

1.검색기 구조

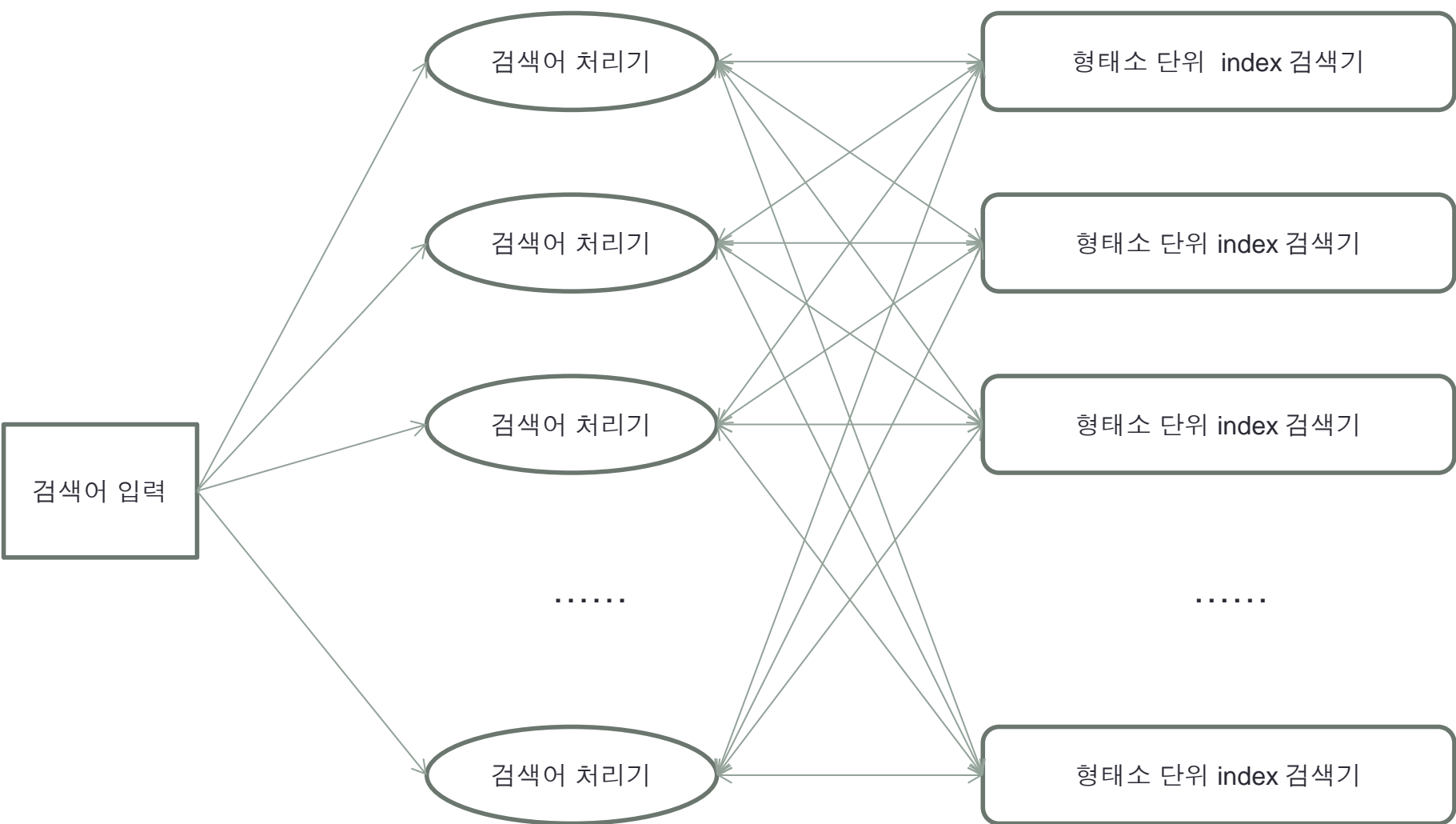
2.Throughput 측정

3.문제점 및 해결 방안

4.장비 산정

5.장비 수를 줄일 수 있는 방법

# 1. 검색기 구조



검색이 요청될 때 마다  
검색요청저리기 호출

검색어의 형태소 단위로  
검색 요청 후 결과 취합

# 1. 검색기 구조

## 1. 검색기 구성

### a. 검색어 처리기

- 검색 요청이 들어올 경우 검색어 처리기 process 생성(멀티 processing)
- 검색어를 공백 기준 형태소로 구분
- 형태소에 대해 첫 글자 알파벳 기준으로 분류하여 각각의 알파벳을 처리하는 index 검색기로 검색 분산처리 요청
- 검색 결과 수집 후 AND / OR 조건으로 후처리 및 검색 빈도 순으로 정렬

