

ICT의 가치를 이끄는 사람들!!

131회

컴퓨터시스템응용기술사 기출풀이 4교시

국가기술자격 기술사 시험문제

정보처리기술사 제 131 회

제 4 교시

분야	정보처리	종목	컴퓨터시스템응용	수험 번호		성명	
----	------	----	----------	-------	--	----	--

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. 강화학습(Reinforcement Learning)은 최적의 행동정책을 찾아가는 기계학습 방법이다. 이와 관련하여 다음을 설명하시오.
 - 가. 가치기반 강화학습, 정책기반 강화학습, 엑터 크리틱(Actor-Critic) 강화학습
 - 나. 정책경사(Policy Gradient) 방식 강화학습
2. 추천시스템은 사용자의 과거 행동데이터 등을 바탕으로 사용자가 좋아할 만한 정보나 제품을 제시해 주는 시스템이다. 이와 관련하여 다음을 설명하시오.
 - 가. 컨텐츠 기반 필터링(Content-based Filtering)과 협업적 필터링(Collaborative Filtering) 기법
 - 나. 행렬분해(Matrix Factorization) 기반 협업적 필터링
3. 캐시 메모리(Cache Memory)에 대하여 다음을 설명하시오.
 - 가. 캐시 메모리의 개념과 구조
 - 나. 지역성(Locality)의 개념과 유형
 - 다. 캐시 일관성(Coherence)문제의 원인과 해결 방법
4. 스마트홈 연결 표준인 매터(Matter)에 대하여 다음을 설명하시오.
 - 가. 매터의 개념
 - 나. 기존 스마트홈 표준의 한계점과 매터의 장점
 - 다. 매터 Network Stack
5. 시스템 성능 테스트 수행 시, 성능 지표 관련하여 다음 물음에 답하시오.
 - 가. 응답시간(Response Time), 대기시간(Think Time), 동시사용자(Concurrent User), 활성사용자(Active User), TPS(Transaction Per Second)를 설명하시오.
 - 나. 동시사용자(Concurrent User)가 100 명이고, 응답시간(Response Time)이 5 초이내이며, 대기시간(Think Time)이 15 초인 시스템의 경우, TPS를 구하시오.
6. 공공기관 정보화 사업 추진 시, 상용 SW 직접구매 제도와 관련하여 다음을 설명하시오.
 - 가. 상용 SW 직접구매 적용대상
 - 나. 상용 SW 직접구매 예외기준
 - 다. 상용 SW 직접구매와 일괄발주 비교

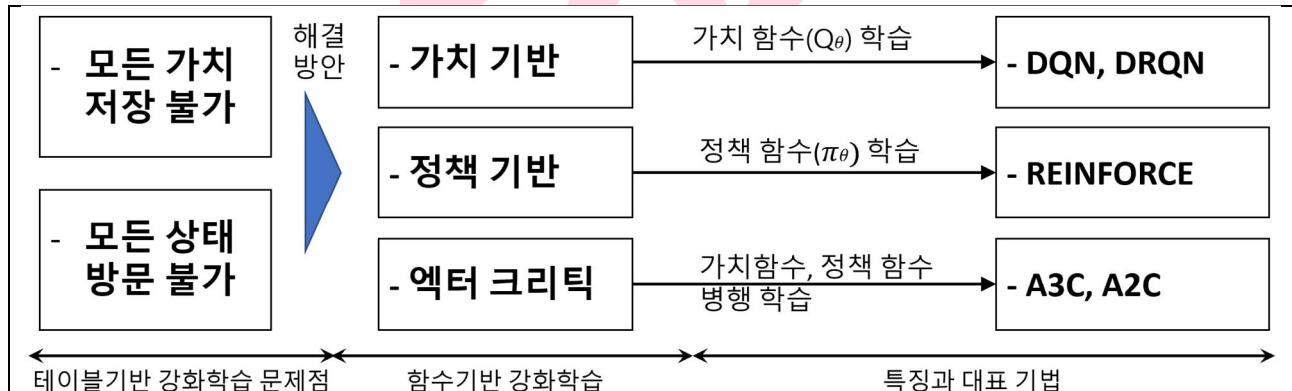
1. 강화학습(Reinforcement Learning)은 최적의 행동정책을 찾아가는 기계학습 방법이다. 이와 관련하여 다음을 설명하시오.

문제 제 가. 가치기반 강화학습, 정책기반 강화학습, 엑터 크리틱(Actor-Critic) 강화학습

나. 정책경사(Policy Gradient) 방식 강화학습

출제영역	인공지능	난이도	★★★★☆
출제배경	- 강화 학습에서 사용하는 Model Free 기반 강화학습 이해여부 확인		
출제빈도	미출제		
참고자료	<ul style="list-style-type: none"> - https://with-rl.tistory.com/entry/바닥부터-배우는-강화-학습-08-가치-기반-에이전트 - https://lee-jaewon.github.io/reinforcement_learning/RL_intro_3/ - http://dmqm.korea.ac.kr/activity/seminar/391 - https://github.com/hilarious-/RL-PolicyGradient_summarization - https://securitynewsteam.tistory.com/544 		
Keyword	- 가치함수, 정책함수, DQN, REINFORCE, A3C, A2C		
풀이	김대원(130회 정보관리기술사 / wdw4004@naver.com)		

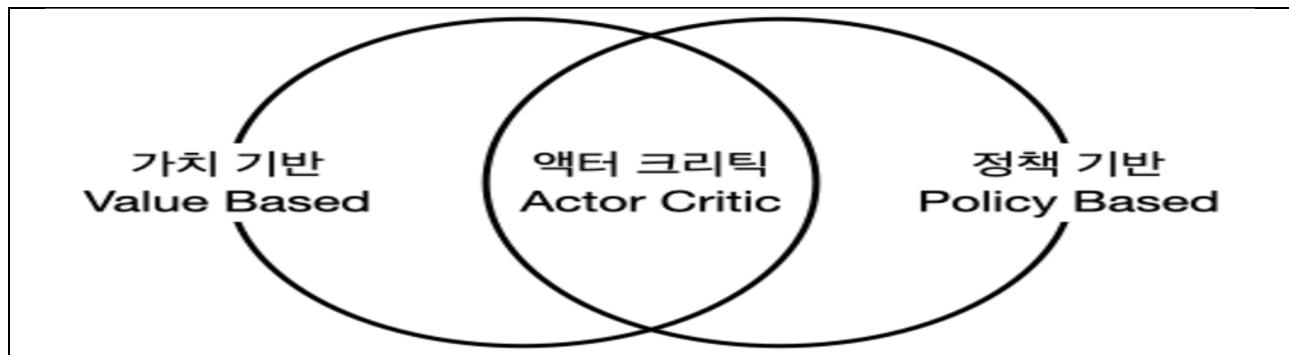
1. 함수기반 강화학습의 등장 배경



- 테이블 기반 강화학습의 가치/상태기반 저장 및 방문 한계 해결을 위해 함수기반 강화학습 등장.

2. 함수기반 강화학습, 가치기반 강화학습, 정책기반 강화학습, 엑터 크리틱 강화학습 설명

가. 함수 기반 강화학습 유형 관계도



- 크게 가치기반과 정책기반으로 구분되며, 가치기반/정책기반 학습을 병행하는 엑터 크리틱을 추가하기도 함.

나. 가치기반, 정책기반, 엑터 크리틱 강화학습 상세 설명

유형	분류	상세설명
가치기반 강화학습	개념	- 실제 가치를 추정할 <u>가치함수(Q_θ)</u> 를 학습하는 가치기반 행동을 선택하는 방법론
	특징	- 별도 명시된 정책함수가 없으며, 가장 높은 가치를 갖는 행동 선택.
	수식	$Q_{k+1}(S_t, A_t) \leftarrow Q_k(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q_k(S_{t+1}, a') - Q_k(S_t, A_t))$
	설명	<ul style="list-style-type: none"> - α : Learning rate - $(R_{t+1} + \gamma \max_{a'} Q_k(S_{t+1}, a') - Q_k(S_t, A_t))$: 시간차 오차(Time-Difference Error) - $Q_{k+1}(S_t, A_t)$: 기존 $Q_k(S_t, A_t)$에서 $\alpha(R_{t+1} + \gamma \max_{a'} Q_k(S_{t+1}, a') - Q_k(S_t, A_t))$를 더한 값
	대표 알고리즘	- DQN, DRQN, DDQN
정책기반 강화학습	개념	- 주어진 상태에 대하여 정책함수(π_θ)가 행동을 직접 선택하는 방법론
	특징	- 확률 분포 기반 행동을 선택
	수식	$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)] = \sum_{s \in S} d(s) * v_{\pi_\theta}(s)$
	설명	<ul style="list-style-type: none"> - $\mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$: Return 의 기대값 - $d(s)$: 시작점의 특정 상태(S)일 확률 분포 - $v_{\pi_\theta}(s)$: 시작점의 상태 가치(=Return)
	대표 알고리즘	- REINFORCE
엑터 크리틱 강화학습	개념	- 정책함수(액터)와 가치함수(크리틱)를 함께 학습하는 방법론
	특징	<ul style="list-style-type: none"> - 액터는 주어진 상태에서 행동을 선택, 크리틱은 선택한 행동의 가치 평가 - 정책함수(π_θ)와 가치함수(Q_w)는 서로 다른 파라미터(θ, w)로 구성되며 각각 학습 필요
	수식	<ul style="list-style-type: none"> - 액터 업데이트 : $\theta \leftarrow \theta + \alpha_1 \nabla_\theta \log \pi_\theta(s, a) * Q_w(s, a)$ - 크리틱 업데이트 : $w \leftarrow w + \alpha_2 (r + \gamma Q_w(s', a') - Q_w(s, a)) \nabla_w Q_w(s, a)$

	설명	- $Qw(s, a)$: 가치함수 추정치 - $r + \gamma Qw(s', a')$: TD Target - $r + \gamma Qw(s', a') - Qw(s, a)$: Time-Difference(TD) Error
	대표 알고리즘	- Q Actor-Critic, Advantage Actor-Critic(A2C), Asynchronous advantage actor-critic(A3C), TD Actor-Critic

- 정책기반 강화학습은 정책 경사(Policy Gradient)를 이용하여 학습 수행.

