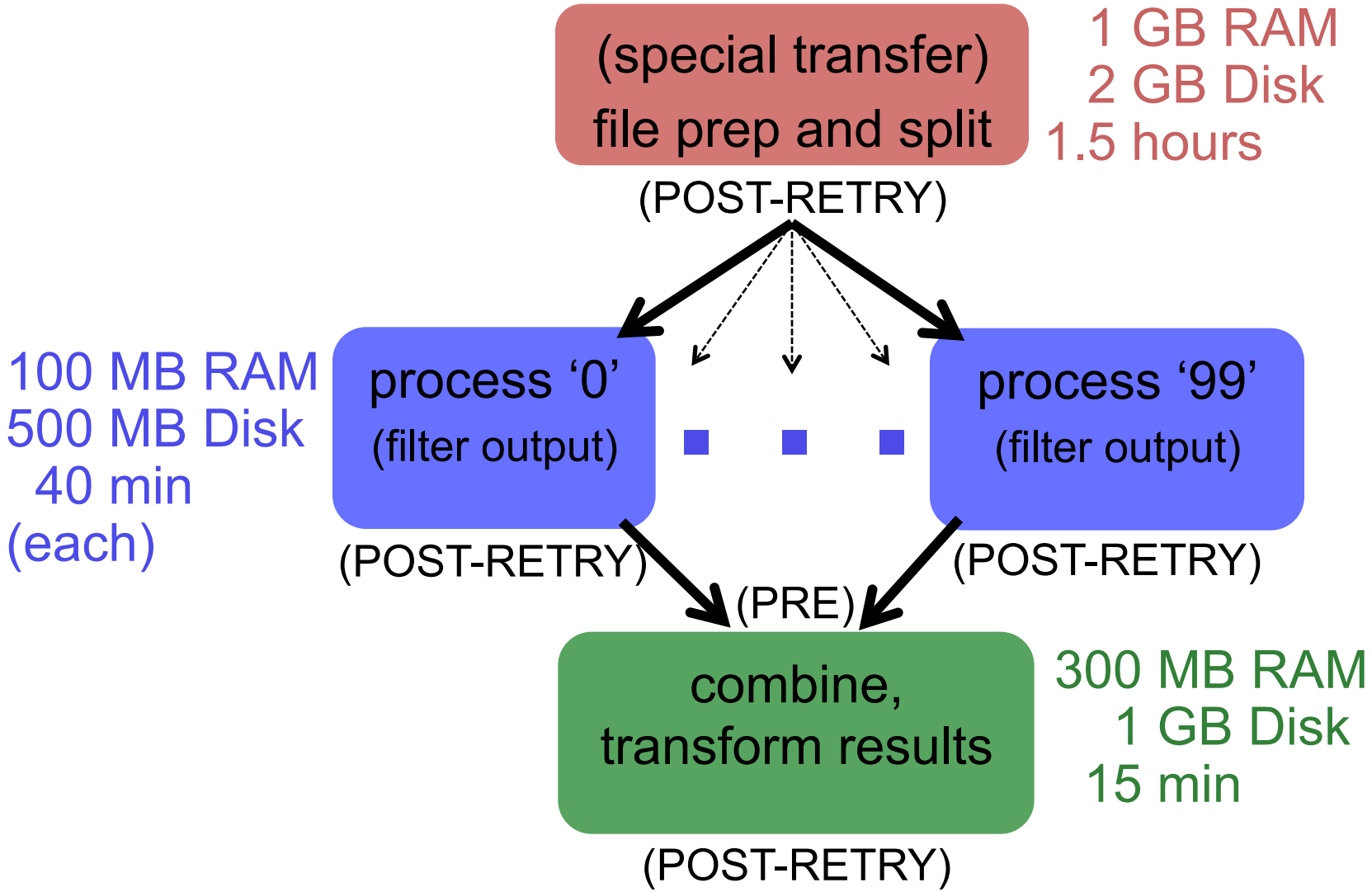
Start with This



Parallelize with HTC Splitting



End Up with This



Open Science Grid

Key HTC Principles

1. Increase Throughput
2. Be Kind to Your Submit Node
3. Bring it With You
4. ‘Scriptify’ As Much As Possible
5. “Testing, testing, 1, 2, 3 …”

Always focus on Throughput

What is High Throughput

- many ‘smaller’ jobs
- persistent job pressure
- automation
- optimizing total workflow times

What is not?

