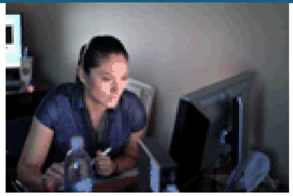
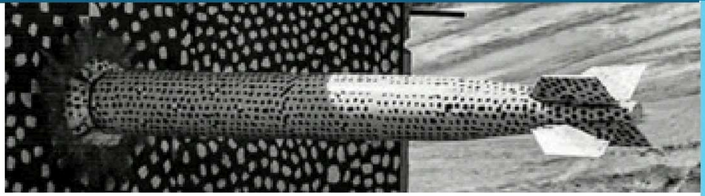
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2020-1791C

Developing Data-transfer Tools, Layers of lessons





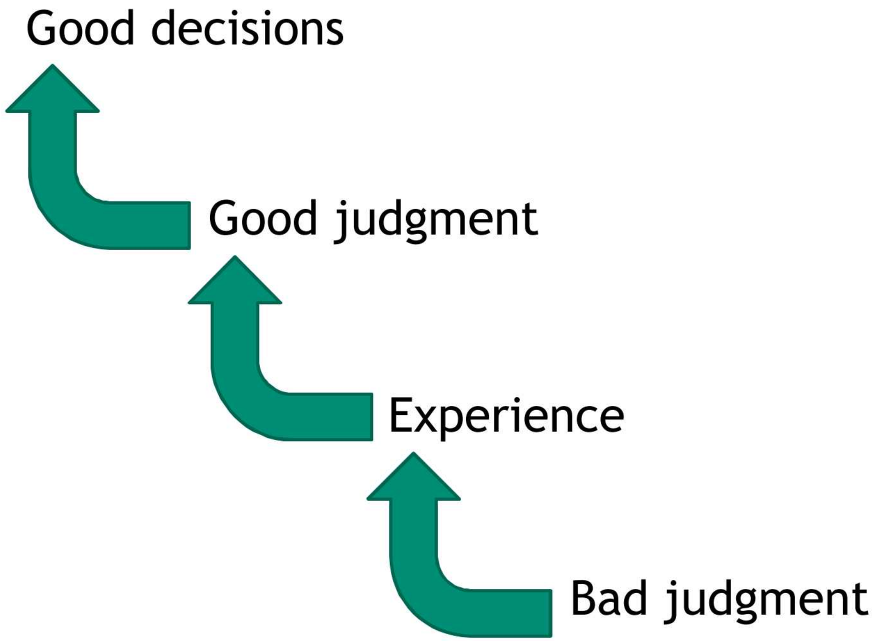
Presented by

William Collins, Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Why am I here?

Projects without forethought and analysis will suffer proportional degrees of pain and suffering. For the health and safety of the participants, we would do well to repeat what we desire, and avoid what we do not…



Good decisions

Good judgment

Experience

Bad judgment

"Fools say that they learn by experience. I prefer to profit by others experience."
— Otto von Bismarck

What’s FrETT?

- Friendly Extensible Transfer Toolkit
  - Storage-end-point agnostic data transfer tool
  - Designed from a contemporary perspective – replacing “venerable” tools
  - Leverage successes, Learn from failure
- “Layers of lessons” – not simply software
  - Management Lessons
    - Where do we start? How do we start? Why did we start?
  - Project Lessons
    - What’s worked, working, wrong? Now and then, and in the future.
  - Development lessons

FrETT, from the beginning

- Management – alternative being mis-management
  - Have PMP – didn't go "full", lightly managed
  - Release method – timed release, prioritized tasking
- Software status – "Do you have a license?"
  - Consider distribution audience – who **is** and **isn't**
  - Discuss with IP Lawyers
    - Sometimes "small print" details, like readline() – GPL, not LGPL!
- Requirements – we did say "Friendly"
  - FrETT – Site visits (e.g., HPSS collaborators), system administrators, developers, and **users**
  - Current/future usage patterns – scalable, modular, …
  - Program API, commands, session shell, GUI – broad "customer" base.

Lessons from the past

- Avoid single point of failure – group develop, some cross training
- Identify suitable IP/license status and it's requirements
  - With consideration to future…
- Successful existing tools, to varying degrees
  - Parallel File Transfer Protocol (PFTP) – striping, multi-interface, but unfortunately "spartan"
  - Hierarchical Storage Interface (HSI) – familiar (Unix-like), multi-thread, but HPSS-integral.
  - Transfer Agent (TA) – multi-host, but too platform/protocol specific
- What's in use (forms customer expectation)
  - GUI – Hopper, Filezilla, Globus web