3. 정책 최적화 기법, 정책 경사(Policy Gradient)방식 강화학습 설명

가. 정책 경사(Policy Gradient)방식 강화학습 개념

개념	- 정책자체를 모델링하여 최적화하기위해 목적함수에 대한 기울기를 계산하고, 기대보상을 최대화하는 기법	
특징	확률적 정책 학습	- 상태에서 각 액션을 취할 확률을 파라미터화하여 학습
	연속적 정책 개선	- 최대화 보상방향으로 지속적 정책 개선
	대규모 액션 공간	- 연속적 액션 공간을 다룰 수 있기때문에 대규모 액션공간에서 잘 동작

- 정책 경사는 파라미터를 조정하여 높은 보상을 획득하는 최적의 정책을 찾기위해 딥러닝을 활용한 강화학습에서 활용

나. 가치기반, 정책기반, 엑터 크리틱 강화학습 상세 설명

구분	유형	상세설명
정책기반 강화학습	REINFORCE	- Monte-Carlo policy gradient, - episode의 샘플들을 활용해 policy parameter θ 를 update 하는 기법
엑터 크리틱 강화학습	Actor-critic	- actor 와 critic 이 매 time-step 에서 TD-error 를 활용해 각각 policy network (parameterized with θ)와 value-function (parameterized with ω)을 업데이트하는 기법
	A3C	- 전역 Actor-Critic 망에 다수 Agent 들의 기울기를 각자 계산하여 전역 Actor-Critic 망 기울기에 비동기적으로 업데이트하는 기법
	A2C	- A3C를 Coordinator 기반 동기적으로 업데이트하는 기법

- 정책기반 강화학습은 정책 경사(Policy Gradient)를 이용한 학습 수행시 샘플 최적화 및 분산성 고려 필요.

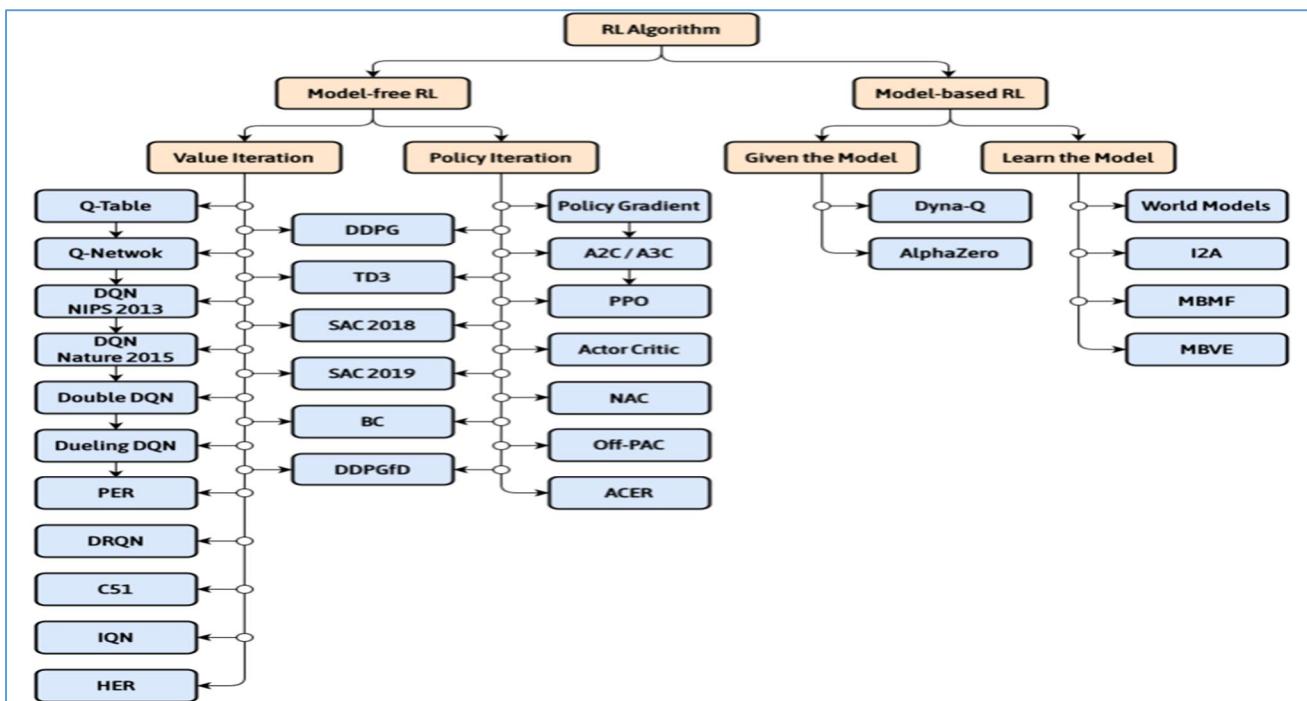
4. 정책 경사(Policy Gradient)방식 강화학습사용시 고려사항

고려사항	설명
샘플 효용성	- 정책 경사방식은 샘플링 기반 방법이므로 적은 샘플로도 좋은 결과를 얻을 수 있도록 최적화 필요
높은 분산성	- 추정값이 가지는 높은 분산성에 의한 안정성과 수렴속도에 대한 영향 고려
Local Optimal	- 국소 최적값에 수렴하지 않도록 초기화 전략 고려

- 강화학습의 다양한 문제와 도메인에 따라 알고리즘과 파라미터 설정 고려 권장.

"끝"

[참고자료]



kpc

기출풀이 의견

1. 강화학습을 심도있게 유형과 특징을 구분하지 못하면 기술하기 어려운 문제입니다. 반대로 Model Free기반 유형에 대해서 알고 있었다면 고득점이 가능한 문제입니다. 또한 유사출제에 대비하여 Model Base 강화학습까지 학습하시기 권장드립니다.

문제

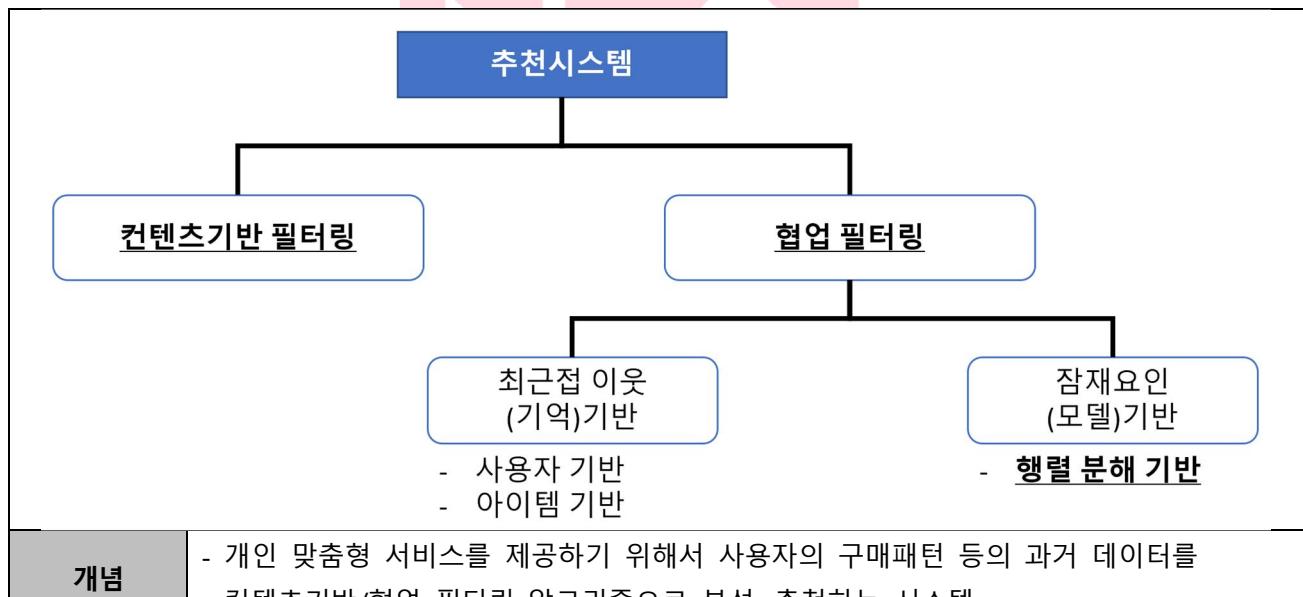
2. 추천시스템은 사용자의 과거 행동데이터 등을 바탕으로 사용자가 좋아할 만한 정보나 제품을 제시해 주는 시스템이다. 이와 관련하여 다음을 설명하시오.

가. 컨텐츠 기반 필터링(Content-based Filtering)과 협업적 필터링(Collaborative Filtering) 기법

나. 행렬분해(Matrix Factorization) 기반 협업적 필터링

출제영역	알고리즘	난이도	★★★★☆
출제배경	- 추천시스템의 주요 분류인 컨텐츠기반/협업적 필터링과 기업에서 가장 사용도가 높은 행렬분해 기반 협업적 필터링의 속지여부 확인		
출제빈도	미출제		
참고자료	<ul style="list-style-type: none"> - 73회, 90회, 97회, 110회 모의고사 - https://re-b-iew.tistory.com/24 - https://s-space.snu.ac.kr/bitstream/10371/123562/1/000000016675.pdf - https://ssongblog.tistory.com/122 		
Keyword	- 최근접 이웃(사용자 기반/아이템 기반), 잠재요인(행렬 분해기반), 유사도 기반, Latent Factor, SGD/ALS, SVD/NMF, Cold Start/필터버블		
풀이	김대원(130회 정보관리기술사 / wdw4004@naver.com)		

1. 개인 맞춤형 서비스 제공위한 추천 시스템 알고리즘의 개요



- 추천시스템 분석 알고리즘은 컨텐츠 기반과 협업 필터링으로 주요 유형을 구분.