- job runtimes **less than 5 min**
- micro-optimizations

'Engineering' a Good Workflow

1. Draw out the *general* workflow
2. Define details (test 'pieces' with HTCondor jobs)
   - divide or consolidate 'pieces'
   - off-load file transfers and consider file transfer times
   - identify steps to be automated or checked
3. Build it piece-by-piece; test and optimize
4. Scale-up: data and computing resources
5. What more can you automate or error-check?

(And remember to document)

Open Science Grid

Resources Jobs Need

- CPU
  - #CPUs and time
- RAM
- Disk
  - Working (execute side)
  - Total (submit side)
  - Compute bandwidth (file transfer)
- Network bandwidth
  - Usually for file transfer only

First run jobs locally: To measure usage

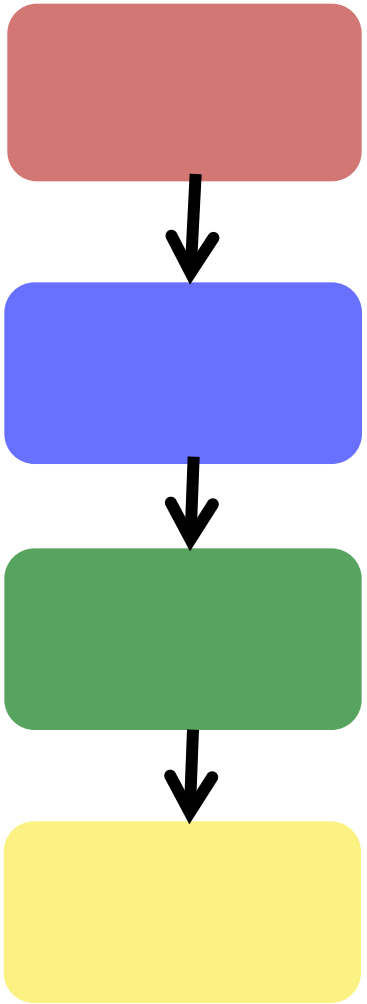
- Did it run correctly?
  - Are you sure?
- Run once remotely
  - (on execute machine, not submit machine)!
- Once working, run a couple of times
- If big variance in resource needs, should you take the…
  - Average? Median? Worst case?

User Log shows all

```
005 (2576205.000.000) 06/07 14:12:55 Job terminated.
        (1) Normal termination (return value 0)
                Usr 0 00:00:00, Sys 0 00:00:00  -  Run Remote Usage
                Usr 0 00:00:00, Sys 0 00:00:00  -  Run Local Usage
                Usr 0 00:00:00, Sys 0 00:00:00  -  Total Remote Usage
                Usr 0 00:00:00, Sys 0 00:00:00  -  Total Local Usage
        5  -  Run Bytes Sent By Job
        104857640  -  Run Bytes Received By Job
        5  -  Total Bytes Sent By Job
        104857640  -  Total Bytes Received By Job
        Partitionable Resources :    Usage  Request
           Cpus                 :                1
           Disk (KB)            :   122358   125000
           Memory (MB)          :       30      100
```

Golden Rules for DAGs

- Beware of the shish kebab!
  - (self-checkpointing, next lecture)
- Use PRE and POST script generously
- RETRY is your friend
- DAGs of DAGs are good
  - SPLICE
  - SUB_DAG_EXTERNAL



Open Science Grid

Wrapper Scripts are Essential

- Before execution (bring it with you!)
  - transfer/prepare files and directories
  - setup/configure environment and other dependencies
    - including run-time libraries (Matlab, R, Python, etc.)
- Execution
  - prepare complex command-line arguments
  - batch together many ‘small’ tasks
- After execution
  - filter, divide, consolidate, and/or compress files
  - check for errors

