무선 통신망의 에러를 고려한 이동 컴퓨팅의 성능 분석

°정승식, 김재훈
아주대학교 정보통신전문대학원

Analysis of Mobile Computing on Error Prone Wireless Link

°Seong-Sik Jung, Jai-Hoon Kim
College of Infromation and Communication
Ajou University

요 약

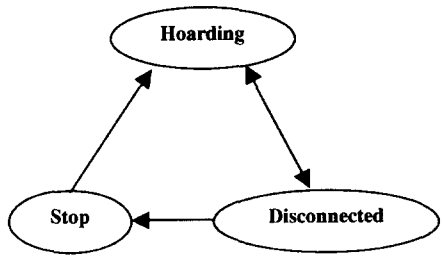
무선 네트웍을 기반으로 하는 모빌 컴퓨팅 기술은 빠르게 성장하고 있다. 무선 네트웍 환경은 특성상 잦은 끊김과 높은 에러율 때문에 모빌 컴퓨팅에서는 비연결 수행을 하는 것이 필요하다. 수많은 개념과 이론들이 이러한 모빌 컴퓨팅 환경에서 비연결 수행기능을 제공하기 위해 제안되고 있다. 본 논문에서 마코프(Marcov) 모델링 기법을 이용하여 모빌 컴퓨팅 환경에서 비연결 수행상태를 포함한 평균 수행시간을 시뮬레이션을 통하여 측정하였다. 모빌 컴퓨팅 환경은 데이터 호딩(Data Hoarding), 비연결 수행(Disconnected operation), 정지(Stop)의 3 가지 상태로 구성이 된다. 이러한 3 가지 상태에서 여러가지 입력 파라메터들(에러율(Error rate), 재연결율(Recovery rate), 호딩 오버헤드(Hoarding overhead), 재연결 오버헤드(Reintegration overhead))로 전체 수행시간을 분석하였다. 이러한 분석을 통해서 모빌 컴퓨팅에서 보다 효과적인 비연결 수행을 위한 방법을 선택할 수 있다.

1. 서론

모빌 컴퓨팅과 무선 네트웍은 매우 빠르게 발전하고 있는 기술이다. 무선 네트웍의 특성중에서 잦은 끊김과 높은 에러율로 인하여 모빌 컴퓨팅에서는 비연결 수행에 관한 해결책을 제공해야 한다. 무선 네트웍에서 끊김이 예상되면 프로세스(또는 데이터)는 비연결 상태에서도 수행이 가능하도록 모빌 호스팅쪽으로 데이터를 이전해야 한다. 이러한 비연결 수행상태로 가기전의 상태를 호딩 상태(Hoarding State[1])라 하며 호딩 상태에서 앞으로 사용하게 될 데이터를 어떻게 예상하는가는 모빌 호스트에서 매우 중요한 문제이다.

무선네트웍에 장애가 발생하면 모빌 호스트는 비연결 상태(Disconnected state)가 된다. 이 상태에서도 모빌 호스트는 이전의 호딩 상태에서 모빌 호스트로 가지고 온 데이터를 이용하여 수행을 계속 할 수 있다. 물론 이때 수행에 사용되는 데이터는 이전 호딩 상태에서 미리 로컬 호스트(모빌 호스트)쪽으로 가지고 온 데이터를 사용하여 수행되며 만약에 재연결이 이루어 지기 전에 모빌 호스트(로컬)내에 없는 데이터의 요구가 발생하면 모빌 호스트는 정지 상태(Stop state)로 변한다. 정지 상태에서는 아무일도 하지 않고 오직 무선 네트웍이 재연결이 되기만을 기다리며 무선 네트웍이 재 연결이 되면 비연결 상태에서 수행한 결과 및 사용했던 데이터중 수정된 것들에 대한 사항을 다른 연결된 호스트들에게 알리게 된다. 이때 재연결 오버헤드(Reintegration overhead)가 발생한다. 만약 비연결 상태에서 정지 상태로 가기 전에 무선 네트웍이 재연결이 되면 모빌 호스트는 정지 상태를 거치지 않고 재연결이 되어 재연결 오버헤드를 수행하고 연결상태가 되어 수행을 지속하게 된다. [1]

2. 관련 연구



<그림 1> 이동 컴퓨팅 task 의 세가지 상태

모빌 컴퓨팅 환경에서 이동 컴퓨팅의 수행의 상태는 위에서 설명한것과 같이 <그림 1>처럼 표현이 가능하다. 많은 개념들이 비연결 상태에서의 수행을 효과적으로 지원하기 위해서 제안되었으며[1,2,3,4,5] 이러한 분야는 크게 file system, Database, Web-browsing 의 3 가지 종류로 나눌 수 있다[1].