FrETT - new approaches, new lessons

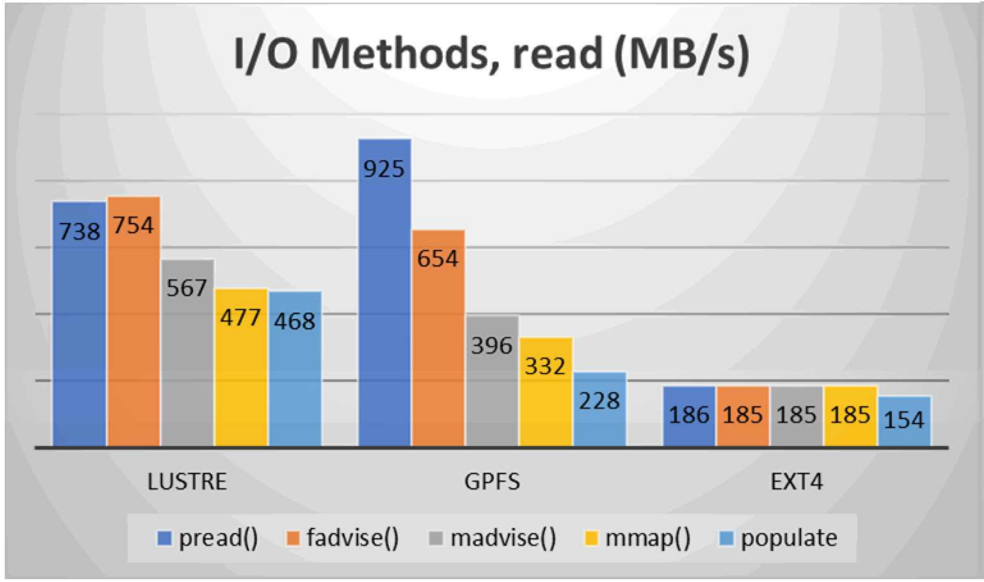
Modularity	
I/O method	✓
Authentication	✓
Pipline fitting	✓
Filesystem config	✓
Server-side module	✓
Transfer discipline	not yet

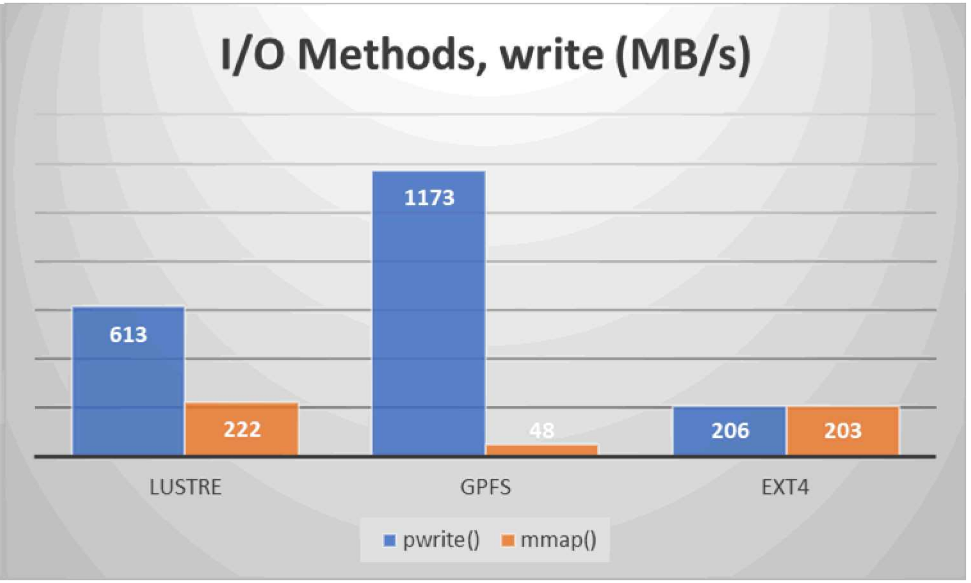
- More modularity – adds specificity
- Connection reuse/pooling
  - Improves small file network performance
    - TCP handshake: 3 * BDP = 100's of K or MB
  - Highlights filesystem meta-data costs
- Pipeline transfer framework
  - Simple case approximates double buffering
  - Simplifies adding encryption, compression, CRC, etc.
  - Additional function – Encryption, compression, data transmission integrity, …
- Early, v1.0, beta: transfer protocol needed fundamental rework

Developmental lessons (so far)

- FrETT Modules – opportunities to test/validate
  - Authentication modules – PAM-like callback over client/server
  - Peripheral flexibility – less "deep coding", simple API, component separation.
    - I/O method – pread/pwrite vs. mmap
- "Clean Code", documentation – doesn't happen by itself
  - Take long view…
- Qt – our GUI tool of choice
  - Rich featured, broad platform base, intuitive IDE
  - Not all features look the same everywhere… (e.g., Mac-OS "help")
  - X11 and event handling complex – ouch!

Validate: mmap I/O Methods? Not so fast.





FrETT project, a lesson in itself

- Some things take more time – even if you expect so
  - Time to feature – interruptions, design set-backs
  - Adoption – some sites have their own
- Acquiring test subjects – difficult, sparing
  - Like adoption, "chicken and egg" scenario
- Yet fulfilled features – stuff you'd like to do may just not happen.
  - Direct Input/Output(DIO), Async/Direct I/O, etc…
- More continuous partner involvement
  - Lab folk are **so busy**

Just data transfer?

32oz
9” wide
$2149.95

(I use this)

.8oz
2.3” long
$20.00

Questions?

Objects reflected
In a project schedule are
Closer than they seem.

- 101 Corporate Haiku, William Warriner