Introduction to Slurm

2024.06.24 전웅배

Slurm?

- Slurm : Simple Linux Utility for Resource Management
- Used in most of world’s most powerful High-Performance Computing(HPC) clusters.
- Lawrence Livermore 연구소에서 만듦. (노벨상 16개)

How many Nobel Prizes are there in LBNL?

16 times

Scientists at Berkeley Lab have earned the highest honor for scientific achievement, the Nobel Prize, 16 times.

Resource Management

극단적 Static Policy:

특정 사람에게 배정된 자원이 절대 변하지 않음.

e.g. X : Node 1,2 / Y : Node 3,4 / …

극단적 Dynamic Policy:

특정 사람에게만 배정되는 자원이 하나도 없음.

i.e. 모든 자원이 on-demand

Resource Management

- 이상적인 Dynamic Policy : Maximum efficiency 를 지향함.
- 이상적인 Static Policy : Minimum maintenance 를 지향함.
- 모든 Resource Management는 둘 사이의 tradeoff.

Previous Episode in MIIL…

- Dynamic Policy. 특정 개인에게 제한된 것이 아무것도 없었음. '이론적으로는' optimal.
- 현실 : Dynamic 과 static의 단점만 절묘하게 더해진 상태.
- Suboptimal efficiency :
  - 사용되는 자원과 실제 사용하고 있는 자원의 mismatch 심했음.
  - 불필요한 리소스 남용, 특정 시기에 인물의 과점 효과 심했음.

    (X가 항상 쓰니까 못씀 – 안씀 – X만 씀 - …) + Deadlock (B는 A가 쓰고있어서, A는 B가 쓰고있어서)
- 극한 난이도의 관리:
  - 관리하기는 더욱 힘듦. 모두가 관리자 권한을 난사하며 시스템 설정을 변경해서 씀.

    e.g. SSH timeout 설정 – 하나씩 해제 – 왜 서버x만 timeout 걸리지?
  - 관리자 권한은 모두가 사용하지만, 책임은? 흠…

왜 이제서야? 지금까지 뭐 했냐?

- 필요성 자체는 22년 11월에 파악. (지금은 Alumni 인 ??? 군의 서버4 디스크 0B 사건)
- 23년 1월 ~ 24년 1월 까지 4차례 실패. (Slurm과 같은 관리 시스템 도입은 사실 작년 신년 목표였음.)
- 2월 Slurm minimum working example 재현 성공.
- 2월~5월 동안 베타 테스트, 이제 완전 통합을 앞두고 있음.

- ??? : Slurm 말고 다른 거 썼으면 금방 해결 했던 것 아닌가요?

  A : Docker 기반 Kubernetes가 있는데, static policy 전용이라서 제외.

  (구글 GCP, 네이버 클라우드 등 종량제 서비스를 지원하는 용도.)

Slurm은 허가제가 아니라 신고제

- Slurm은 사용자의 리소스 신청을 '허가' 하는 시스템이 아닙니다!
- 사용자들의 요청을 '접수' 하는 시스템이고,
  접수된 요청 사항이 현실 제한과 충돌하지 않는지 확인하는 시스템.
- 접수된 요청 사항이 조건을 만족한다면 무조건 배정되는 시스템.
- 하지만 이는 사용하고 싶은 자원을 전부 사용할 수 있도록 보장하는 시스템은 아님.

- 유사 예시 :
  마트 계산대. 내가 사고싶은 물건을 돈 있으면 마음껏 살 수 있지만 계산대 줄은 서야함.
  웬만한 한도 내에서는 사고싶은 만큼 살 수 있지만, 적당히 해야함.

신고 해야하는 자원의 종류

- GPU 뿐만 아니라, 서버에서 실행하는 프로그램의 모든 요소는 자원을 소모함.
- GPU, CPU, MEM(RAM) + 사용 시간
- Slurm이 상황을 통제해서 최적의 자원을 '기본값' 으로 배당하는게 아님.
  내가 필요한 자원을 '파악해서' Slurm에게 신고하는게 기본.

쓰는 방법은 가이드 참조

- wbjeon2k.github.io/miil 링크 참조.
- 해당 가이드에 적혀 있는 내용을 질문하면 응답하지 않을것. (관리자는 ChatGPT가 아님)
  베타 서비스 도중에는 FAQ / 편의성 개선 파악 등 정보 수집을 위해서 응답 했지만,
  가이드가 많이 업그레이드 된 현재 시점에서는 가이드 내용으로 충분하다 판단됨.

- 대신 contribution 제공시 소정의 선물을 드릴 예정입니다! 🎁 🎁 🎁
  Tips & Tricks : 어떤 기능은 ~ 하면 잘 쓸 수 있음
  Errata : 가이드에는 X로 나와있는데 실제로는 Y가 맞음
  Update : X 관련 내용이 없어서 Y 자료를 만들어 왔으니 추가해주세요 (Ref : A,B,C)

기타 내용 및 사용자 설문 응답

- 현재는 접근 device 제한 목록에 MEM(RAM) 이 빠져있음.
  i.e. 현재는 사용하겠다고 신고 한 양 보다 RAM을 더 많이 쓴다고 작업이 종료되지 않음.
  OOM으로 터지는 현상 또 생기면 그 때는 도입 해야함. (제발! 생기지 않기를 빕니다.)
- X 기능을 추가 해주세요 / 개선 해주세요 :
  필수적인 기능이 아니면 웬만해선 기능 추가를 할 예정이 없습니다.
  추가 가능 예시 : Torch DDP가 안돼요. 고쳐주세요
  추가 불가 예시 : 방법이 없는건 아닌데 불편합니다
- Slurm등 시스템 프로그램들은 상상 이상으로 설정 변경을 수동으로 해야하는 경우가 많습니다 ㅠ

