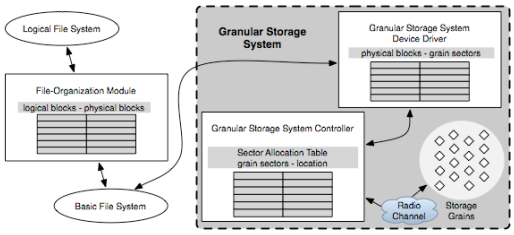
Grains of SANs: Building Storage Area Networks from Memory Spots

Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel

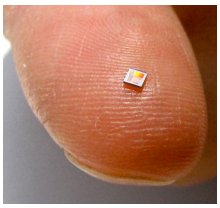
Grains of SANs

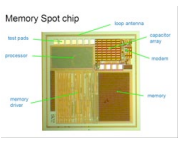
Emerging storage platforms are increasingly smaller, faster, and inexpensive. Organizing these small storage devices into a larger logical storage device would enable the idea of a portable, flexible storage device. We present such a system using memory spot-like devices, storage grains, as the basis of the storage device. This granular storage system is designed to dynamically support addition and removal of storage grains, thus supporting flexible capacity. Such a design changes the way that we think about storage: as a fixed capacity, stationary device.



HP's Memory Spot

HP announced "memory spots" in July 2006. These 2-4 mm^2 storage devices have between 265kb to 4Mb of flash memory and communicate data wirelessly at WiFi rates. They are wireless and have no on-board power, thus receive all operational power through inductive coupling, very similar to passive RFID. Upon possible release in 2 years, the price of a memory spot is estimated to be between 10 and 50 cents.





Research Challenges

Security

Challenges in securing a granular storage system:

- The unprotected wireless interface over which the grains communicate
- The naïve security capabilities of storage grains

In order to ensure a secure storage system, we must protect data confidentiality and integrity. We develop communication protocols to achieve these goals.

Initialization

1a. $\sigma = \{Kw_{SG}|K_{B,SG}\}_{K_{M,SG}}$ (key confidentiality)
1b. $h = \text{HMAC}_{Kw_{SG}}(ID_{SG}, n, \sigma)$ (message integrity)
1. $C \rightarrow SG : (INIT\_REQ, m, l)\ ID_{SG}, n, \sigma, h$ (initialization request)

2a. $h = \text{HMAC}_{Kw_{SG}}(succ, n)$ (message integrity)
2. $SG \rightarrow C : (INIT\_RESP, m, l)\ succ, n, h$ (counter value response)

Discovery

3. $C \rightarrow * : (HELLO\_REQ, m, l)$ (hello broadcast)
4. $SG \rightarrow C : (HELLO\_RESP, m, l)\ ID_{SG}, size$ (hello response)

Counter Sync

5. $C \rightarrow SG : (COUNTER\_REQ, m, l)\ ID_{SG}$ (counter value request)
6. $SG \rightarrow C : (COUNTER\_RESP, m, l)\ counter\_val$ (counter value response)

Read

7a. $sdata = \{\{data\}_{K_G}|\text{HMAC}_{K_G}(\{data\}_{K_G})\}$ (data confidentiality & integrity)
7b. $h = \text{HMAC}_{Kn_{SG}}(ID_{SG}, n, offset)$ (message integrity)
7. $C \rightarrow SG : (READ\_REQ, tn, l)\ ID_{SG}, n, offset, h$ (read request)

8a. $h = \text{HMAC}_{Kw_{SG}}(succ, n, sdata)$ (message integrity)
8. $SG \rightarrow C : (READ\_RESP, tn, l)\ succ, n, sdata, h$ (read response)

Write

9a. $h = \text{HMAC}_{Kw_{SG}}(ID_{SG}, n, offset, sdata)$ (message integrity)
9. $C \rightarrow SG : (WRITE\_REQ, tn, l)\ ID_{SG}, n, offset, sdata, h$ (write request)

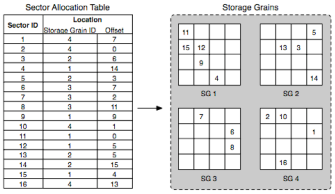
10a. $h = \text{HMAC}_{Kw_{SG}}(succ, n)$ (message integrity)
10. $SG \rightarrow C : (WRITE\_RESP, m, l)\ succ, n, h$ (write response)

Organization

Similar to the placement of data on a disk, organization of data in a granular storage system storage will effect performance. The difference lies in how performance is attained:

- Disks: exploitation of disk geometry
- Granular Storage: utilization of parallelism

We examine different methods of organizing data across the storage device in order to optimize performance.



Reliability

Reliability of our system involves ensuring the integrity of data, thus, minimizing the effects of data errors or loss. Challenges in providing reliability in a granular storage system:

- The system design enables dynamic addition and removal of storage grains possibly leading to immense data loss.
- The physical vulnerabilities of the devices may result in many data errors.

Techniques such as mirroring and error correcting codes can be used to aid in supporting reliability of a granular storage system.

Implementation and Initial Results

We implemented an initial system in order to:

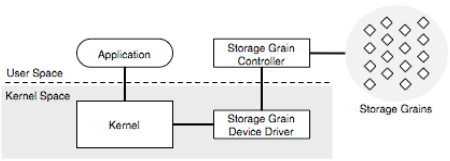
- Determine the viability of our design and security protocols.
- To investigate the effects of different data organization methods on the performance.

This implementation required the development of three central pieces:

Storage grain: Storage grains were developed with respect to the power and processing constraints of a memory spot.

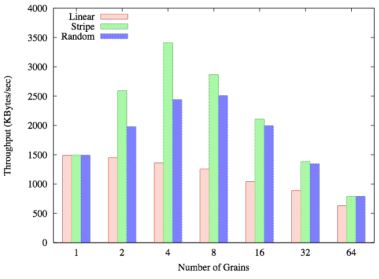
Controller: The storage grain controller interfaces between the kernel and the individual storage grains. It stores the mappings between sectors and their location within the memory spot.

Device Driver: A thin device driver was created in the kernel to interface between kernel commands and the storage grain controller. It stores the mappings between blocks and memory spot sectors.



We tested three different data organization strategies: linear, striping, and random. In order to see the effect of parallelism, we ran tests using varying numbers of grains between 1 and 64. Our initial results show:

- A positive effect of parallelism on throughput and degraded performance due to experimentation setup.
- Striping organization results in perfect equal distribution of data, thus maximizing parallelism and performance.



- A linear organization does not distribute data evenly and, as a result, has degraded performance.
- Under large distribution, the random method results in the same throughput as striping.