### b. Index 검색기

- B-Tree 구조로 메모리 올려져있는 index 정보에서 n 건의 Keyword에 대한 Index 정보를 검색 후 공유 메모리를 통해 검색어 처리기로 반환
- 알파벳 2개에 대한 Index를 하나의 index 검색기에서 처리(총 13개)  
해당 알파벳에 대한 키워드를 많이 가진 순으로 정렬하여 많은 키워드를 가진 알파벳과 적은 키워드를 가진 알파벳의 조합으로 선정

알파벳	s	c	a	p	m	b	t	d	r	e	h	l	g
키워드 수	44525	39203	35271	32478	31571	30484	25578	23460	21047	19149	18712	18505	17648
알파벳	x	q	y	z	j	u	v	o	w	k	n	i	f
키워드 수	1906	2846	3094	3699	6648	8887	9755	12181	12587	13284	14695	15354	15956
키워드 합	46431	42049	38365	36177	38219	39371	35333	35641	33634	32433	33407	33859	33604

- 서버 부하를 고려하여 13개 index 검색기를 하나의 그룹으로 12개 그룹의 index 검색기를 실행  
➔ 13 개 process \* 12그룹 : 156개 process

# 1. 검색기 구조

## 2. 검색어 처리 process

- 1) 검색어 입력
- 2) 공백 기준 형태소 추출 및 형태소를 알파벳 기준으로 분류
- 3) 각 index 검색기로 조회 요청 후 index 조회 결과 수집
- 4) 수집된 결과를 바탕으로 최종 검색 결과 도출

## 3. 검색어 처리 process 예시

- 입력 검색어 : “chris” “evans” “captain”

- 1) 검색어 처리기 process 생성
- 2) 공백 기준 추출 및 첫 글자 알파벳 기준 분류
  - a. 알파벳 c : chris, captain
  - b. 알파벳 e : evans
- 3) index 검색기로 요청 및 결과 수집
  - a. 알파벳 c에 해당하는 index 검색기로 chris, captain 동시 요청 및 결과 수집  
index 검색기에선 각각의 형태소를 순차적으로 처리 후 결과는 한번에 리턴
  - b. 알파벳 e에 해당하는 index 검색기로 evans 요청 및 결과 수집
- 4) 모든 형태소에 대해 결과가 수집되면 수집된 index에 대해 AND/OR 조건 취합 및 빈도 수에 맞춰 결과 정렬
  - 입력 검색어는 모두 AND 조건이므로 모든 Keyword 가 포함된 문서를 출력

## 2. Throughput 측정

### 1. 측정 환경

- a. 서버 모델 : HP DL380e Gen8
- b. Processor : Intel Xeon E5-2430 2.2GHz 6Core

### 2. Row data

- a. Wikipedia 문서 56만건/2.8GB

### 3. 측정 내역

- a. 검색어 요청에 따른 검색 소요 시간/결과 처리 시간 측정
- b. 검색 성능을 보장하는 최대 동시 요청 건 수 측정

### 4. 측정 방법

- a. 서버 최대 부하 측정
  - 1) 15개 검색어들에 대해 1초 단위로 10초간 동시에 10회/20회....70회 요청  
→ 10초동안 매 초 150개/300개..../1050개 검색어 요청 (총 7회 시험)
  - 2) 실시간 부하 감시 후 cpu 사용률이 100%가 되는 순간의 검색어 동시 처리 수 확인 (idle : 50%)
    - index 검색기의 경우 동시 12개의 index 검색기 group(13개) 구동 후 동시 처리 시 cpu 사용률 100% 도달
    - 검색어 처리기의 경우 750여개 검색 요청에 100% 도달
- b. 검색 성능 측정
  - index 검색기 13그룹을 실행시켜 index를 메모리에 올려놓은 후 15개 검색어에 대해 동시에 50회 요청하여 750개 검색어 요청을 처리한 시간을 측정