2. 컨텐츠기반 필터링(Content-based Filtering)과 협업적 필터링(Collaborative Filtering) 기법 설명

가. 컨텐츠기반 필터링 설명

개념	<ul style="list-style-type: none"> - 사용자가 소비한 아이템에 대해 아이템의 내용(content)이 비슷하거나 특별한 관계가 있는 다른 아이템을 추천하는 기법 	
프로세스	<p>The diagram illustrates the Content-based Filtering process. On the left, a cartoon character labeled "이용자" (User) is shown. An arrow labeled ① 사용 (Use) points from the user to a blue rectangle labeled "콘텐츠 A" (Content A). From Content A, a yellow double-headed arrow labeled "유사도 높음" (High Similarity) connects to another blue rectangle labeled "콘텐츠 A#" (Content A#). An arrow labeled ② 추천 (Recommend) points from Content A# back to the user.</p>	
설명	<ul style="list-style-type: none"> - 다른 유저의 행동 이력을 사용하지 않고 유저가 관심있는 아이템의 속성을 분석하여 새로운 아이템을 추천 - 아이템들의 feature를 추출하기위한 TF-IDF나 Word2Vec과 같은 Feature Extraction 방법론 사용 - 추출된 feature 통해 아이템을 비교할 유사도(Similarity)에 대한 선택이 중요 	
유사도 측정 방법	아이템 특징(Feature) 점 추출	<ul style="list-style-type: none"> - 아이템을 표현하는 데이터의 유형에 따라 원 핫 인코딩(One-hot encoding), 임베딩(Embedding) 등의 방법으로 벡터(Vector)로 표현
	유사도 측정	<ul style="list-style-type: none"> - 다양한 벡터 유사도 측정 방식을 이용 - 코사인 유사도, 자카드계수, 유클리디언 유사도, 피어슨 상관 계수

- 아이템간 유사도 측정을 위해 유클리디언 유사도, 자카드 계수, 코사인 유사도, 피어슨 상관계수로 측정.

나. 협업적 필터링 설명

개념	<ul style="list-style-type: none"> - 사용자나 아이템 프로필 데이터없이, 사용자의 과거 행동 데이터(구매, 시청, 부여평점 등)를 가지고 유사도를 측정, 추천하는 기법 	
프로세스	<p>The diagram illustrates the Collaborative Filtering process. On the left, a user labeled "이용자 A" (User A) uses item C (콘텐츠 C). On the right, a user labeled "이용자 B" (User B) uses item D (콘텐츠 D). A yellow double-headed arrow labeled "유사도 높음" (High Similarity) connects Content C and Content D. An arrow labeled ③ 추천 (Recommend) points from Content C back to User A.</p>	
설명	<ul style="list-style-type: none"> - 사용자 간 유사도(Similarity)기반, 높은 유사도를 가진 사용자에게 관련 아이템을 추천. 	
주요 유형	기억 기반	<ul style="list-style-type: none"> - User-Based 필터링, Item-Based 필터링, - Hybrid 필터링, 수동 필터링
	모델 기반	<ul style="list-style-type: none"> - 군집화/나이브베이즈, 데이터 마이닝, AL/ML 혼합, - 행렬 분해기반 협업필터링

- 협업적 필터링은 기억기반/모델기반 등으로 구분되며 User 기반, Item 기반, 행렬 분해기반 협업 필터링을 주로 활용.

3. 잠재 요인(Latent Factor)을 활용하는 행렬분해(Matrix Factorization)기반 협업적 필터링 설명

가. 행렬분해기반 협업필터링 개념

개념	- 사용자-아이템 평점 행렬속에서 숨어있는 잠재요인을 추출하여 추천 예측하는 기법		
개념도	사용자 아이템 선호도	사용자 잠재변수 사용자 요인	잠재변수 아이템 아이템 요인
특징	잠재요인 분해	- 확인되지 않은 잠재요인 기반 분석	
	차원 분해	- 다차원 희소 행렬을 저차원 밀집 행렬로 전치	

- 행렬분해는 사용자-아이템 평점 행렬과 같은 다차원 행렬을 저차원 행렬로 분해하는 것을 의미

나. 행렬분해기반 협업 필터링 상세 기법

기법	구분	상세 설명
SGD	개념	- 고유값 분해와 같은 행렬을 대각화 하는 기법
	프로세스	1. User Latent 와 Item Latent 을 임의로 초기화 2. Gradient Descent 진행 (regularization 적용) - Back Propagation 3. 모든 평점에 대해서 반복 (epoch: 1) 4. 2~3 과정을 10 번 반복 (epoch: 10) - total cost 떨어지며 값이 비슷하게 수렴. 5. 4 를 바탕으로 예측 (높은 평점을 추천, 마이너스는 추천 x)
	핵심요소	- 실제값과 예측값의 오류 최소화
	장점	- 매우 유연한 모델로 다양한 손실함수 사용 가능
	단점	- 매우 느린 수렴 속도
ALS	개념	- 두 행렬 중 하나를 고정시키고 다른 하나의 행렬을 순차적으로 반복하면서 최적화하는 기법
	프로세스	1. 초기 아이템, 사용자 행렬을 초기화 (?는 0 으로 설정) 2. 아이템 행렬을 고정하고 사용자 행렬을 최적화 3. 사용자 행렬을 고정하고 아이템 행렬을 최적화 4. 위 2~3 과정을 반복
	핵심요소	- 변수를 하나씩 고정하여 모델 최적화
	장점	- SGD 보다 수렴속도 빠름.
	단점	- Loss Squares 만 사용 가능

- 이외의 행렬 분해기법으로 SVD(Singular Vector Decomposition), NMF(Non-Negative Matrix Factorization)등이

있으며, 널(NULL) 값이 존재하지 않는 경우에는 SVD, 널(NULL)값이 존재하는 경우 SGD 기법을 주로 활용.

- 추천 시스템의 기법 선택에 따라 추천 품질에 영향을 주고, 추천시스템의 이슈인 Cold Start, 필터버블 등의 문제를 사전에 고려하여 적용 필요.

4. 추천 시스템의 비즈니스 적용시 고려사항

고려사항	대응 방안
Cold Start 문제	- Hybrid Filtering, 인기 아이템 추천
아이템 쓸림 현상	- Recommendation Engine, - Recommendation + Discovery Engine (비인기 상품 추천 기능 추가)
신뢰성 문제	- 추천 과정의 블랙 박스화 (추천하게된 고객 행동 스토리)
필터버블 문제	- Flip Feed, RAA(Read Across the Aisle)을 통한 정보 제공 다양화 시도

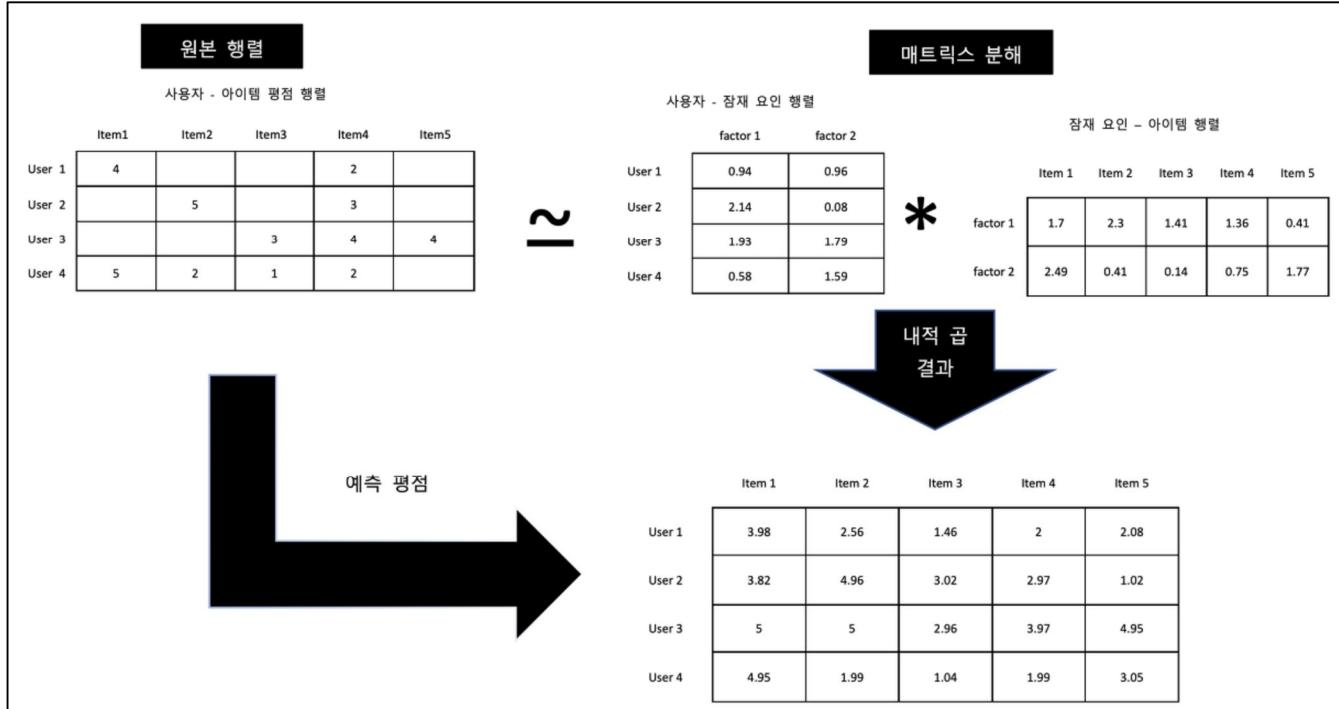
- 최근 의도적 AI 기반 추천 시스템의 데이터 왜곡, 의도적 공격등 편향 우려 증가, XAI, 데이트 필터링 등 추천 정확성 및 다양성 확보를 위한 노력 필요.

