After ARC : Verification about “Self-Tunable”

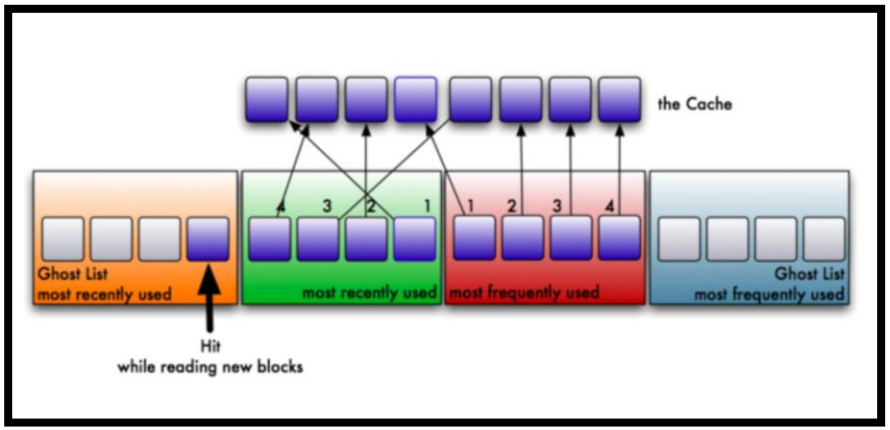
Jaehwan Jeong, Tackhee Lee

밑빠진독에코인붓기

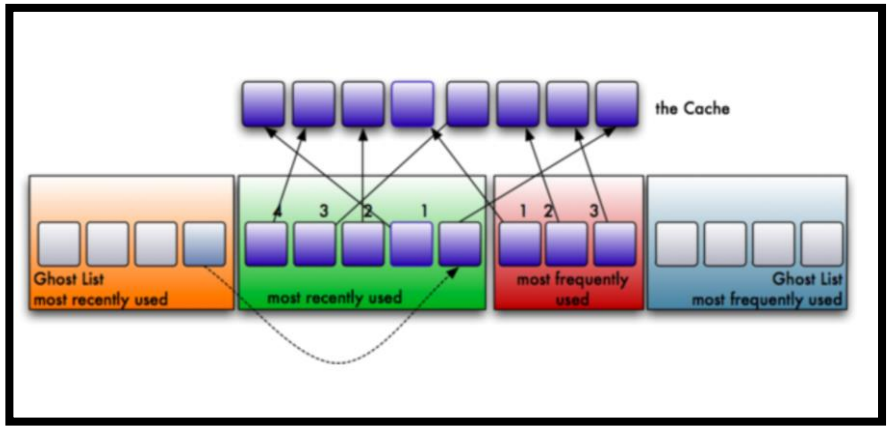
Problem Definition

Main Problem : Cache Replacement Algorithm

ARC : Adaptive Replacement Cache



Cache Miss(Phantom Hit)



Adaptive Re-sizing

Key idea : Phantom Hit -> re-size

Motivation

Is it truly "Self-tunable"?

1) Initial Value

ARC(c)

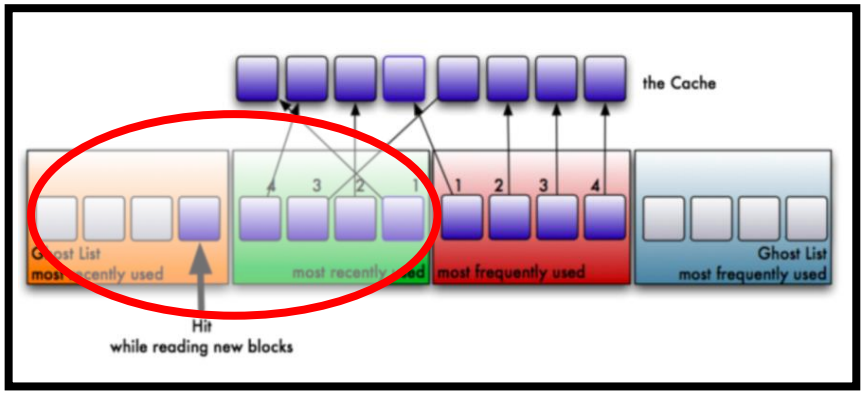
INPUT: The request stream $x_1, x_2, \ldots, x_t, \ldots$.
INITIALIZATION: Set $p = 0$ and set the LRU lists T_1, B_1, T_2, and B_2 to empty.