Open Science Grid

Extra DAG tips

- PRE and POST scripts run on the submit node
  - avoid combining/splitting large files
  - avoid compiling

- Remember: DAGs don't do loops well

Solution: move more tasks into a 'job'

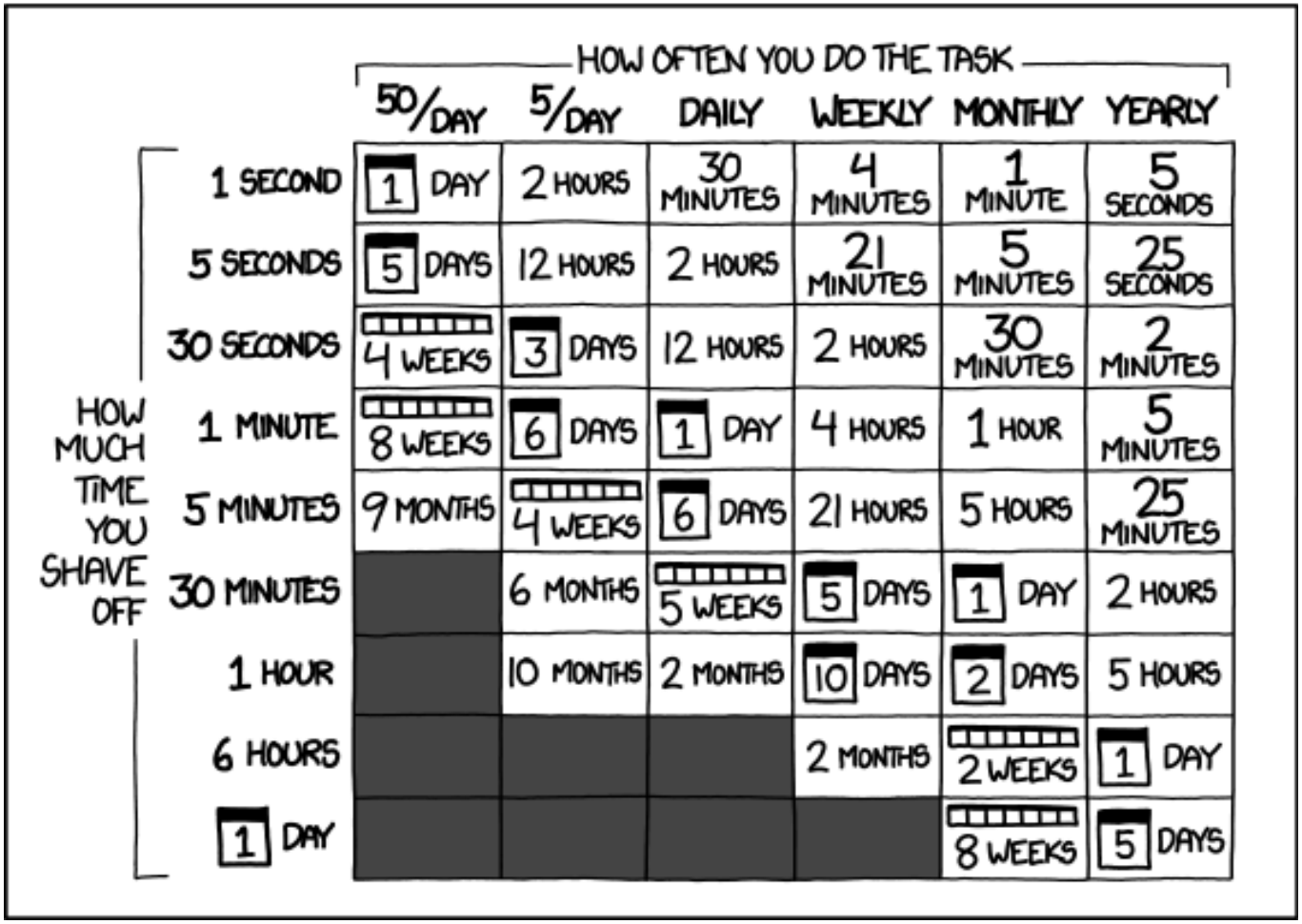
Automate *All* The Things

- well, not really, but kind of …
- Really: What is the minimal number of manual steps necessary?

  even 1 might be too many; zero is perfect!

