Contribution ID: **197** Type: **not specified**

Re-think of richACLs in AI/LLM era

This is Coly Li. I've been maintaining bcache for a while and have met Linus, Greg, Ted, and other maintainers in person at many conferences. Yes, I am a sustained and reliable kernel developer.

Recently, I joined a startup (https://fnnas.com) that provides AI/LLM capabilities for personal or micro-enterprise storage. We help users share and communicate AI/LLM-processed information from their stored data more conveniently.

Our users can run highly compact LLMs on their own normal and inexpensive hardware to process photos, videos, and documents using AI. Of course, it's slow but that's expected and acceptable. They can even come back to check the results weeks later.

In our use case, different people or roles store their personal and sensitive data in the same storage pool, with different access controls granted to AI/LLM processing tasks. When they share specific information or data with others within the same machine or over the internet, the access control hierarchy or rules become highly complicated and impossible to handle with POSIX ACLs.

We tried bypassing access control to user space, which worked well except for scalability and performance:
- As the number and size of files increase, storing all access control rules in user space memory doesn't scale—especially on normal machines without huge memory resources.
- For some hot data sets (a group of files and directories), checking access control rules in user space and hooking back to the kernel is highly inefficient.

Therefore, the RichACL project comes back to mind. Of course, RichACL alone isn't enough. A high-level policy agent (in user space) is still needed for task/session-oriented access and sharing policy control, but RichACL can help implement file system-level access control. This would give us a context-aware and highly efficient access control implementation.

What I'd like to discuss is:
- After almost 10 years, should we reconsider RichACL in the AI/LLM era?
- What are the major barriers or remaining work needed to get RichACLs into upstream?

Since our first public beta was released 13 months ago, we now have over one-million active installations running daily. This is a real workload for RichACL and represents real feature demand from end users. If you're interested in this topic, we'd be happy to provide more details about the access control requirements in AI workloads and even show a live demo of the use case.

Primary author: LI, Coly (fnnas.com)

Presenter: LI, Coly (fnnas.com)

Session Classification: Kernel Summit Track

Track Classification: Kernel Summit Track