"끝"



[참고자료]

행렬 분배기반 협업 필터링 프로세스



기출풀이 의견

- 추천시스템의 컨텐츠 기반과 협업 필터링은 기본 토픽이지만 행렬분해기반 협업 필터링은 모의고사의 3단락에서 간략히 소개하는 정도여서 작성하기 쉽지 않은 문제였습니다. 다만 행렬분해기반 협업 필터링의 SVD, NMF, SGD, ALS, 잠재요인 등의 키워드를 기술했다면 고득점이 기대됩니다.

3. 캐시 메모리(Cache Memory)에 대하여 다음을 설명하시오.

가. 캐시 메모리의 개념과 구조

나. 지역성(Locality)의 개념과 유형

다. 캐시 일관성(Coherence)문제의 원인과 해결 방법

출 제 영 역	컴퓨터 구조	난 이 도	★★☆☆☆
출 제 배 경	- 컴퓨터 구조 영역의 고전토픽으로 캐시의 개념, 원리, 캐시 문제 및 해결까지 캐시의 전체영역과 인사이트 확인		
출 제 빈 도	- 98 회 3 교시, 101 회 응용 1 교시, 123 회 응용 3 교시, 모의고사/합숙 다수 출제		
참 고 자 료	<ul style="list-style-type: none"> - 68 회, 98 회 모의고사 - 96 회, 114 회 대비 합숙 - https://jaehyeon48.github.io/computer-architecture/cache-memory/ - https://slideplayer.com/slide/1508898/ 		
Key word	<ul style="list-style-type: none"> - 지역성 원리, 빠른 접근 속도 기억장치, Set/Line/Block, 시간적, 공간적 지역성, 변경가능한 데이터 공유, I/O, 프로세스 이주, 공유캐시 사용/공유캐시 미사용/스누피 제어기/디렉토리 기반, VI 프로토콜, MESI 프로토콜 		
풀 이	김대원(130 회 정보관리기술사 / wdw4004@naver.com)		

1. 캐시 메모리의 개념과 구조 설명

가. 캐시 메모리의 개념

개념	<ul style="list-style-type: none"> - 프로그램 수행시 지역성(Locality)의 원리를 이용하여 메모리나 디스크에서 사용되었던 데이터를 별도 기억공간에 보관하여 재사용시 빠르게 접근할 수 있도록 하는 기억장치 		
개념도	<p>Fast</p> <p>Slow</p> <p>워드 전송</p> <p>블록 전송</p> <p>주기억장치</p> <p>지역성(Locality)</p> <p>사상(Mapping)</p> <p>"주기억장치에 비해 5~10 배정도 빠른 접근속도"</p>		

- 자주 사용되는 명령들을 저장하고 있다가 중앙처리장치에 빠른 속도로 제공

나. 캐시 메모리의 구조 설명

캐시 메모리 구조	<p>The diagram illustrates the cache memory structure. It shows multiple sets of cache lines, each containing multiple lines. A specific line is selected by the set number and line number. The block format at the bottom shows fields for valid bit (v), tag, offset (o), and byte addresses (1, 2, ..., B-1). The total cache size is calculated as $S \times E \times B$.</p>	
	$\text{캐시메모리 전체 크기} = S \times E \times B$	
구성요소	- Set(S)	- 집합의 개수
	- line(E)	- 각 집합에 들어있는 Line의 수
	- Block(B)	- 각 라인에 들어있는 블록의 크기

- 캐시 메모리 블록은 Valid bit, Tag, 캐시 블록으로 구성되며, 일정시간에 데이터를 집중사용하는 데이터 지역성을 활용하여 빠른 접근성 제공

2. 캐시 적중율 극대화 특성, 지역성(Locality)의 개념과 유형 설명

가. 지역성의 개념

현상 측면	- 실행중인 프로세스가 일정 시간에 기억장치의 특정 부분(데이터, 프로그램 코드)을 집중적으로 액세스하는 참조 특성
활용 측면	- 한번 참조한 데이터는 다시 참조될 가능성이 높고, 참조된 데이터 주변의 데이터 역시 같이 참조될 가능성이 높은 현상

- 지역성은 시간적, 공간적 지역성으로 구분되며, 경우에 따라 순차적 지역성으로 추가 구분.

나. 지역성의 유형 설명

유형	구분	설명
시간적 (Temporal) 지역성	개념	- 프로세스가 실행되면서 하나의 페이지를 일정 시간 동안 집중적으로 액세스 하는 현상
	개념도	<p>The diagram shows a CPU [CPU] containing a 'Loop 문 i++' block and a Memory [Memory] containing a '변수i' block. An arrow labeled '동일 주소 반복 참조' points from the CPU to the Memory, illustrating how the same memory location is accessed repeatedly within a loop.</p>
	발생 CASE	- Loop, Stack, Sub Routine, Counting, Totaling 사용 변수
	활용 방안	- 한 번 참조한 페이지는 가까운 시간 내에 계속 참조할 가능성이 높은 특성 활용, 특정 시간 내 참조 빈도 기반 Paging 설계
공간적	개념	- 프로세스 실행 시 일정 위치의 페이지를 집중적으로 액세스하는 현상

(Spatial) 지역성	개념도	
	발생 CASE	<ul style="list-style-type: none"> - 배열 순회, 순차적(Sequential) 코드 실행 - 분기 미 발생 시 저장 순서대로 인출되는 순차적 지역성 포함
	활용 방안	<ul style="list-style-type: none"> - 한번 접근한 데이터와 인접한 주소를 계속 참조할 가능성이 높은 특성 활용, 인접 공간 포함한 블록단위 로딩 수행

- 지역성을 이용한 캐시메모리의 Write back 기반 캐시 쓰기 수행시 캐시-기억장치 일관성 문제 발생 가능.

3. 캐시 일관성(Coherence)문제의 원인과 해결 방법

가. 캐시 일관성(Coherence)문제의 원인

원인	개념도	설명
변경 가능한 데이터의 공유 (sharing of writable data)		<ul style="list-style-type: none"> - P1이 쓰기동작시 캐시에만 X'를 저장한 경우 - P2가 이미 읽은 X값과 X'값이 불일치현상 발생. - P1 캐시의 데이터와 주기억장치의 데이터값이 불일치현상 발생.
입출력 동작 (I/O activity)문제		<ul style="list-style-type: none"> - 주변 장치(Peripheral device)의 입출력 연산 - DMA 주변 장치의 데이터 수정 및 업데이트 - I/O 동작에 의한 데이터 불일치 현상 발생
프로세스 이주에 의한 데이터 불일치		<ul style="list-style-type: none"> - 프로세스가 변수 X에 엑세스하여 프로세서 캐시에 저장된 다음 다른 프로세서로 이동하여 X를 수정하면 데이터 불일치현상 발생. - CPU 부하 감소를 위한 프로세스 재배치 스케줄링

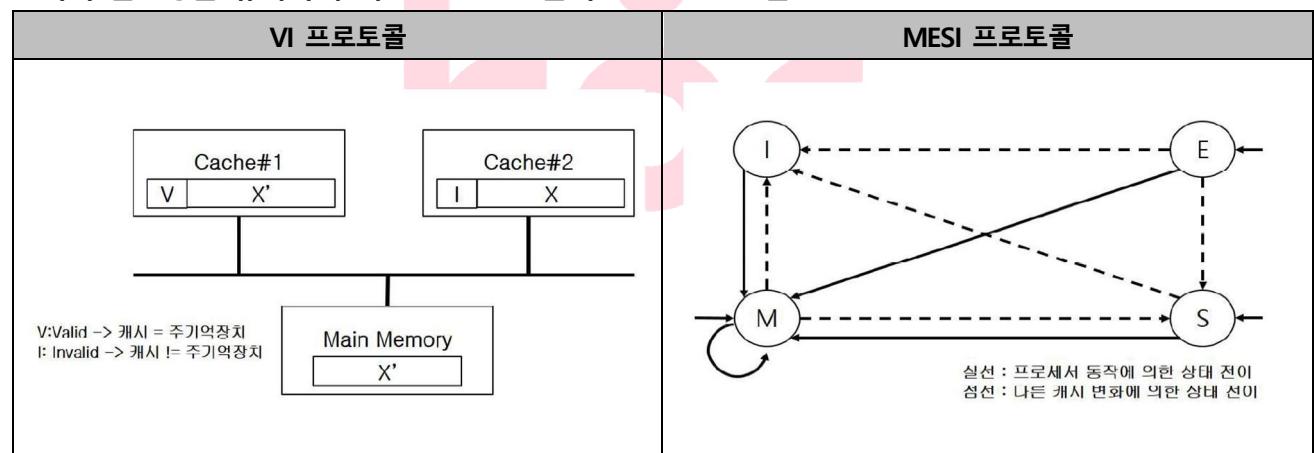
- 단일 프로세스에서는 Write Back에서 캐시 일관성 문제가 발생하고 멀티 프로세스에서는 Write Through, Write Back 모두에서 캐시 일관성문제 발생 가능.