2) Adaptive Re-Sizing

$$p = \min\{p+\delta_1, c\}, \text{ where } \delta_1 = \begin{cases} 1 & \text{if} |B_1| \geq |B_2| \\ |B_2|/|B_1| & \text{otherwise.} \end{cases} \quad (1)$$

Why?

3) Size Limit of LRU Portion



$$0 \leq |L_1| \leq c$$

Our Idea

Find a tunable-parameter which gives more higher hit ratio

INITIALIZATION: Set $p = 0$

1) Initial Value

- p=0
- p=c

$$\delta_1 = \begin{cases} 1 & \text{if} |B_1| \geq |B_2| \\ |B_2|/|B_1| & \text{otherwise.} \end{cases}$$

2) Adaptive Re-Sizing (=Learning Rate)

- +2, -2 (CONST)
- $log_2\delta$ (LOG)
- 2^{δ} (EXP)

$$0 \leq |L_1| \leq c$$

3) Size Limit of LRU Portion

	$LRU : LFU$
$p_{max} = \frac{1}{3}c$	$(1:5)$
$p_{max} = \frac{2}{3}c$	$(1:2)$
$p_{max} = \frac{4}{3}c$	$(2:1)$
$p_{max} = \frac{5}{3}c$	$(5:1)$

Design - Traces

Real-Life benchmark disk I/O traces

- **EXCH : MS Exchange Server (mail server)**
- **DAP-DS : Display ads platform data server**
- **MSN : MSN metadata and file server**
- **RAD-AS : remote-access authentication server**
- **RAD-ES : remote-access back-end SQL server**
- **HOMES : NFS server from FIU**
- **WEB-VM : Virtual machine traces on running 2 web servers.**
- **MAIL : Mail server from FIU**

SNIA
IOTTA Repository

Summary : Diverse, Realistic Disk I/O traces

Design

Parsing Disk I/O Trace Result

Trace Name	Number of Request	Unique Pages
Exch	77304451	21724991
DAP-DS	11184349	5092722
MSN	11115258	7209398
RAD-AS	5529468	3323691
RAD-ES	39405995	20608323
HOMES	21163638	4760647
WEB-VM	14294158	549174
MAIL	14010588	1913909

TABLE I: Summary of disk traces

Methodology

Cache Behavior Simulation

Page-size : 4KB
Disk Cache : 256MB

+

Trace Name
Exch
DAP-DS
MSN
RAD-AS
RAD-ES
HOMES
WEB-VM
MAIL

Real-world
Disk I/O traces

→ ARC

p=0

p=c

+2, -2 (CONST)

$log_2\delta$ (LOG)

2^{δ} (EXP)

$p_{max}=\frac{1}{3}c$

$p_{max}=\frac{2}{3}c$

$p_{max}=\frac{4}{3}c$

$p_{max}=\frac{5}{3}c$

→ After-ARC → Higher Hit Ratio!

Evaluation

	EXCH	DAP-DS	MSN	RAD-AS	RAD-ES	HOMES	WEB-VM	MAIL
$p = 0$	12.71	3.59	16.71	22.19	19.43	59.28	73.16	54.61
$p = c$	12.65	3.69	15.77	20.34	19.57	59.39	73.13	54.60

TABLE II: Hit ratio as varying initial value p

- **Minor Difference**
- **p=0 wins up to 2% on RAD-AS**
- **p=0 -> good-choice**

Evaluation

	EXCH	DAP-DS	MSN	RAD-AS	RAD-ES	HOMES	WEB-VM	MAIL
$p_{max} = c$	12.71	3.59	16.71	22.19	19.43	59.28	73.16	54.61
$p_{max} = \frac{1}{3}c$	12.86	2.84	16.26	19.59	12.62	56.18	67.59	54.26
$p_{max} = \frac{2}{3}c$	12.88	3.45	18.07	21.58	20.62	59.05	74.60	55.81
$p_{max} = \frac{4}{3}c$	12.85	3.56	9.84	19.09	20.91	49.87	79.47	49.87
$p_{max} = \frac{5}{3}c$	12.48	3.37	7.01	15.80	19.77	42.87	78.90	42.87

TABLE III: Hit ratio as varying limit value of p

- **Big-Difference on Many trace**
- **Worst-case always occurs in (1:5) or (5:1)**
- **Best-case sometimes occurs in (1:2) or (2:1) -> not all workloads!**
- **p=c (always above the average)-> best-choice**
- **Finding 1 : ARC performs well better than LRU! (see (5:1) case)**