- Consider what you get out of automation
  - time savings (including less 'babysitting' time)
  - reliability and reproducibility

Is It Worth the Time?



http://xkcd.com/1205/

Open Science Grid

Batching (Merging) is easy

- Scripting
  - Avoids transfer of intermediate files
  - Debugging can be a bit tricky without scripted error reporting

Open Science Grid

Breaking up is hard to do…

- Ideally into parallel (separate) jobs
  - reduced job requirements = more matches
  - not always possible
- Often need checkpoints
  - *standard* universe can help
  - user-defined check-pointing
  - checkpoint images can be hard to manage

Automation: Parameter Sweeps

Command arguments can become complicated and messy.

- Wrapper scripts could:
  - Hardcode “extra” arguments
  - Compute arguments
  - Look up arguments from a table

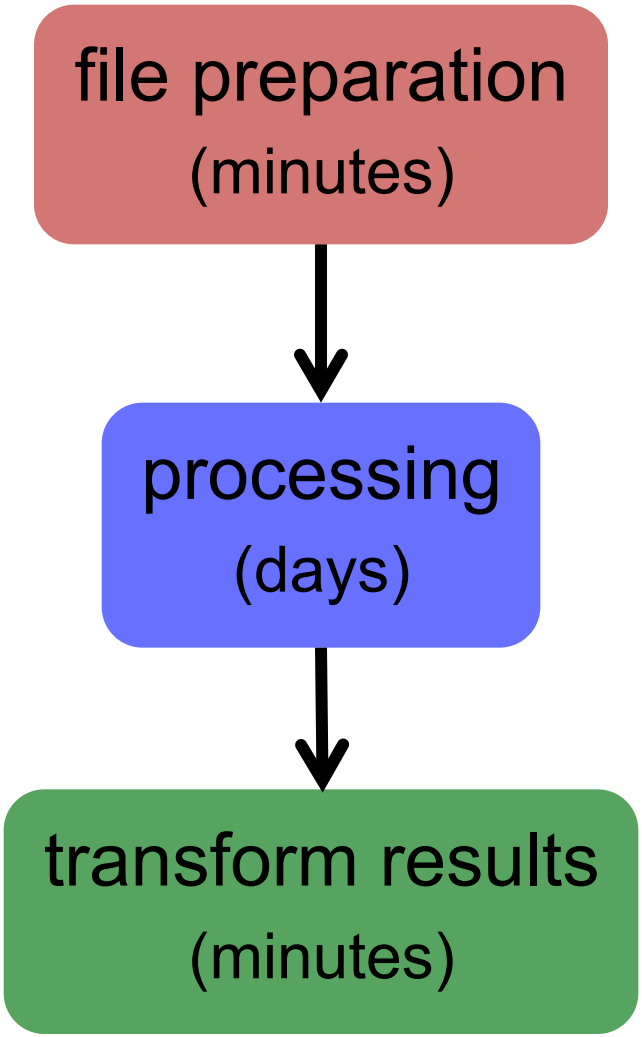
'Engineering' a Good Workflow

1. Draw out the *general* workflow
2. Define details (test 'pieces' with HTCondor jobs)
   - divide or consolidate 'pieces'
   - off-load file transfers and consider file transfer times
   - identify steps to be automated or checked
3. Build it piece-by-piece; test and optimize
4. Scale-up: data and computing resources
5. What more can you automate or error-check?