나. 캐시 일관성(Coherence)문제의 해결 방법

구분	해결 방법	설명
SW 측면	공유 캐시 사용	<ul style="list-style-type: none"> - 모든 프로세서가 하나의 캐시를 사용하는 기법 - 간단한 구조, 메모리 충돌로 성능 저하
	공유변수 캐시 미사용	<ul style="list-style-type: none"> - 공유변수는 캐시에 저장하지 않는 기법 - 성능 저하 고려 필요
HW 측면	스누피 제어기	<ul style="list-style-type: none"> - 주소버스 감시, 캐시 접근 검사 다른 캐시 쓰기 발생 시 정보 무효화 - <u>Write Through</u>: 캐시 수정 메모리 갱신 - <u>Write Back</u>: 캐시 수정 메모리 갱신 X
	디렉토리기반 캐시이용	<ul style="list-style-type: none"> - 캐시 블록 공유 상태 저장하는 공간 이용 - <u>Full map</u>: 중앙집중식, 복사본 포인터 - <u>Limited</u>: Full map을 작게 유지, 메모리 효율화 - <u>Cache</u>: 포인터를 Linked list로 연결

- 캐시 쓰기 정책으로 Write-through 는 VI(Valid, Invalid) 프로토콜이 활용되며, Write Back 은 MESI 프로토콜 등이 활용.

4. 캐시 일관성을 유지하기 위한 VI 프로토콜과 MESI 프로토콜



- VI 프로토콜은 주기억장치와 일치 여부를 나타내는 플래그를 공유하며, MESI 프로토콜은 수정, 배타, 공유, 무효상태를 나타내는 플래그를 공유하여 캐시 일관성을 유지하도록 역할 수행.

"끝"

[참고자료]

캐시 메모리 계층 구조



캐시 메모리 일관성 제어를 위한 세부 기법의 고려사항

세부 기법	설명
공유 캐시 비활성화	<ul style="list-style-type: none"> - 프로세스 응답 속도와 지연 시간(Latency)이 증가 - 소형기기 및 소용량 데이터 처리환경에 적합
상호 배제	<ul style="list-style-type: none"> - 임계구역 설정에 따른 CPU 부하 증가 - 공유자원의 제어를 위한 알고리즘(뮤텍스, 세마포어, Busy Waiting) 필요
버스 감시 제어기	<ul style="list-style-type: none"> - 코어 개수 증가에 따른 캐시 상태정보 트래픽의 증가 가능성 - 중소규모의 시스템 환경에 적합하며 낮은 지연 시간을 보장
디렉토리 기반	<ul style="list-style-type: none"> - 버스감시 기법의 상태정보 트래픽 증가에 따른 단점을 보완 - 대규모 환경에 캐시 상태를 전역적으로 통합 관리하여 확장이 용이

- 수백, 수천단위 멀티코어 환경에서 디렉토리 기반 기법은 성능과 지연시간에 대한 문제점과 최적화 필요

Cache Write Back 일관성 유지위한 프로토콜 설명

유형	설명
MESI	<ul style="list-style-type: none"> - 수정, 배타, 공유, 무효 상태로 관리
MEI	<ul style="list-style-type: none"> - 공유상태가 없기 때문에 2개 이상의 프로세서에서 데이터를 읽을 때 비용이 큼 - PowerPC 755 계열에서 사용
MSI	<ul style="list-style-type: none"> - 처음 Read되어도 공유상태가 됨. - 새로운 프로세서의 Write에 대해 무효화 신호를 날려야 함
MOESI	<ul style="list-style-type: none"> - O(Owned)상태: 변경 상태의 캐시 블록을 다른 프로세서에서 읽을 경우, O상태로 변경되며 캐시간 전송, 재시도처럼 메모리를 갱신하지 않음 - 다른 프로세서의 Read요청때마다 메모리에 쓰는 Latency를 줄임 - AMD Opteron, UltraAPARC에서 사용

MESIF	<ul style="list-style-type: none"> - F(Forwarding): 다수 프로세서에 공유된 캐시 블록에 접근하는 새로운 프로세서에게 대표로 전달해주는 프로세서, 전달후에는 F상태를 넘겨줌 - Intel Nehalem에서 사용
-------	---

캐시 메모리 쓰기 정책

구분	Write Through	Write Back
개념도	<pre> graph TD CPU[중앙처리장치] -- 동시에 write --> MM[주기억장치] CPU -- 동시에 write --> CM[캐시 메모리] </pre>	<pre> graph TD CPU[중앙처리장치] --> MM[주기억장치] MM -- swap-out 시 write --> CM[캐시 메모리] </pre>
개념	<ul style="list-style-type: none"> - CPU의 연산 결과를 주기억장치와 캐시에 동시에 쓰는 방식 	<ul style="list-style-type: none"> - CPU의 연산 결과를 캐시에는 쓰고 주기억장치에는 나중에 쓰는 방식
장점	<ul style="list-style-type: none"> - 단순한 구조의 아키텍처 - 캐시와 주기억장치의 일관성 유지 	<ul style="list-style-type: none"> - 쓰기(Write) 시간의 단축 - CPU의 메모리 접근 최소화
단점	<ul style="list-style-type: none"> - 쓰기 동작으로 인한 버스의 트래픽 증가 - 빈번한 주기억장치 접근으로 쓰기 시간의 증가 	<ul style="list-style-type: none"> - 상대적으로 복잡한 구조의 아키텍처 - 캐시 일관성(Cache Coherence) 문제 발생

기출풀이 의견

3. 강화학습을 심도있게 유형과 특징을 구분하지 못하면 기술하기 어려운 문제입니다. 반대로 Model Free 기반 유형에 대해서 알고 있었다면 고득점이 가능한 문제입니다. 또한 유사출제에 대비하여 Model Base 강화학습까지 학습하시기 권장드립니다.

문제 4. 스마트홈 연결 표준인 매터(Matter)에 대하여 다음을 설명하시오.

가. 매터의 개념

나. 기존 스마트홈 표준의 한계점과 매터의 장점

다. 매터 Network Stack

출제영역	네트워크	난이도	★★☆☆☆
출제배경	- 매터보급 확산과 128회 정보통신 기술사 출제에 따른 교체 출제		
출제번호	- 128회 정보통신 기술사, 모의고사, 주간모의고사 다수 출제		
참고자료	- 2021년~2023년 모의고사 - 정보통신기획평가원(IITP), "개방형 스마트홈 연동 표준 매터의 주요 특징과 시장 전망"		
Keyword	- IP 기반 개방형, 멀티어드민, 로컬제어, 단순성, 상호연동성, 신뢰성, 안전성, IEEE 802.11, IEEE 802.15.4, Wi-Fi, Thread, IPv6, BLE, 매터허브, 매터브리지		
풀이	김대원(130회 정보관리기술사 / wdw4004@naver.com)		

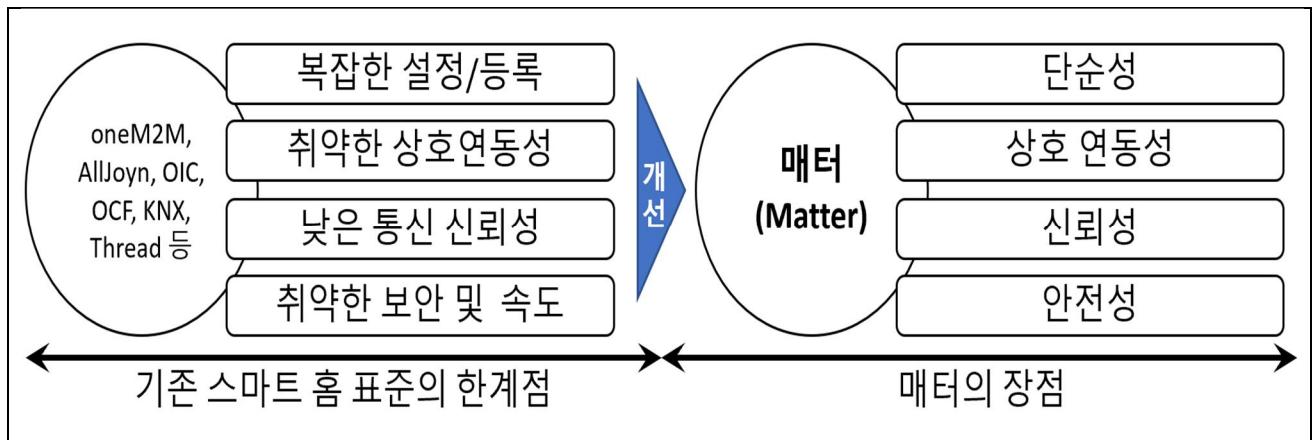
1. IP 기반 개방형 스마트홈 연결 표준, 매터(Matter) 개념

개념	<ul style="list-style-type: none"> - 스마트홈의 상호 연동성, 신뢰성, 안전성 제공위해 IP 프로토콜, 와이파이, 쓰레드 등의 기본 통신 프로토콜기반 개방형 스마트홈 기기 연동 표준 										
개념도	<p>“ 플랫폼, 허브, 디바이스로 구성 ”</p>										
특징	<table border="1"> <tr> <td>멀티 어드민</td> <td> <ul style="list-style-type: none"> - 모든 디바이스들에 고유한 IP기반 글로벌 ID체계 활용 - 여러 플랫폼 동시 활용 및 반영 가능 </td> </tr> <tr> <td>넓은 통신 범위 지원</td> <td> <ul style="list-style-type: none"> - 쓰레드기반 메시 네트워크 구성으로 네트워크 커버리지 확장 - IP 기반 D2D, D2C 통신 가능하여 심리스(seamless) 가능 </td> </tr> <tr> <td>저/고속 통신 지원</td> <td> <ul style="list-style-type: none"> - 와이파이기반 고속 통신 지원 - 쓰레드 기반 저속 저전력 통신 지원 </td> </tr> <tr> <td>로컬 제어</td> <td> <ul style="list-style-type: none"> - 클라우드가 아닌 로컬에서 디바이스 간 연동 자동화 - 동작 속도 개선 및 개인 프라이버시 보호 가능 </td> </tr> </table>			멀티 어드민	<ul style="list-style-type: none"> - 모든 디바이스들에 고유한 IP기반 글로벌 ID체계 활용 - 여러 플랫폼 동시 활용 및 반영 가능 	넓은 통신 범위 지원	<ul style="list-style-type: none"> - 쓰레드기반 메시 네트워크 구성으로 네트워크 커버리지 확장 - IP 기반 D2D, D2C 통신 가능하여 심리스(seamless) 가능 	저/고속 통신 지원	<ul style="list-style-type: none"> - 와이파이기반 고속 통신 지원 - 쓰레드 기반 저속 저전력 통신 지원 	로컬 제어	<ul style="list-style-type: none"> - 클라우드가 아닌 로컬에서 디바이스 간 연동 자동화 - 동작 속도 개선 및 개인 프라이버시 보호 가능
멀티 어드민	<ul style="list-style-type: none"> - 모든 디바이스들에 고유한 IP기반 글로벌 ID체계 활용 - 여러 플랫폼 동시 활용 및 반영 가능 										
넓은 통신 범위 지원	<ul style="list-style-type: none"> - 쓰레드기반 메시 네트워크 구성으로 네트워크 커버리지 확장 - IP 기반 D2D, D2C 통신 가능하여 심리스(seamless) 가능 										
저/고속 통신 지원	<ul style="list-style-type: none"> - 와이파이기반 고속 통신 지원 - 쓰레드 기반 저속 저전력 통신 지원 										
로컬 제어	<ul style="list-style-type: none"> - 클라우드가 아닌 로컬에서 디바이스 간 연동 자동화 - 동작 속도 개선 및 개인 프라이버시 보호 가능 										