File system

비연결 상태에서의 수행을 지원하기 위해서 제안된 대부분은 확장된 캐쉬 관리를 도입하여 모빌 환경에 적용하여 이용을 한다. 연결 상태일 동안 앞으로 사용하게 될 파일을 미리 모빌 client 로 가지고 와서 비연결 상태가 되면 이를 이용한다. 이때 미리 가져오는 방법은 사용자에 의해서 명시적으로 지정된 파일을 가져오게 하는 방법과 자동적으로 지난 과거동안의 파일참조 정도를 파악하고 미리 가져올 파일을 결정하는 방법이 있다. Coda file system[2]에서는 이 두가지 방법을 혼용해서 사용하며, 이러한 파일의 관계를 tree 로 표현하여 결정하는 방법[3]도 있다. 또한 비연결 상태에서 수행할 때 모빌 클라이언트내에서 비연결 상태 동안에 일어난 일들을 기록해 놓았다가 재연결 되었을 때 reintegreation 을 하면서 용하

게되는 log 파일을 최적화시키는 방법[6]을 제안하기도 했다. 재연결이 되면 연결이 끊어진 동안 모빌 클라이언트에서 수행했던 것들을 기록한 log 파일을 기반으로 해서 타 모빌 호스트들과의 의존성을 검사하며 reintegration 을 수행하며, 다른 관련된 모빌 호스트와 파일과 관련하여 충돌이 발생하면 ASRs(Application-specific resolvers)를 각 파일마다 이용해서 해결을 하며[7] 해결이 안되는 경우에는 사용자에게 수동으로 문제를 해결할 수 이는 도구를 제공하기도 한다.

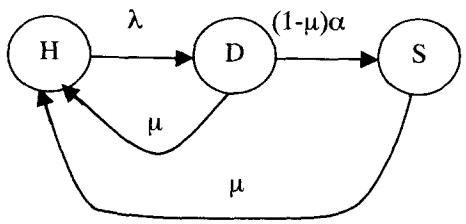
Database

모빌 환경에서 database 서비스를 제공하기 위해서는 file system 과 마찬가지로 어떤 데이터를 hoarding 하는가는 중요한 문제이며, 이를 위해서 (a)사용자에게 자주 사용하는 쿼리를 등록해서 이용하는 방법과 (b)사용자의 과거 사용했던 것을 유지하고 추적해서 hoarding 해 올 쿼리와 객체들을 결정하는 방법이 있다[5]. 또한 데이터 베이스 안에 모빌 클라이언트의 접근경향을 체크할수 있는 새로운 튜플(*hoard key*)을 추가하여 이 키를 가지고 hoarding 시에 전체 테이블을 hoarding 하는것이 아니라 hoard key 를 가지고 관련된것들만으로 새로운 데이터 베이스 조각을 만들어서 그것만을 hoarding 하는 방법도 있다[8].

Web-browsing

약한 연결성과 느린 무선 연결 환경을 고려한 file system 과 database 와 마찬가지로 Web-browsing 에서도 많은 연구가 진행 되었다. 여기서도 캐쉥 기법을 이용한다. 그래픽이 더해진 웹 문서의 경우에는 Distillation 과 Refinement 라는 기술이 사용되어 진다[9,10,11].

3. 비용 분석



<그림 2> Marcov 모델

위 그림은 모빌 환경에서 모빌 클라이언트의 수행 상태를 Marcov chain 을 이용하여 표시한 것이다. H 는 Hoarding 상태, D 는 Disconnected 상태이며 S 는 Stop 상태를 나타낸다. 이때 λ는 연결이 끊어질 비율(rate)을 나타내며 μ는 끊어진 연결이 다시 연결될 비율을 나타낸다. 그리고 α는 끊어진 상태에서 수행을 지속하다가 재연결되지 않고 정지 상태로 갈 비율을 나타낸다. 이때 α는 H 상태에서 얼마만큼의 Hoarding 을 했는지에 따라 변하는 값이 된다. 모든 에러 또는 재연결의 발생은 Poisson process 라 가정한다.