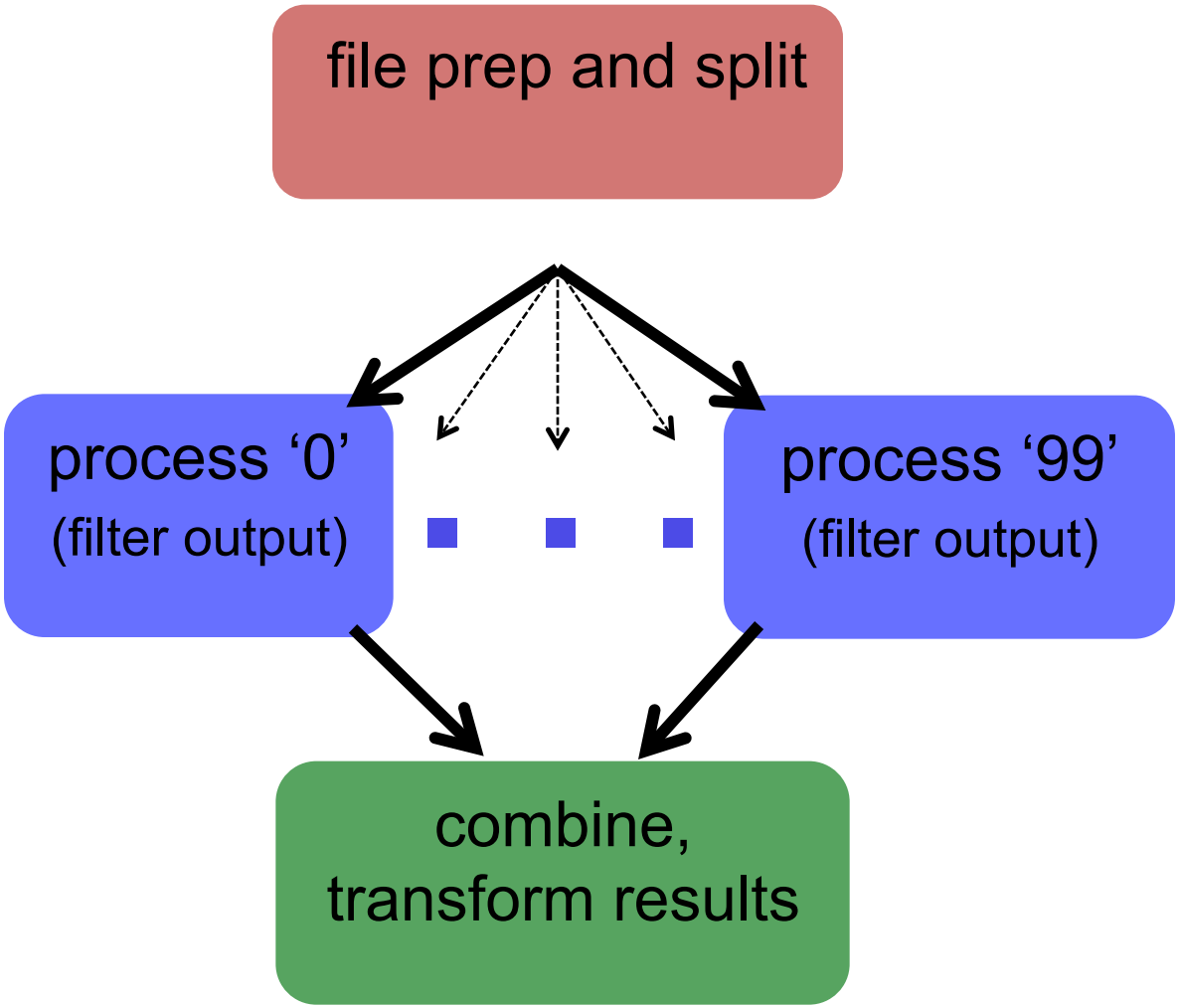
(And remember to document)