- 다른 벤더에서 생산한 기존 스마트홈 기기간 통신 및 제어의 어려움, 취약한 보안등의 이슈를 해결하기 위해 매터 등장.

2. 기존 스마트홈 표준의 한계점과 매터의 장점 설명

가. 기존 스마트홈 표준의 한계점과 매터의 장점



- 기존 IoT 스마트홈 디바이스의 벤더별 표준이 존재하고 벤더간 상호연동의 한계를 극복하기 위해 IP 기반 매터 표준 제정.

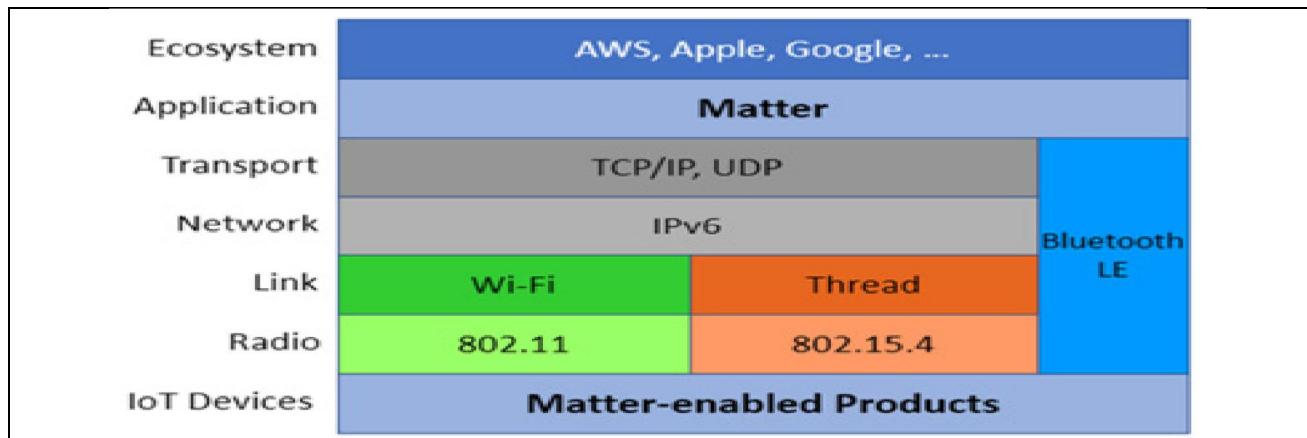
나. 기존 스마트홈 표준의 한계점과 매터의 장점 상세 설명

구분	핵심 내용	상세 설명
기존 스마트홈 표준의 한계점	복잡한 설정/등록	- 제조사별 상이한 디바이스 등록 및 복잡한 설정 절차
	취약한 상호연동성	- 타 브랜드 디바이스 간 상호연동 불가
	낮은 통신 신뢰성	- 다양한 통신 프로토콜 혼합 사용에 따른 신뢰성 저하
	취약한 보안 및 속도	- 인터넷기반 반응 속도 저하, 보안 취약점 노출
매터의 장점	단순성	- 벤더, 제조사별 디바이스들이 단일한 방식으로 디바이스 등록과 설정
	상호 연동성	- 서로 다른 제품, 디바이스들끼리 상호 연동, 다양한 스마트홈 서비스 제공
	신뢰성	- 단일화된 통신 프로토콜을 이용해 스마트 홈 네트워크를 구성
	안전성	- 빠른 반응 속도와 높은 보안성 제공

- 인터넷 프로토콜(IP) 기술을 이용해서 스마트홈을 구성하는 것으로 단순하며 신뢰할 수 있고 확장성이 뛰어나며 안전한 스마트홈 구축을 목표로 함

3. 매터 Network Stack 설명

가. 매터 Network Stack



- IPv6, Wi-Fi 등 프로토콜과 매터허브, 매터 브리지등 Application으로 구성.

나. 매터 Network Stack 구성요소

구분	구성요소	설명
프로토콜	IPv6	- 가정 내의 모든 디바이스들에 고유한 IP 주소 부여
	Wi-Fi	- IEEE 802.11 표준기반 고속 데이터 통신을 위한 활용
	Thread	- IEEE 802.15.4 표준 기반 저전력 디바이스 간의 통신 지원
	BLE	- 저전력 디바이스의 등록, 연결위해 활용
기술요소	멀티 어드민	- 하나의 장치를 여러 플랫폼에 동시 등록 사용
	로컬 제어	- 디바이스의 상태 확인 및 제어, 연결을 클라우드가 아닌 로컬에서 수행
	매터 허브	- 디바이스를 동시에 여러 플랫폼에 등록 사용하기 위한 플랫폼별 허브 장치
	매터 브리지	- 타 표준에 속한 기기간 상호 호환성 지원
	OTA(Over The Air)	- 무선통신으로 소프트웨어 업그레이드
	응용 계층	- 서비스 플랫폼 기반 표준 규격 - 오픈소스로 공개(Apache 2.0)

- 매터는 IoT 디바이스 특성에 따라 저/고속 무선망 통신지원 및 안정성 제공.

4. 매터 표준화에 따른 스마트홈 시장 경쟁력 확보 방안

방안	설명
차별화 전략	- 디바이스 제조사는 경쟁사 대비 시장 차별화된 전략 개발
맞춤형 서비스	- 인공지능 기반 사용자 맞춤형 및 예측 기반 지능형 서비스 제공
Servitization	- 생활 서비스와 결합하는 디바이스의 서비스화(Servitization)
크로스 플랫폼 전략	- 멤버십 서비스 기반 크로스 플랫폼 전략 수립

- 글로벌 기업들은 스마트홈 디바이스기반의 다양한 생활 서비스를 제공 중이며 매터 표준을 활용한 경쟁력 확보위해 단순 디바이스 제조사 지원뿐만 아니라 서비스 기업 지원으로 동시 성장 고려된 정책 마련 필요

"끝"

[참고자료]

매터 표준의 예상 기대효과

기대효과	설명
사용자 중심 서비스 전환 가속	<ul style="list-style-type: none"> - 기존 플랫폼 사업자 중심에서 기반 사용자 중심 서비스로 전환 - 디바이스 전원, QR 코드 스캔으로 여러 플랫폼 서비스 즉시이용 가능 - 상태, 동작은 로컬 관리로 동작 속도와 프라이버시 보호 가능
디바이스 제조사 경쟁력 제고	<ul style="list-style-type: none"> - 개방형 표준으로 디바이스 제조사들의 부담 경감 - 하나의 디바이스로 크로스 플랫폼 전략 지원 가능 - 제품 개발 시간, 비용, 노력 절감으로 수익성과 경쟁력 제고
지능형 생태계 활성	<ul style="list-style-type: none"> - 단순한 조건부 자동화에서 인공지능 활용도 증대 기대 - 인공지능 기반 사용자 맞춤형, 예측 기반 지능형 서비스 확대 - 사용자 데이터와 생활 서비스 결합으로 디바이스의 서비스화 기대



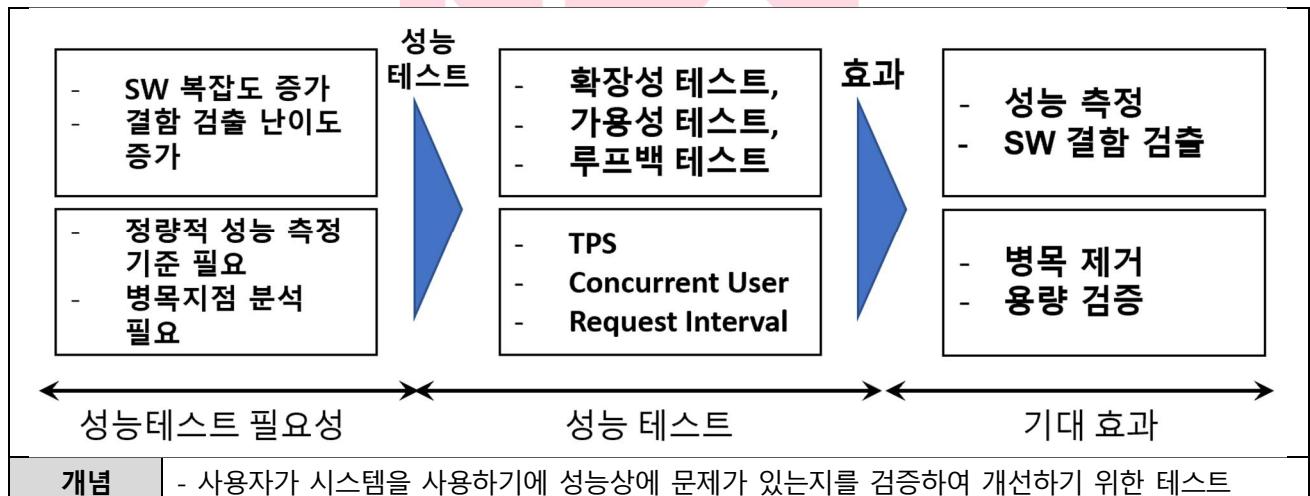
기출풀이 의견

4. 2021년부터 모의고사에서 지속적으로 출제되고 128회 정보통신 기술사에도 출제되어 비교적 난이도가 어렵지 않은 문제였습니다. 키워드위주로 주어진 문제의 답안을 작성하고 고득점 획득을 위해 마지막 단락에 인사이트 제시가 필요합니다.