기타 내용 및 사용자 설문 응답

- 노드 당 사용 가능한 GPU를 늘려주세요 :
  반려입니다. 여기서 더 늘리면 한 사람이 서버 전체를 점거하는 상황이 발생함.
  차선책 : 5,6 통합 후 1인당 최대 사용 GPU 개수 증가, MIIL HPC에 맞는 multi-node training 방법 찾는중
  (GPU 기종 달라도 Torch DDP 가능한 것 확인)

- 연구실 전체 합의 하에 특수 상황에 1인당 최대 사용 GPU 개수 늘리는 제도 :
  필요성에는 동의함. (특정 컨퍼런스 제출하는 사람이 적은데, GPU 사용량은 낮은 경우)
  형평성 문제 때문에 연구실 내부 갈등 요인이 될 가능성 매우 높음. 조심스러움.

앞으로의 전망?

- 서버 5,6 하드 포맷 + Slurm 통합

  6/25(내일) 공지 했던대로 HW 업그레이드 + 전원 끈 김에 통합까지 진행 예정입니다.

- Multi-node Multi-GPU training 방법 찾기 :

  현재는 node=1 로 사실상 고정. 모든 노드에서 공통적으로 사용할 수 있는 디렉토리가 없음.

  NFS 추가를 통해서 모든 노드에서 공통적으로 접근 가능한 경로를 늘릴 예정.

  GPU 모델이 달라도 Torch DDP 가능 (똑같아야 가능했던 건 옛날)

Appendix

- 다음 내용부터는 Appendix. 심심하면 읽어보세요.

Design Principle

- Dynamic : Static 하면 효율성이 박살남. e.g. GPU 8개 필요한 실험이 생기면 GPU 8개가 있는 노드를 새로 만들고, 필요한 때가 올 때 까지 남겨둬야함.
  만약 해당 노드를 유동적으로 만들 수 있다면 그건 더 이상 static이 아닌데…?
  따라서 dynamic policy 를 사용할 수 있는 Slurm 밖에 답이 없음.
- Easy to use : 쓰기 어려우면 애써 만들고도 사서 욕먹는 상황 발생. 근데 slurm 진입 장벽이 있는건 부정할 수 없는 사실. Slurm-jupyter , 가이드 작성 등으로 보완.
- Fairness : 관리자 마음대로 좌지우지하는 상황은 생기면 안됨. 개입은 최소화. 문제가 생기면 해결 해야하지만 그 외에는 개입 x
- SSH : 서버에 ssh 접속이 가능해야함. 이거 하나만 포기하면 gpu backdoor 등 거의 모든 문제가 해결되지만 연구실 여론 폭발 예정

Background Stories

- 서버 0B 사건 : 디스크 용량이 문자 그대로 0B 남음. Bash 에서 자동완성 탭을 누르면 임시파일 생성이 안돼서 자동완성이 안됨. 새로운 파일 수정 불가능.
- 타 연구실 서버 도용 사건 : 서버랙 위치 등 서버 관리 자료가 없는 채로 초대 랩장 졸업. 서버랙 2번 위치를 착각하게 됨 (표지가 없었음)
  착각한 서버랙의 문이 우리 연구실 열쇠로 열림 (알고보니 제조사가 같으면 그냥 전부 호환됨. 전체 서버실 중 90% 열기 가능)
  GPU가 없던 새 서버에 GPU 설치를 하려는 상황에 해당 서버는 하필 CPU 전용 서버였고… 우리것이라 확신함. 다른 연구실 서버를 탈취하고 우리 GPU를 꽂아서 사용.
  몇 개월 뒤에서야 잘못된 것을 파악한 상대 연구실 (L교수 연구실) 학생의 전화를 받고서 어마어마한 일이 벌어졌음을 알게 됨…
- SSH 사건 : 원래는 보안상 ssh 최대 연결시간이 제한됨. 매 년 보안감사때 제한함. 그러다가 언젠가 서버 x번만 ssh 연결이 주기적으로 끊긴다는 문의 접수.
  원래는 전부 제한이 되어 있었지만 사람들이 임의로 하나씩 해제 하다보니 제한 된 서버x만 예외가 된 것 처럼 주객전도가 되었던 상태…
- 서버실 온도 제한 사건 : 기존 서버실은 딥러닝 인기 폭발하기 전에 지어진 서버실. 그러나 딥러닝 붐으로 서버실이 일종의 난개발 상태를 겪음.
  기존 서버실 수용량 감안 없이 많은 연구실들이 서버를 확충하며 한계점을 아득히 넘어섬. (계산 결과 냉각기 용량의 2배 이상의 열 발생 중, 서버실 전체 전력 한계치 근접)
  5월부터 서버실 온도가 40도씩 올라가기 시작한 결과… 관리자 톡방이 생기고 오창주 선생님의 온도 경보때 마다 사용중인 서버를 일정 수량 이상 꺼야했음.
  물론 약속 지키는 사람 따로 안 지키는 사람 따로 있어서 끄는 연구실만 맨날 끔. 대형 공사를 몇 달 동안 한 결과 이제는 옛날 얘기가 되어버림.
- OOM 사건 : 서버 사용량이 폭주해서 시스템 RAM OOM 발생. ssh 접속이 안돼서 물리적으로 서버를 끄고 다시 켜야했음.