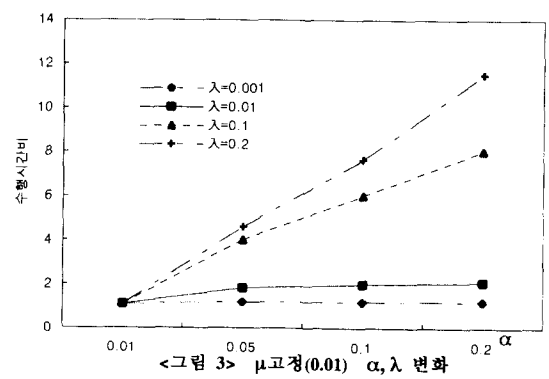
모빌 환경에서 위와 같은 현상이 반복된다고 할 때 우리는 위와 같은 주변환경에서 받은 영향의 정도를 분석하여 끊김 에러율(λ)과 재연결율(μ) 그리고 끊어진 상태에서 정지 상태로 갈 비율(α)을 가지고 시뮬레이션을 통해서 이러한 환경 요소들이 실제 모빌 컴퓨팅 타스크 평균 수행 시간에 얼마나 영향을 주는지 분석해 보았다.

4. 성능 평가

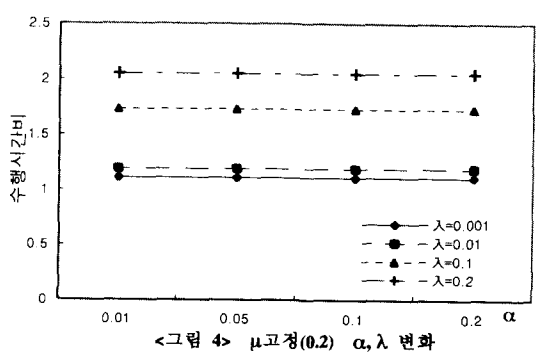
4.1 시뮬레이션 환경

입력 변수 λ, μ, α를 가지고 실험을 했으며 hoarding rate 는 값을 1.1 로 고정하여 실험하였다. 물론 이 값도 변수가 될 수도 있지만 hoarding rate 와 α값 사이의 적절한 관계를 연구한 후 포함시킬 예정이다. 재연결시에 필요로 하는 reintegration overhead 는 고정값으로 처리하였다. 전체적인 수행은 에러가 없는 경우의 프로세싱 시간을 미리주고 이 프로세싱 처리를 위에서 설명한 모빌 환경에 적용하였을 때 에러를 고려한 프로세싱 처리 시간 대 에러가 없을 때 프로세싱 처리시간의 비율을 시뮬레이션을 통하여 측정하였다. 프로세싱 시간은 100,000 으로 하고, 각 λ, μ, α중의 일부 파라메터를 변화시키며 수행시간을 측정하였다. 또한 랜덤 변수에 의한 정확도를 높이기 위해서 10000 번의 수행을 한 후 평균값으로 성능을 계산하였다.

4.2 시뮬레이션 결과

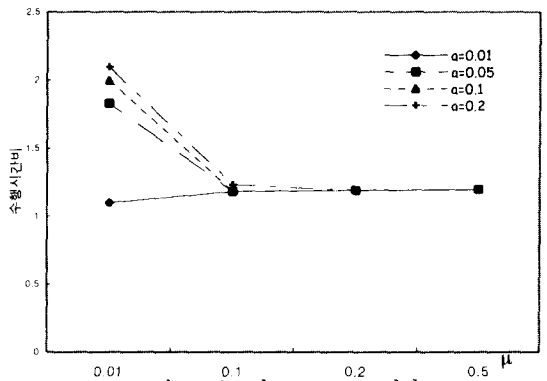


<그림 3> μ고정(0.01) α, λ 변화



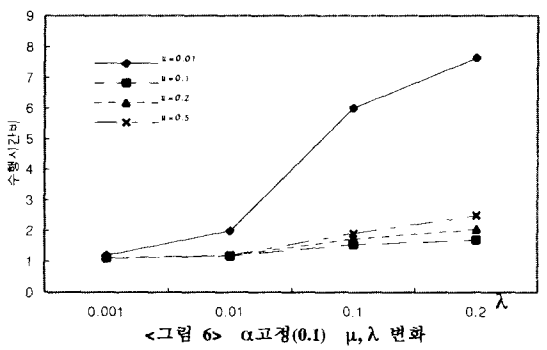
<그림 4> μ고정(0.2) α, λ 변화

<그림 3> 에서는 μ값이 0.01 로 고정하고 λ, α값을 변화시키면서 시뮬레이션을 수행한 결과를 나타내었다. 이 그래프에서는 재연결율이 낮은 환경에서 λ, α값이 변화할 때 수행시간 비율의 변화를 보여준다. α값이 작을때는 수행시간 비율에 큰 차이가 없지만 α값이 커지게 되면, λ값이 클수록 수행시간 비율이 커지게 되는 결과를 나타낸다. <그림 4>에서는 재연결 비율이 큰값(μ값이 0.2)일 때 넣어서 시뮬레이션을 수행한 결과이다. 결과는 α값에 거의 영향을 받지 않고 일정한 비율을 나타내는데 이유는 μ값이 일정한 값 이상이 되면 stop 상태로 전이하기 이(또는 stop 으로 전이되더라도 곧)전에 재연결이 되어 버리므로 α값에 영향을 안받는 것이다.