문 제	5. 시스템 성능 테스트 수행 시, 성능 지표 관련하여 다음 물음에 답하시오.
	가. 응답시간(Response Time), 대기시간(Think Time), 동시사용자(Concurrent User), 활성사용자(Active User), TPS(Transaction Per Second)를 설명하시오. 나. 동시사용자(Concurrent User)가 100 명이고, 응답시간(Response Time)이 5 초이내이며, 대기시간(Think Time)이 15 초인 시스템의 경우, TPS를 구하시오.

출 제 영 역	소프트웨어공학	난 이 도	★★☆☆☆
출 제 배 경	- 소프트웨어공학 고전토픽으로 계산문제를 추가하여 응용능력 여부 확인		
출 제 빈 도	- 95 회, 98 회 외 모의고사 다수 출제		
참 고 자 료	<ul style="list-style-type: none"> - 95 회 정보관리 - https://dewble.tistory.com/entry/concept-of-performance-test - 116 회 합숙자료 		
Key word	- 목적(단위, 복합, 임계), 방법(확장성, 가용성, 루프백테스트), TPS=Concurrent User / (Response Time + Think Time)		
풀 이	김대원(130 회 정보관리기술사 / wdw4004@naver.com)		

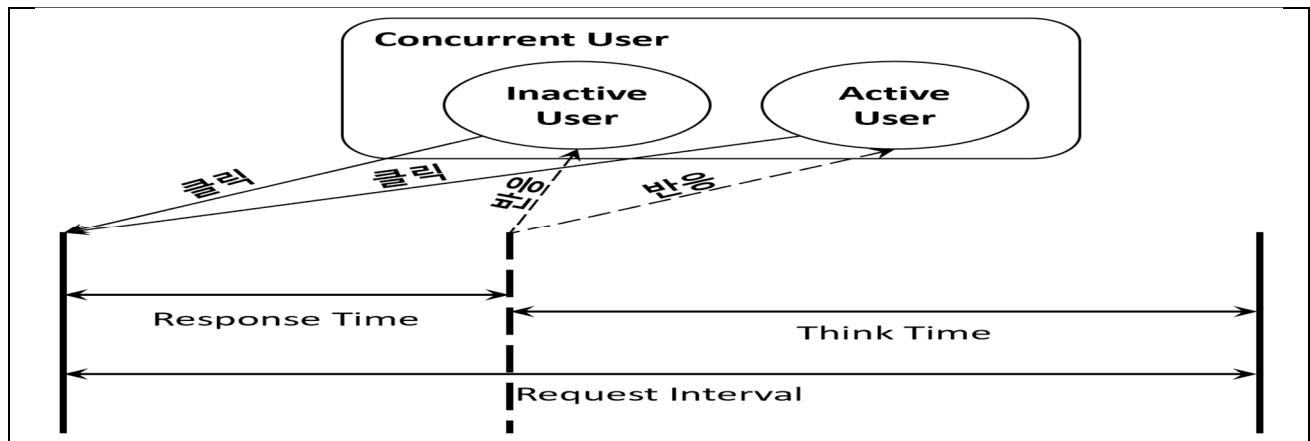
1. 시스템이 허용할 수 있는 최대 사용자 분석, 성능테스트의 개요



- 성능 개선과 안정적 서비스 제공을 위해 성능테스트를 통한 정량적 성능 분석 및 서비스 가용성 검증 필요.

2. 성능 테스트 정량적 주요지표, 응답시간(Response Time), 대기시간(Think Time), 동시사용자(Concurrent User), 활성사용자(Active User), TPS(Transaction Per Second) 설명

가. 성능 테스트 주요 지표 구성



- 주요 성능지표인 TPS 분석을 위해 Request Interval Time, Concurrent User 등이 필요.

나. 응답시간(Response Time), 대기시간(Think Time), 동시사용자(Concurrent User), 활성사용자(Active User), TPS(Transaction Per Second) 상세 설명

구분	지표	설명
시간 지표	응답시간 (Response Time)	- 요청으로 응답을 받아 화면에 표현할 때까지 시간
	대기시간 (Think Time)	- 사용자가 생각하는 시간 (서버 Idle 타임)
사용자 지표	동시사용자 (Concurrent User)	- 특정 시점 접속사용자 (Active User + Inactive User)
	활성사용자 (Active User)	- 특정 요청을 보낸 다음 응답을 기다리는 사용자
처리량	TPS (Transaction Per Second)	- Transaction Per Second(초당 처리가능한 트랜잭션) - AU (Active User) / MRT (Mean Response Time)

- TPS 분석을 위해 Concurrent User, Active User, Response Time, Think Time 지표 필요.

3. TPS 문제 풀이

가. TPS 계산위한 지표 및 수식 설명

구분	지표	수식
시간 지표	Request Interval	= Response Time + Think Time
	Concurrent User	= Active User + Inactive User = Active User + (TPS * Think Time)
사용자 지표	Active User	= Concurrent User * Response Time / Request Interval Time = Concurrent User * Response Time / (Response Time + Think Time) = TPS * Average Response Time

처리량	TPS	= Concurrent User / Request Interval = Concurrent User / (Response Time + Think Time) 또는, = Active user / Response Time
-----	-----	--

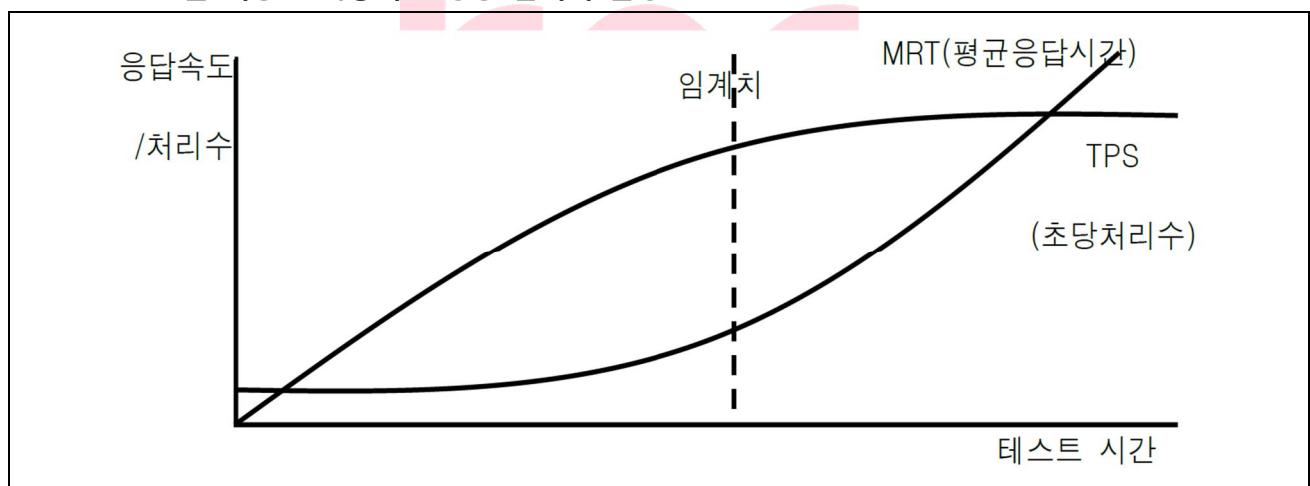
- 주어진 문제에서 TPS 계산을 위한 Concurrent User, Response Time, Think Time 제시

나. TPS 문제 계산

구분	내용
문제	<ul style="list-style-type: none"> - 동시사용자(Concurrent User) = 100 명 - 응답시간(Response Time) = 5 초 - 대기시간(Think Time) = 15 초인 시스템의 TPS 계산.
풀이	$\begin{aligned} \text{TPS} &= \text{Concurrent User} / \text{Request Interval time} \\ &= \text{Concurrent user} / (\text{Response Time} + \text{Think Time}) \\ &= 100 / (5 + 15) = 5 \end{aligned}$
결과	<ul style="list-style-type: none"> - TPS = 5 - 1 초당 5 개의 트랜잭션 처리

- TPS 가 클수록 성능이 좋은 SW이며 Little's Law 를 통해 이상적 성능 임계치 분석 가능.

4. Little's Law 를 적용한 이상적인 성능 임계치 설명



- 사용자가 증가함에 따라서 응답시간이 느려지고 TPS는 증가하며, TPS가 더 이상 증가하지 않고 완만하게 되는 시점이 그 시스템의 임계치임

"끝"

[참고자료]

정확한 성능 테스트를 위한 고려사항

고려사항	설명
표준화 기반 프로세스 체계화	- TMMi, ISO29119 기반 지속적 개선가능 역량 및 테스트 프로세스 구축
정확한 거래모델	- 실제환경에서 발생하는 트랜잭션에 대한 정확한 분석을 통한 트랜잭션모델 필요
트랜잭션 변화분석	- 시간대별 또는 환경에 따라 변화하는 트랜잭션 분석, 사용자 사용수준에 따라 변화하는 트랜잭션 분석
UnKnown 트랜잭션분석	- 개발한 소프트웨어 이외의 백신이나 윈도우 업데이트와 같은 타 SW 트랜잭션)의 영향 고려
성능 테스트툴 이용	- 테스트 정확성과 효율증가, 로그 생성 및 분석



기출풀이 의견

5. 성능테스트는 고전 토픽이며 주어진 계산문제도 TPS계산 방법을 숙지하고 있었다면 간단하게 기술할 수 있는 문제입니다. 주요지표에 대한 설명과 문제풀이 그리고 Little's Law를 통합하여 인사이트를 제공하면 고득점이 예상됩니다.

6. 공공기관 정보화 사업 추진 시, 상용 SW 직접구매 제도와 관련하여 다음을 설명하시오.

문제 제 1. 상용 SW 직접구매 적용대상

나. 상용 SW 직접구매 예외기준

다. 상용 SW 직접구매와 일괄발주 비교

출제영역	소프트웨어공학	난이도	★★☆☆☆
출제배경	- 분리발주 개념이 상용 SW 직접구매로 변경되면서 해당 개념이해 여부 파악		
출제빈도	<ul style="list-style-type: none"> - 2022년 응용 2회 출제 - 117회, 126회 합숙 자료 		
참고자료	<ul style="list-style-type: none"> - 상용 SW 직접 구매 가이드 - 117회, 126회 합숙 자료 - 110회 모의고사 		
Keyword	- 상용 SW 별도 발주, SW 진흥법 제 54 조, 3 억원, 5 천만원 이상, 민간투자형 SW 사업, 현저한 비용/지연/효율저하,		
풀이	김대원(130회 정보관리기술사 / wdw4004@naver.com)		

1. 상용 SW 직접 구매제도의 개념

개념	<ul style="list-style-type: none"> - 발주기관이 공공 정보화사업 추진 시 HWSW 구매, 시스템통합 등의 사업에 상용소프트웨어 구매를 포함하여 발주하는 것이 아니라, 상용소프트웨어만을 별도로 발주, 평가·선정, 계약하는 방식으로 직접구매(조달 쇼핑몰 구매 등 포함)하는 것을 말함 (구, 분리발주 제도) 		
개념도			
법적 근거	<ul style="list-style-type: none"> - 소프트웨어 진흥법 제 54 조 (국가기관 등의 상용 소프트웨어 구매) - 소프트웨어 사업 계약 및 관리감독에 관한 지침 제 7 조, 제 8 조 		

- 2020년 12월 소프트웨어 계약 및 관리감독에 관한 지침 전부 개정에 따라 분리발주 1차 조건의 사업규모가 5억에서 3억으로 변경.

2. 상용 SW 직접구매 적용대상 및 상용 SW 직접구매 예외기준 설명

가. 상용 SW 직접구매 적용대상

구분	설명
1차 조건	- 3 억원 이상(VAT 포함) 사업
2차 조건	<ul style="list-style-type: none"> - 조달청 종합쇼핑몰 등록 SW 포함된 경우(5 천만원 미만 포함) - 5 천만원 이상 또는 동일 SW 다량 구매로 5 천만원을 초과하는 경우에는 5 천만원 이상인 소프트웨어로 간주 - SW 품질 인증(GS), CC, NEP, NET 및 국가정보원 검증/지정 SW 가 포함된 경우
법적 근거	<ul style="list-style-type: none"> - 소프트웨어 진흥법 제 54 조(국가기관등의 상용소프트웨어 구매) - 소프트웨어사업 계약 및 관리감독에 관한 지침 제 7 조(상용소프트웨어 직접구매 대상)

- 1차 조건 충족 시 2차 조건 중 어느 하나라도 해당하는 경우 상용 SW 직접구매 대상 사업.

나. 상용 SW 직접구매 예외기준

구분	예외 기준	설명
대상 사업 제외	민간투자형 SW 사업	<ul style="list-style-type: none"> - 국민생활의 편익을 증진하기 위하여 민간의 자본과 기술을 활용하여 공공부문과 민간부문이 협력하여 추진하는 소프트웨어사업
대상 SW 제외	현저한 비용상승 현저한 지연 현저한 비효율성	<ul style="list-style-type: none"> - 기존 정보시스템이나 새롭게 구축하는 정보시스템과 통합이 불가능하거나 현저한 비용상승이 초래되는 경우 - 해당 사업이 사업기간 내에 완성될 수 없을 정도로 현저하게 지연될 우려가 있는 경우 - 행정업무 증가 외에 소프트웨어 제품을 직접 구매하여 공급하는 것이 현저하게 효율적이라고 판단되는 경우
법적 근거	<ul style="list-style-type: none"> - 소프트웨어 진흥법 제 40 조(민간투자형 소프트웨어사업) - 소프트웨어사업 계약 및 관리감독에 관한 지침 제 8 조(상용소프트웨어 직접구매의 제외) - 국가를 당사자로 하는 계약에 관한 법률 시행규칙 제 84 조(소프트웨어사업에 대한 소프트웨어의 관급) - 지방자치단체를 당사자로 하는 계약에 관한 법률 시행규칙 제 87 조(소프트웨어 제품 및 콘텐츠 제작의 관급) 	

- 경쟁입찰로 상용 SW 직접구매시 BMT 실시 및 평가 반영하나 조달청 등록 제품은 BMT 수행제외 가능.

3. 상용 SW 직접구매와 일괄발주 비교

가. 상용 SW 직접구매와 일괄발주 개념 비교

상용 SW 직접구매	일괄발주
<ul style="list-style-type: none"> - 일괄발주 형태에서 SW 구매만을 별도로 분리하여 발주(조달구매 포함)하고, 평가, 선정, 관리 등을 실시하는 형태의 구매 방법 	<ul style="list-style-type: none"> - 발주기관이 소프트웨어사업을 추진할 때 HW 구매, SW 구매, 시스템통합 등의 사업에 상용소프트웨어 구매를 포함하여 발주하는 방법

- 상용 SW 직접구매는 시스템 구축과 상용 SW 구매를 별도로 하기 때문에 복수의 사업자로 구성.

나. 상용 SW 직접구매와 일괄발주 상세 비교

비교	상용 SW 직접구매	일괄발주
사업 형태	- 시스템 설계/구축과 상용SW구매를 별도로 각각 발주하는 형태	- 시스템 설계/구축 상용SW구매, HW 도입 등 일괄하여 발주하는 형태
사업자 수	- 복수사업자(SW 사업자, 구축사업자)	- 단일사업자(설계/구현 통합)
시스템 통합 책임	- 복수사업자(SW 사업자, 구축사업자) 간 통합/하자 책임 분쟁 조정 역할자 필요	- 시스템 구축사업자가 시스템 통합에 대한 책임
발주행정 부담	- 복수 계약자 간 업무처리 및 업무부담 가중	- 단일사업자에 의한 수행으로 행정적 절차 및 부담 적음
사업자간 이슈	- SW 설치 사업자와 시스템 구축 사업자 간 성능, 연계, 일정 등 전체 관리 필요	- 사업자 선정 및 사업수행관점 에서는 단일사업자 수행으로 이슈 발생 적음
<ul style="list-style-type: none"> - 기존 일괄 발주로 공공에 도입되던 공개 SW 가 심사를 통해 일정 기준을 충족한 제품을 조달 등록해 두고 등록된 제품 간 경쟁을 통해 납품사를 결정하는 다수공급자계약으로 도입 추진됨 		

4. 상용 SW 직접 구매제도의 업무 프로세스

단계	절차	상세활동
1 단계	SW 사업 유형 및 사업비 파악	<ul style="list-style-type: none"> - SW 사업 총 사업규모 산정 후 대상 사업기준 확인(3 억원 이상) - 민간투자형 SW 사업인지 여부
2 단계	직접구매대상 SW 조사 및 검토	<ul style="list-style-type: none"> - 조달청 종합 쇼핑몰에 등록된 품목인지 확인 - 단일 SW 가격 5 천만원이상 또는 동종 SW 구매금액 합산이 5 천만원 초과여부 조사 - 도입대상 SW 품목이 국가인증 획득 여부 조사
3 단계	직접구매대상 SW 제외사유 사전검토	<ul style="list-style-type: none"> - 조달 발주하는 사업*의 경우 제외사유 적용 시 조달청에 미리검토 요청 - 자체발주 시 과업심의위원회 및 상위기관 등에 "미리검토" 요청
4 단계	SW 사업발주	<ul style="list-style-type: none"> - 상용 SW 직접구매 사업 및 본(통합) 사업의 발주 우선 순서 결정 - 상용 SW 직접구매 대상 SW 발주 방법 결정

- 경쟁 입찰공고(1 억원 이상)시 소프트웨어 품질성능평가(BMT) 실시 및 평가 반영.

"끝"

기출풀이 의견

6. 상용 SW 직접 구매는 고전토픽으로 정확한 답이 있는 문제입니다. "상용 SW 직접 구매 가이드"의 내용을 숙지하고 있고 추가적으로 상용 SW직접구매에 대한 마지막 단락 인사이트를 제공한다면 고득점을 기대할 수 있습니다.

[참고자료]

상용 SW 직접 구매 적용 대상

1차 조건	2차 조건
3 억원 이상 (VAT 포함) 사업	<ul style="list-style-type: none"> - 조달청 종합쇼핑몰 등록 SW 포함된 경우(5천만원 미만 포함) - 5 천만원 이상 또는 동일 SW 다량 구매로 5 천만원을 초과하는 경우에는 5 천만원 이상인 소프트웨어로 간주 - SW 품질 인증(GS), CC, NEP, NET 및 국가정보원 검증/지정 SW가 포함된 경우

상용 SW 직접구매 업무처리 흐름도