Evaluation

	EXCH	DAP-DS	MSN	RAD-AS	RAD-ES	HOMES	WEB-VM	MAIL
ORIGIN	12.71	3.59	16.71	22.19	19.43	59.28	73.16	54.61
CONST	12.79	3.64	15.81	21.32	20.60	58.82	71.94	52.87
LOG	12.71	3.57	16.26	21.61	19.44	59.22	73.45	53.91
EXP	12.88	3.72	16.37	22.47	20.60	58.72	70.63	53.59

TABLE IV: Hit ratio as varying the value of learning rate δ

- **EXP Wins 4 traces, ORIGIN wins 3, LOG wins 1.**
- **Finding 2 : EXP is quite good-choice**
- **ORIGIN -> above the average for all workloads.**

+2, -2 (CONST)

$log_2\delta$ **(LOG)**

2^δ **(EXP)**

Contribution & Conclusion

Contribution

- **Implementing ARC with tunable-parameter and real-world trace**
- **Traces : Was So Big**

Point 1

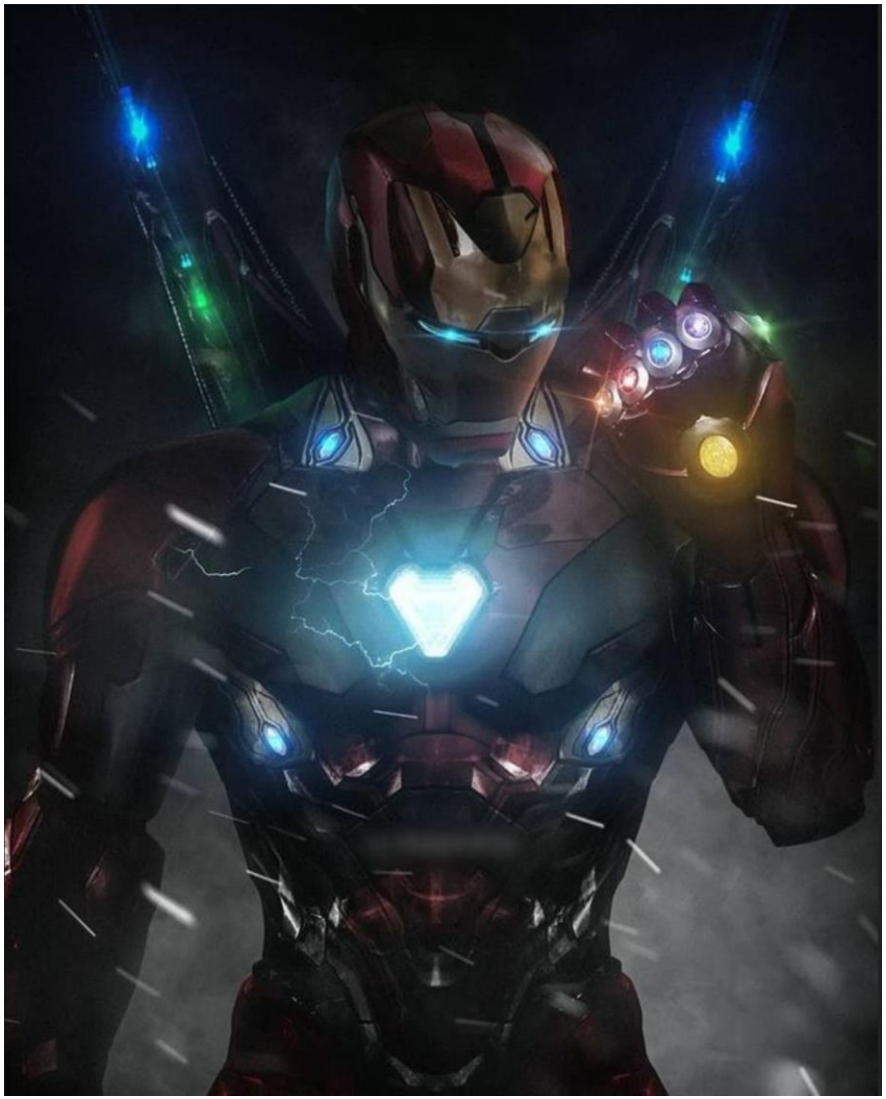
- **Finding 1) : Effectiveness of ARC is still valid on current I/O traces!**
- **Finding 2) : EXP is quite good-choice**

Point 2

- **Summary - ARC is well-made**
- **- Suggest EXP with diverse disk I/O pattern trace.**

Conclusion

Q&A



Come and Get Some