<그림 5> λ고정(0.01) μ, α 변화

<그림 5>에서는 λ값을 고정(0.01)하고 μ, α 값을 변화하였는데, 여기서는 μ값이 증가함에따라 α값의 영향을 덜 받는다는 것을 알 수 있다. 이유는 stop 상태로 전이 되어도 곧바로 재연결이 되기때문이다.



<그림 6> α고정(0.1) μ, λ 변화

<그림 6>은 작업 시간을 충분히 크게 고정(100,000)하고 stop 의 전이 비율 α를 0.1 로 고정한후 λ, μ값을 변화 시켜 보았다. 그림에서 보듯이 각 μ값은 λ값이 증가함에 따라 수행시간비는 증가한다. 이때 λ가 높은 값(0.2)이고 μ가 0.1 일때 수행시간 비가 가장 낮고, 연결율이 더 좋을 때(μ값이 0.5) 오히려 수행시간비가 증가 한 것을 알 수 있다. 이는 앞에서 설명한 모빌 환경에서의 특징으로 <그림 2>에서 H 와 D 상태 변화를 자주 반복하여 reintegreation 비용이 과다하게 되기 때문이다.

5. 결론 및 향후 과제

모빌 환경에서는 주변 환경에서 영향을 받는 요인들이 많다. 본 논문에서는 이러한 환경요인들이 모빌 컴퓨팅 성능에 어떠한 영향을 미치는지 시뮬레이션을 통해 확인하였다. 이러한 실험 결과는 효율적인 모빌 컴퓨팅 알고리즘을 선택하는데 도움이 될 수 있다. 향후 연구로서 보다 다양하고 현실적인 시스템 변수를 포함한 시뮬레이션과 수학적 분석을 포함할 예정이다.

참고 문헌

[1] E. Pitoura and G. Samaras, "Data Management for Mobile Computing," *kluwer Academic Publishers*, : 37-48, 1997.

[2] J. J. Kistler and M. Satyanarayanan, "Disconnected Operationg in the CodaFile System," *ACM Transactions on Computer Systems*, 10(1):213-225, February 1992.

[3] C. Tait, H. Lei, S. Acharya, and H. Chang, "Intellgent File Hoardingfor Mobile Computers," In *Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (Mobicom '95)*, Berkeley, Ca, October 1995.

[4] G. H. Kuenning, "The Design of the Seer Predictive Caching System," In *Processing of the IEEE Workshop on Mobile Computing system and Applications*, Santa Cruz, CA, December 1994.

[5] R. Gruber, F. Kaashoek, N. Liskov, and L. Shrira, "Disconnected Operation in the Thor Object-Oriented Database System," In *Processdings of the IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA. December 1994.

[6] L. Huston and P. Honeyman, "Disconnected Operation for AFS," In *Proceedings USENIX Syposium on Mobile and Location-Independent Computing*, pp. 1-10, Cambrige, Messachusetts, August 1993.

[7] P. Kumar and M. Satyanarayanan, "Flexible and Safe Resolution of File Conflicts," In Proceedings of the USENIX Winter 1995 Conference, pp. 246-285. Spiner-Verlag, LCS 105, 1981.

[8] I. F. Akyildiz and J. S. M. Ho, "Dynamic Mobile User Location Update for Wireless PCS Networks," *ACM/Baltzer Wireless Networks Joural*, 1(2), 1995.

[9] V. N. Radmanabhan and J. C. Mogulm, "Improving HTTP Latency," *Computer Networks and SDN Syatems*, 28(1), December 1995.

[10] A. Fox and E. A. Brewer, "Reducing WWW Latencyand Bandwidth Requirments by Real-TimeDistillation," In *Proceedings of the 5th International World Wide Web Confernece*, Paris, France, May 1996.

[11] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir, "Adapting to Network and Client Variability via On-Demand Dynamic Distillation," In *Processdings of the ASPLOS-VII*, Cambrige, MA, October 1996.