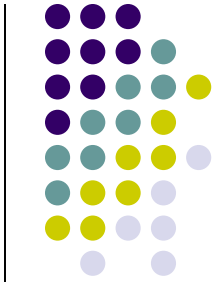
Huge Data is outgrowing the Internet's file transfer protocols

Prof. Craig Partridge
Colorado State University

Joint thinking with Prof. Susmit Shannigrahi of Tenn Tech

Keynote: NSF Huge Data Workshop, April 2020

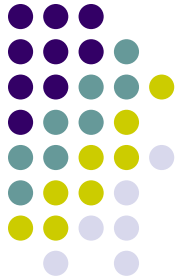
Roughly 1 in every 121 huge file transfer delivers bad data



Liu et al, *HPDC '18* found that about 1 in every 121 FTPs of large data delivered a file that FTP said was OK, but a message digest computed over the file showed was not an accurate copy of the original file

This was using Globus FTP, which enhances FTP to compute and check a message digest over the file.

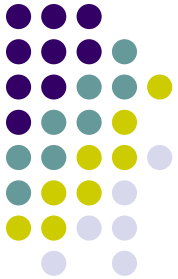
What Could Be Causing That Level of Errors?



- Work 20 years ago showed that most end-to-end errors were in hosts, routers, and middleboxes
- On some of those errors, the TCP checksum was not very effective
- A new wrinkle: the checksum is right but data is bad
  - Recent unpublished work suggests middleboxes no longer incrementally update the checksum but rather just recompute it – so they give a good checksum to packets they've trashed!

Sources: Stone & Partridge, *SIGCOMM 2000*; Stone, Hughes, Partridge, *SIGCOMM 1995*; Jan Rüth, private note

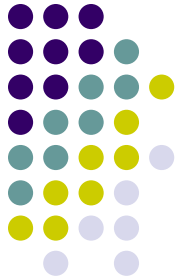
Errors, cont.



- There's also reason to believe link layer errors may be creeping through
- CRC-32 is excellent
  - Catches any one error < 32 bits and any single 2-bit error within 2048 bits
- But CRC-32 may be overwhelmed with errors
  - One study suggests as WiFi data rates increase, the error rates jump substantially (as high as 34%)

Source: Feher, *Access Networks*, 2011.

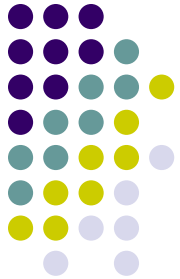
Est. 5B-10B Large Data Downloads/year



- This is a handwaving estimate, based on more narrow studies of specific environments
  - CERN transfers 1.1Billion files/year
- Growing exponentially

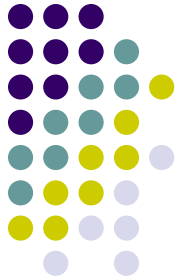
Source: https://home.cern/news/news/computing/lhc-pushing-computing-limits

Only about half of file transfers at DoE use Globus



- Regular FTP, scp and http[s] also common
- Plethora of other applications
  - FDT, Aspera, Fcache
- Implications….
- As much as 40M bad files, delivered as “good” and undetected per year!
  - 10B × 50% not caught by Globus × 1/121

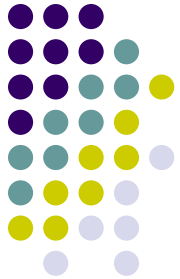
That Many Bad Files? Really?



- Our guess is that the number is lower

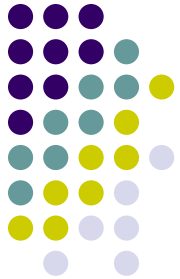
- But that's only because the scientific community has been doing a lot to double check their data
  - Computing message digests on files if Globus doesn't
  - Double checking copies by copying multiple times

Copying Multiple Times?!?

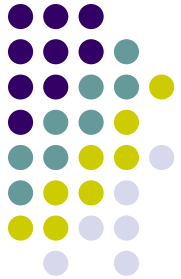


- Yep!
- And there's a preference to bypass replicated copies to get the "authoritative" copy…
- Undoing replication systems because they don't trust copies

What Does This Mean for Huge Data?

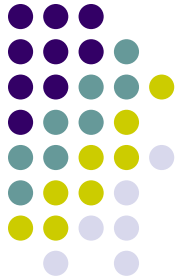


- We have file transfer protocols delivering bad files
- As a result, the scientists are
  - Copying multiple times (consuming large amounts of bandwidth)
  - Doing large file transfers, realizing the file is bad, and throwing it away (can't do incremental updates)
  - Avoiding replication and caching systems (which also makes it hard to better use bandwidth)
  - Possibly utilizing bad data unknowingly (with consequences for big science)



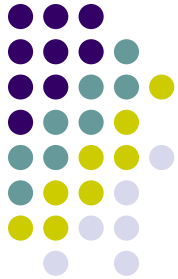
How Might We Move Forward?

For the Next Couple of Years



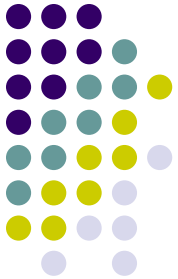
- Use message digests on files!
- But 32-bit message digests (ala Globus) will stop protecting us shortly
  - 1 bad file in every 121 × 2^32 message digest = 1 in 53B transfers… close to the level we're at
- We could use a bigger message digest but that's a mistake (see a few slides down)

Create a Next Gen FTP



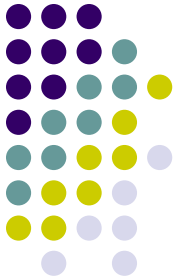
- Message checksums on files
  - Both total file and increments
- Better checkpointing
  - Support incremental repair of files during transfer (don't throw a bad file away, fix it!)
  - Allow copying from multiple replicated locations concurrently (performance)
- Ability to check against authoritative copy w/o copying
  - Scientists want an authoritative validity check

Why Message Checksums?



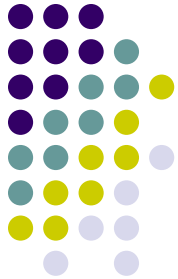
- Digests
  - Are expensive to compute (bad idea for huge data)
  - Have poor error detection properties (simply 1 in 2^x, where x is digest size)
- Checksums
  - Are fast to compute
  - If you know the error patterns, can be 100% effective
  - Match digest error detection on unknown error patterns (2^x)

Networking last looked deeply at checksums in the 1970s. There's been a lot of mathematical work since.



Bigger Picture for Huge Data

Suggested Takeaways



- We need to look at where the volume of data is stressing our systems
  - FTP was designed in 1971, when a big file held a megabyte
  - Deep Medhi's talk @ CoNext ENCP 2019
- We need applications to log when they are in distress and share that data with researchers and operators
  - Errors tend to cluster (a bad system or protocol)
  - We want to find those errors (replace a bad system, improve a protocol)

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov