Summarizer:
Trading Communication with Computing Near Storage

*Gunjae Koo*, Kiran Kumar Matam*, Te I†, H.V. Krishna Giri Nara*, Jing Li‡, Hung-Wei Tseng†, Steven Swanson‡, Murali Annavaram*

*University of Southern California
†North Carolina State University
‡University of California, San Diego
Motivation - High Data Movement Cost

- **High access latency**
- **Limited data bandwidth**

**Diagram:**
- Host
  - **CPU**
  - Storage interface
- Data computation @ host
- Data transfer from storage
- External (Host ↔ Storage)
- Internal
Near Data Processing (NDP)

Host

- CPU
- Storage interface

Storage Processor (SP)

Data computation @ host
Data transfer from storage

Internal

External (host – storage)
Near Data Processing (NDP)

- **Host**: CPU
- **Storage Processor**
- **Storage interface**

**Without NDP**:
- Data computation @ host
- Data transfer from storage
- Internal

**With NDP**:
- Data computation @ storage

Comparing data transfer from storage with/without NDP.
Near Data Processing (NDP) on SSDs

- **Host**: CPU
- **Storage interface**: connects Host to Storage Processor (SP)
- **SP** contains Garbage collection and Wear-leveling

**W/O NDP**
- Data computation @ host
- Data transfer from storage
- Internal

**With NDP**
- Data computation @ storage
- Data transfer from storage
- Data computation @ storage
Near Data Processing (NDP) on SSDs

Obstacles to in-SSD processing

• Less powerful embedded processor
• Dynamic computation resource availability

Summarizer: Dynamic NDP framework for SSD
Summarizer – Basic Concept

- Host
- CPU
- Storage interface
- Monitoring resources
Summarizer – Basic Concept

Host

CPU

Storage interface

Monitoring resources
Summarizer – Detailed Firmware Architecture

- **Host Memory**
  - **Host CPU**
    - User Applications / Operating Systems
    - NVMe Host Driver

- **Storage Interface (PCIe / NVMe)**

- **SSD Firmware**
  - I/O Controller (NVMe command decoder)
  - Flash Translation Layer (FTL)

- **SSD Embedded Processors**
  - SSD SoC Interconnection
    - Flash Controller
    - DRAM Controller
    - NAND Flash
    - SSD DRAM

- **User Functions**
  - Task Controller
    - TQ
  - User Applications / Operating Systems

---

**Notes:**
- **SQ** and **CQ** are request and response queues for I/O operations.
- SSD Embedded Processors include SSD SoC Interconnection, Flash Controller, DRAM Controller, NAND Flash, and SSD DRAM.
Normal Page Read Request

Host CPU
- User Applications / Operating Systems
- NVMe Host Driver
  - RD (LBA)

Host Memory
- SQ
- CQ

Storage Interface (PCIe / NVMe)

SSD Firmware
- I/O Controller (NVMe command decoder)
- Flash Translation Layer (FTL)
  - (RD) PPA

Request queue
Response queue

SSD SoC Interconnection
- Flash Controller
- DRAM Controller
- NAND Flash
- SSD DRAM

Task Controller

User Functions

Summarizer
Normal Page Read Request

Host Memory
- SQ
- CQ

Storage Interface (PCIe / NVMe)

Host CPU
- User Applications / Operating Systems

NVMe Host Driver

SSD Firmware
- I/O Controller (NVMe command decoder)

Flash Translation Layer (FTL)

SSD SoC Interconnection
- Flash Controller
- DRAM Controller
- SSD DRAM

User Functions
- Task Controller
- Summarizer

Request queue
- RD(PPA 1)
- RD(PPA 2)

Response queue
Normal Page Read Request

Host CPU

User Applications / Operating Systems

NVMe Host Driver

Storage Interface (PCIe / NVMe)

SSD Firmware

I/O Controller (NVMe command decoder)

Flash Translation Layer (FTL)

Request queue

Response queue

SSD SoC Interconnection

Flash Controller

DRAM Controller

NAND Flash

Page data

SSD DRAM

User Functions

Task Controller

Summarizer
Summarizer – Initialization (Function Offloading)

Host CPU
User Applications / Operating Systems

Storage Interface (PCIe / NVMe)

SSD Firmware
I/O Controller (NVMe command decoder)

Flash Translation Layer (FTL)

SSD SoC Interconnection

Flash Controller
NAND Flash
SSD DRAM

User Applications / Operating Systems
User Functions

Task Controller

New NVMe command
NVMe Host Driver

INIT (foo)

foo()

Request queue
Response queue

Function registration
Function offloading

TQ
Summarizer – Computation (Dynamic mode)

Host CPU
User Applications / Operating Systems

Storage Interface (PCIe / NVMe)

Host Memory

SQ
CQ

New NVMe command decode

I/O Controller
(NVMe command decoder)

Flash Translation Layer (FTL)
RD&PROC(PPA,foo)

Request queue
Response queue

SSD SoC Interconnection

Flash Controller

DRAM Controller

NAND Flash

SSD DRAM

Task Controller

User Functions
f#1
foo()
f#2
goo()

New NVMe command

NVMe Host Driver
RD&PROC( LBA,foo)
Summarizer – Computation (Dynamic mode)

- User Applications / Operating Systems
- NVMe Host Driver
- Host Memory
- SSD Firmware
  - I/O Controller (NVMe command decoder)
  - Flash Translation Layer (FTL)
    - RD&PROC(PPA,foo)
    - RD&P(PPA1,foo)
    - RD&P(PPA2,foo)
- SSD SoC Interconnection
- Flash Controller
- DRAM Controller
- SSD DRAM
- Task Controller
- User Functions
  - f#1: foo()
  - f#2: goo()

Storage Interface (PCIe / NVMe)
Summarizer – Computation (Dynamic mode)