### 5. 요청 검색어 리스트 및 성능 측정 결과

검색어	문서 검색 건수(건)		index 검색 시간(초)			검색 조건 정렬(검색시간 포함)(초)		
	키워드 전체 문서	조건 검색 결과	최소	최대	평균	최소	최대	평균
whitehouse	690	690	0	0	0	0	0.1	0.014
labrador retriever	1403	1331	0	0	0	0	0.1	0.085
amazon	3549	3549	0	0	0	0	0.1	0.043
"apple" "keyboard"	9899	521	0.01	0.03	0.02	1.14	1.74	1.55
wireless mouse	5942	5850	0.01	0.04	0.014	0.83	1.0	0.91
america	43229	43229	0.73	1.3	1.2	0.74	1.31	1.22
africa	19782	19782	0.13	0.15	0.142	0.13	0.16	0.153
"captain" "america"	49920	1518	0.73	1.3	1.2	38.91	46.21	42.77
"merry" "christmas"	7781	462	0.01	0.02	0.18	1.06	1.18	1.15
linux "kernel" version	49066	3276	0.85	1.12	1.025	1.04	1.47	1.33
"mavel" "spiderman"	213	0	0	0	0	0	0	0
"chris" "evans"	12254	582	0.01	0.04	0.029	2.4	2.65	2.56
"chris" "evans" "captain"	18945	29	0.09	0.14	0.11	3.81	4.02	3.94
chris evans captain	18945	18886	0.1	0.15	0.134	2.35	2.54	2.484
google glasses	15829	15704	0.1	0.19	0.157	1.07	1.26	1.17

### 6. 분석

- index 검색에 소요되는 시간은 “africa” 검색어와 “america” 검색어의 결과를 비교하였을 때 문서 건 수가 증가하면 기하급수적으로 증가  
➔ 약 3만8천건 정도를 최대 1초 내에 처리
- 검색 조건에 대한 처리에 따른 소요 시간은 “chris” “evans” “captain” 검색어와 “chris evans captain” 비교하였을 때 중복 처리와 AND/OR 처리에 소요되는 시간이 적지 않음

### 7. 개선 내용

- Index 검색 시간을 단축하기 위하여 키워드를 3만건 이상 가진 알파벳에 대하여 2개의 index 검색기로 분산 처리  
(s, c, a, m, p, n, b)
- 각각의 형태소에 대한 검색결과를 정렬하여 조건 검색 중 10건에 대한 결과를 추출한 시간 측정

### 8. 개선 후 재측정

검색어	index 검색 시간(초)			10건 검색 조건 정렬(검색시간 포함)(초)		
	최소	최대	평균	최소	최대	평균
whitehouse	0	0	0	0	0.1	0.014
labrador retriever	0	0	0	0	0.1	0.085
amazon	0	0	0	0	0.1	0.043
"apple" "keyboard"	0.01	0.03	0.02	0.34	0.75	0.57
wireless mouse	0.01	0.04	0.014	0.15	0.21	0.19
america	0.42	0.78	0.56	0.43	0.79	0.57
africa	0.13	0.15	0.142	0.13	0.16	0.153
"captain" "america"	0.42	0.78	0.56	0.78	1.02	0.96
"merry" "christmas"	0.01	0.02	0.18	0.27	0.43	0.38
linux "kernel" version	0.85	1.12	1.025	0.92	1.02	0.94
"mavel" "spiderman"	0	0	0	0	0	0
"chris" "evans"	0.01	0.04	0.029	0.76	0.92	0.88
"chris" "evans" "captain"	0.09	0.14	0.11	0.99	1.21	1.12
chris evans captain	0.1	0.15	0.134	0.12	0.16	0.13
google glasses	0.1	0.19	0.157	0.11	0.2	0.18
평균			0.195			0.413

### 3. 문제점 및 해결 방안

#### 1. 검색어 처리기의 멀티 프로세싱에 따른 자원 소모

- 최초 설계 시 검색 요청 건당 프로세스가 생성되고 종료되는 구조로 접근
- 프로세스 단위 처리 시 자원소모가 많음
- 부하 측정을 위해 검색어 처리기 프로세스의 수를 제한하지 않음