Open Science Grid

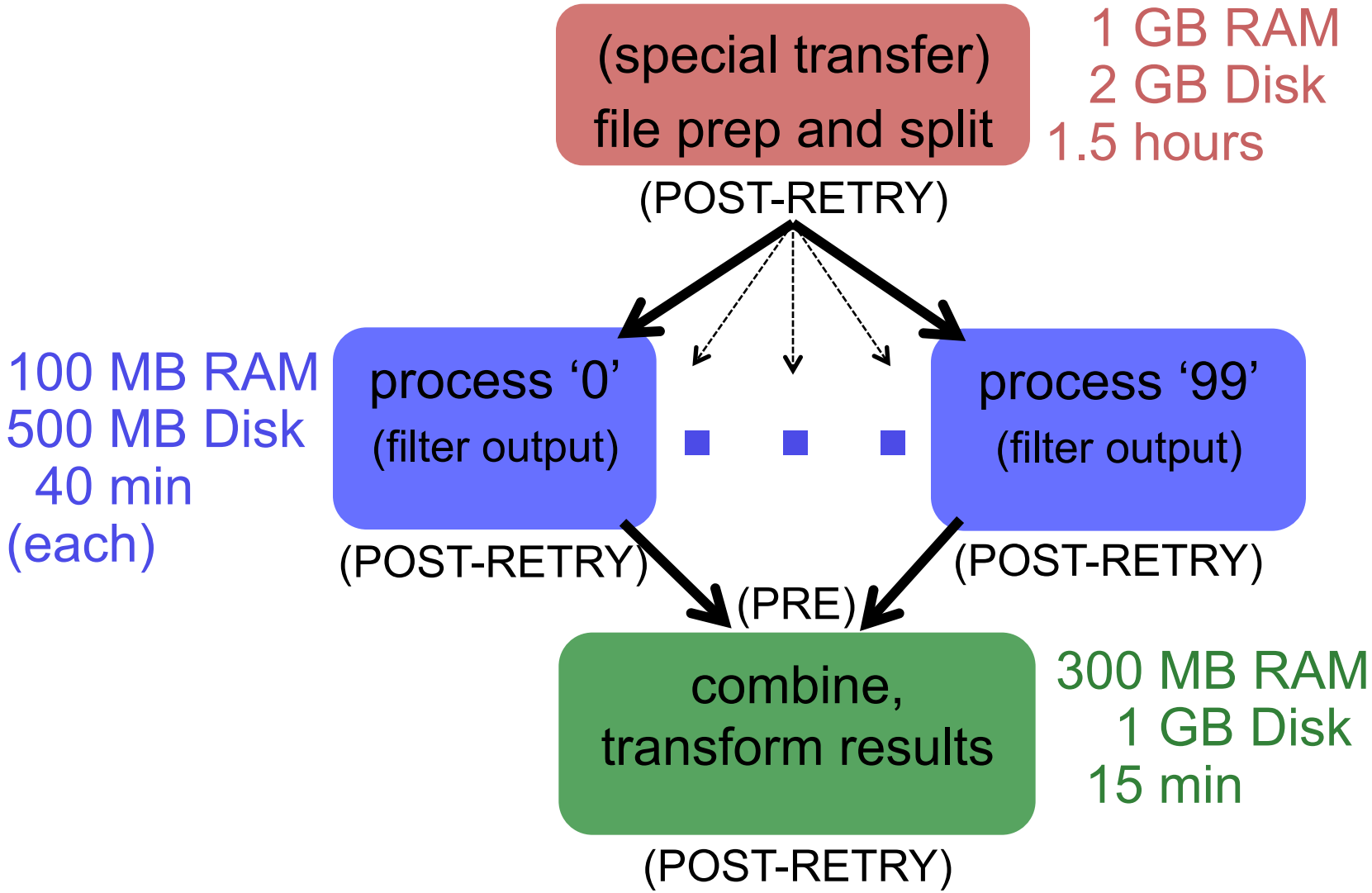
Start with This



Exercise 1


file prep and split
process ‘0’
(filter output)
process ‘99’
(filter output)
combine,
transform results

Exercise 2


(special transfer)
file prep and split
1 GB RAM
2 GB Disk
1.5 hours
(POST-RETRY)
100 MB RAM
500 MB Disk
40 min
(each)
process '0'
(filter output)
process '99'
(filter output)
(POST-RETRY)
(POST-RETRY)
(PRE)
combine,
transform results
300 MB RAM
1 GB Disk
15 min
(POST-RETRY)

Open Science Grid

Questions?

- Feel free to contact me:
  - lmichael@wisc.edu
- Now: “Joe’s Workflow” Exercise 6.1
  - 9:30-10am, in groups
- Later:
  - 10-10:30am: From Workflow to Production
  - 10:30-10:45am: Break
  - 10:45am-12:15: Exercises 6.2, 6.3