Host CPU
- User Applications / Operating Systems
- NVMe Host Driver

Host Memory
- SQ
- CQ

Storage Interface (PCIe / NVMe)

SSD Firmware
- I/O Controller (NVMe command decoder)
- Flash Translation Layer (FTL)
  - RD&PROC(PPA,foo)
  - RD&P(PPA1,foo)

I/O Controller

SSD SoC Interconnection
- Flash Controller
- DRAM Controller
- NAND Flash
- Page data

User Functions
- f#1
- foo()
- f#2
- goo()

Task Controller
- TQ
- CC
- TQ is full

Request queue
- foo()
- f#1

Response queue
- foo()
New NVMe command
NVMe Host Driver

User Applications / Operating Systems

Storage Interface (PCIe / NVMe)

Host Memory

Host CPU

SSD Firmware

I/O Controller (NVMe command decoder)

Flash Translation Layer (FTL)

SSD SoC Interconnection

Flash Controller

DRAM Controller

NAND Flash

Results

Task Controller

User Functions

f#1 foo()  
f#2 goo()
Summarizer API and NVMe commands

**Initialization**
- NVMe command: `INIT_TSKn`
- Transfer a *in-SSD procedure* to SSD memory
- Initialize data structure and temporal variables for in-SSD computation

**Computation**
- NVMe command: `READ_PROC_TSKn`
- Page read command is issued with the *flag* indicating the *user procedure* embedded in SSD memory
- Return the special code if the requested page is processed in SSD
- Page data is transferred to the host if the requested page is *NOT* computed in SSD

**Finalization**
- NVMe command: `FINAL_TSKn`
- Gather final in-SSD computation results and transfer to the host
Evaluation Platform

- LS2085a intelligent SSD development platform
- ARM cores running FTL and *Summarizer* firmware
- FPGA implementing NAND flash controller
- PCIe Gen. 3 4x lanes for host communication
Evaluation - Performance

TPC-H Query 6

- SDD time
- Host time

Static workload offloading

Static workload offloading
Evaluation - Performance

TPC-H Query6

- SDD time
- Host time

CPU only processing (baseline)

SSD only processing

Static

Dynamic

0

0.2

0.4

0.6

0.8

1

0

0.2

0.4

0.6

0.8

1
Evaluation - Performance

TPC-H Query 6

- SDD time
- Host time

Static

0 0.2 0.4 0.6 0.8 1

0 1 2 3 4

Summarizer Dynamic Offloading
Evaluation - Performance

TPC-H Query 6

- **SSD processing + transfer time (internal + external + In-SSD processing)**
- **Host CPU processing time**
TPC-H Query6

Execution time normalized to baseline (CPU only)
Evaluation - Performance

TPC-H Query6

Execution time (normalized to baseline)

- SDD time
- Host time

Static

Dynamic
Evaluation - Performance

Chart Title

- SDD time
- Host time

Execution time (normalized to baseline)

CPU only
- SDD time: 0.30
- Host time: 0.70

Dynamic
- SDD time: 0.24
- Host time: 0.60

Legend

- SDD time
- Host time
Evaluation - Performance

Performance improved by 14%

Comparison of SDD time and Host time between W/O NDP and With NDP.

W/O NDP:
- Data computation @ host
- External (host – storage)
- Internal

With NDP:
- Data computation @ storage
- Data transfer from storage
- Internal

CPU only: Static
Dynamic
Evaluation - Performance

TPC-H Query 6

Performance degraded by static NDP
Evaluation - Performance

TPC-H Query6
- SDD time
- Host time

Execution time (normalized to baseline)
- Static: 0, 0.2, 0.4, 0.6, 0.8, 1
- Dynamic

16%

TPC-H Query1
- SDD time
- Host time

Execution time (normalized to baseline)
- Static: 0, 0.2, 0.4, 0.6, 0.8, 1
- Dynamic

10%

TPC-H Query14
- SDD time
- Host time

Execution time (normalized to baseline)
- Static: 0, 0.2, 0.4, 0.6, 0.8, 1
- Dynamic

20%

String Similarity Join
- SDD time
- Host time

Execution time (normalized to baseline)
- Static: 0, 0.2, 0.4, 0.6, 0.8, 1
- Dynamic

7%
Design Exploration – Higher Internal Bandwidth

Commercial SSD maintains internal bandwidth ≈ external bandwidth
Design Exploration – Higher Internal Bandwidth

Higher internal bandwidth without increasing external bandwidth
Design Exploration – Higher Internal Bandwidth

External : Internal bandwidth ratio

- TPC-H Query 6
- TPC-H Query 1
- TPC-H Query 14
- String Similarity Join
- Average

Speedup

0% 100%

1:1 1:2 1:3 1:4

TPC-H Query 6  TPC-H Query 1  TPC-H Query 14  String Similarity Join  Average
Design Exploration – Higher Internal Bandwidth

**Summarizer is effective if an SSD platform has higher internal bandwidth**

![Graph showing speedup for different queries and operations]
Better embedded processor is cost effective
Design Exploration – Better SSD Processor

- Embedded processor performance
Summarizer is a cost effective NDP solution with powerful storage processors.
Conclusion

✓ **Dynamic NDP framework for SSDs**
  - Opportunistically enables in-SSD processing
  - Page-level NDP control
  - Automatic workload partitioning

✓ **Summarizer programming model**
  - Evaluation on the real development platform
  - Explored design space for future SSDs
Thank you

Summarizer:
Trading Communication with Computing Near Storage

Gunjae Koo, Kiran Kumar Matam, Te I, H. V. Krishna Giri Nara, Jing Li, Hung-Wei Tseng, Steven Swanson, Murali Annavaram

(We thank to Dell EMC for supporting the SSD development board)