##### ➔ 해결 방안

- 일정 수의 검색어 처리기 데몬 프로세스를 띄워 놓고 검색 요청이 들어올 경우 기존 검색처 처리기 역할을 수행하는 함수에 대해 스레드 단위로 처리
- 검색 요청이 갑자기 증가하더라도 세마포어를 통해 index 처리기로의 요청에 대한 스레드의 부하 제어
- Index 처리기로의 접근 시 뮤텍스를 이용하여 Index 처리기가 요청을 접수 받을 때까지 다른 스레드의 요청이 들어가는 것을 방지

#### 2. Index 처리기의 Context switching Issue

- 검색어 처리기로 부터 공유 메모리로 요청 검색어 및 공유 메모리 고유Key를 입력 받아 검색 결과를 고유Key를 바탕으로 검색어 처리기로 넘겨주도록 구현
- 13개 검색어가 동시 입력될 때 동일한 알파벳에 대한 검색 요청 시 12 그룹의 index 검색기에서 작업하는 동안 1개의 검색어 처리기는 ready 상태

##### ➔ 해결 방안

- Index 검색기로 들어오는 요청에 대해 스레드 단위로 검색하여 처리
- 12개의 index 검색기 그룹에서 동일한 알파벳 keyword는 동시에 12개까지 처리하나 50개의 스레드를 이용해 처리한다면 600개까지 처리 가능



## 4. 장비 산정

### 1. 장비 산정 기준

#### a. 일일 10억건 처리

Peak time / Issue로 인한 실급검 발생시 1시간에 1억건을 처리한다고 가정한다면 초당 처리 건 수는

$$100,000,000 / (60\text{분} * 60\text{초}) = 27,777 \text{ 건}$$

30% 정도의 Buffer 부하를 두어 초당 4만건을 처리한다고 가정

#### b. Throughput 측정 시 적용한 구조로 처리한다고 가정하여 검색어 처리기 + index 검색기 로 장비 산정

### 2. 장비 산정

#### - 보완 사항 적용 후를 기준으로 장비 산정

##### a. 검색어 처리기

최대 동시처리 건수 / 평균 검색 소요 시간 =  $750 / 0.413 = 1816$  건

→ 필요 장비 수 =  $40,000 \text{ 건} / 1816 \text{ 건} = 22 \rightarrow 22\text{대}$

##### b. Index 검색기

- Index 검색기에서 소요되는 검색 시간은 검색어 처리기 처리 시간에 포함

- 검색어 처리기가 동시에 처리 가능한 최대 건수를 기준으로 장비 산정

(검색어 동시 처리 가능 수 \* 검색어 처리기 장비 수) / index 검색기 동시 처리 가능건수

→ 필요 장비 수 =  $(750 * 22) / 600 = 27.5 \rightarrow 28\text{대}$

## 5. 장비 수를 줄일 수 있는 방법

### 1. 검색 결과 **cache** 사용

검색어가 대량으로 처리되는 **peak time**의 경우 특정 **keyword**에 대한 검색 요청이 많을 것이고 이 부분에 대해 매번 같은 검색 처리를 하기보다 검색 결과에 대한 **cache**를 사용하여 동시 처리 부하가 많은 검색어에 대해 **cache** 검색을 통하여 결과를 보여준다면 검색어 처리기에 대한 처리 부하가 감소할 것으로 예상.

Cache hit rate 20%일 경우 검색어 처리기 대수 :  $22 \% 0.8 = 17.6 \rightarrow 18$ 대

### 2. **Achiving**

특정 인물, 지명 등에 대한 **keyword**의 경우 실시간 데이터 이외의 데이터에 대해 일정 시간 단위로 **Archive**화 하여 검색 요청에 대해 **Archive** 된 **page**를 실시간 데이터와 함께 보여준다면 검색 처리 시간이 줄어들 것으로 예상.

### 3. 요청 빈도에 따른 검색어 관리

검색 요청이 적을 것으로 예측되는, 혹은 검색 요청이 적은 **Keyword** 들에 대해 따로 수집하고 하나의 파티션에 과적하여 운용한다면 검색 요청이 많은 **keyword**들에 대한 검색 요청에 대해 더 빨리 처리할 수 있을 것으로 예상.