HRVATSKI OGRANAK MEĐUNARODNE ELEKTRODISTRIBUCIJSKE KONFERENCIJE - HO CIRED
9. (15.) savjetovanje
Šibenik, 25. - 27. svibnja 2025.

SO3 - 6

Branimir Gabrić
HEP ODS
branimir.gabric1@hep.hr

Kristijan Frano Ćavar
HEP ODS
kristijanFrano.cavar@hep.hr

Ivan Periša
HEP ODS
ivan.perisa@hep.hr

KORIŠTENJE MODERNIJIH RAČUNALNIH TEHNOLOGIJA PRI MJERINFO PLATFORMI

SAŽETAK

U radu će se primjerima pokazati primjena modernijih računalnih tehnologija i to u tri poglavlja. U prvom poglavlju će se pokazati kako generativna umjetna inteligencija može primijeniti za bržu izradu web aplikacija i izvještaja pri MJERinfo platformi. U drugom poglavlju će biti prikazana uporaba CUDA platforme kojom se neki izračuni i izvještaji pokušavaju brže izvršavati koristeći procesor grafičke kartice. U trećem poglavlju će se pokušati pronaći korist u uporabi velikih jezičnih modela koji su odspojeni od interneta produkta za ubrzanje razvoja MJERinfo aplikacija te brže snalaženje unutar dokumentacije.

Ključne riječi: mjerinfo, cuda, generativna umjetna inteligencija, python

PREPARATION OF PAPER

SUMMARY

In this paper, examples will demonstrate the application of modern computing technologies in three chapters. The first chapter will show how generative artificial intelligence can be applied for faster development of web applications and reports on the MJERinfo platform. The second chapter will present the use of the CUDA platform to attempt faster execution of some calculations and reports using the graphics card processor. The third chapter will attempt to find the benefit of using large language models that are disconnected from the internet to accelerate the development of MJERinfo applications and facilitate quicker navigation within the documentation.

Key words: mjerinfo, cuda, generative artificial intelligence, python

1. UVOD

Ovim radom će se pokušati opisati iskustvo korištenja modernih informacijskih tehnologija i to primijenjene na izradu i održavanje MJERinfo platforme. Rad će biti razdijeljen u tri potpoglavlja. U prvom će se napraviti osvrt na primjenu sustava generativne umjetne inteligencije u svakodnevnom poslu. Riječ je o sustavima poput ChatGPT, Gemini, Copilot i sl. U drugom potpoglavlju će se pokušati prenijeti iskustvo korištenja procesorske moći grafičke kartice u ubrzavanju obrade podataka. Treće potpoglavlje će govoriti o izradi lokalnog LLM modela koji je odspojen od interneta, koristeći vlastite hardverske resurse te kojem je ideja omogućiti razgovor korisnika s pohranjenim internim dokumentima MJERinfo sustava.
MJERinfo platforma je sustav za pohranu mjerenja iz različitih izvora, a za potrebe Sektora za vođenje HEP ODS-a. MJERinfo je detaljnije opisan na prethodnim CIRED skupovima, prvi rad koji tematizira ovu platformu je iz 2018. godine. [1]

2. TEKSTUALNI AI ALATI

2.1. Uvod

Možda je već i anakrono diviti se mogućnostima generativne umjetne inteligencije na prijelazu u 2025. godinu, ali autori ovih redaka su značajnije počeli koristiti ove računalne blagodati tek nedavno. Šale radi, tek kad se pojavio famozni *ChatGPT*, tražili smo sustav da nam napiše sonet na temu SCADA inženjeringa, i, usuđujemo se reći, nije bilo loše napisano. Pjesmu smo zataknuli na uredsku ploču i tamo visi od prosinca 2022.

Dakle, tek nedavno, kada nam je firma otvorila račun za jedan ovakav servis, koristimo ove alate u svrhu poboljšanja i ubrzanja rada MJERinfo sustava. Korporativnim računom riješena je problematika sigurnosti i povjerljivosti podataka koji se šalju na vanjske poslužitelje, nadalje, plaćeni servisi su bitno moćniji od besplatnih te je korištenje istih elegantnije i manje zamorno. U idućim potpoglavljima su neki primjeri korištenja spomenute tehnologije.

2.2. Provjera valjanosti podataka

Početni zadatak je bio izraditi web servis koristeći *.NET core* tehnologiju koji je trebao prozivati veliku količinu mjerenja iz MJERinfo sustava. Servis rezultat vraća u JSON formatu. Međutim javio se problem validnosti tako poslanih podataka, početna povratna informacija od subjekta koji je koristio servis je bila da ti podaci nisu valjani i da ih se ne može iskoristiti na dogovoreni način.

Premišljali smo se hoćemo li analizirati nepreglednu i nezgrapnu količinu podataka u JSON formatu i to nas je odbijalo jer bi oduzelo previše vremena uz postojeće obveze. Rješenje problema je bilo u prozoru do: zašto ne iskoristiti umjetnu inteligenciju da ona to načini mjesto nas: u *prompt* je uneseno par redaka JSON formata, ovaj je odmah prepoznao o kakvom tipu se podataka radi i o čemu je otprilike riječ, u nastavku razgovora je *chat*-u rečeno da napravi sučelje za učitavanje ovakovog tipa podataka i da ih pohrani u *pandas dataframe*. Nadalje, naređeno mu je da izradi *flask web* aplikaciju koja će vizualizirati dohvaćene podatke. U nastavku je *chat* predloženo da aplikacija treba imati sučelje za unos vremenskog raspona te naziva konkretnog mjerenja. U konačnici je u *promptu* opisano kakvu agregaciju rezultata želimo (grupiranje po jednom tipu, pa drugom, pa zbroji, pa pobroji, pa usrednji…) Drugim riječima, unutar 45 minuta je izrađen vizualni validator podataka te podskup analitičkih potprograma.

2.3. Web aplikacija

Izrada baze i web sučelja za procesnu aplikaciju za održavanje popisa procesnih uređaja s IP adresom. Koristeći *prompt* izrađene su upute koji su omogućile izradu *postgre* SQL baze te izradu web aplikacije koja služi za pohranu informacija o IP uređajima te za evidentiranja dostupnosti uređaja na mreži. Aplikacija vjerojatno nikada ne bi bila započeta da se nije uvidjelo kako se početni kostur može vrlo lako da izgraditi koristeći generativnu umjetnu inteligenciju. Pošto će se aplikacija nalaziti unutar

procesne mreže, promptu je naređeno i da predloži na koji način se čitav projekt treba pohraniti u *dockeru*, a kako bi bio neovisan o vanjskim uvjetima i kako bi ga se na jednostavan način moglo ugraditi unutar procesne mreže (koja je odvojena od interneta). Web aplikacija je dio većeg projekta unutar procesne mreže koji vizualizira poveznice IP procesnih uređaja te obavještava u slučaju prekida komunikacije. Cjelokupni projekt će biti opisan, ako ne na ovogodišnjem, onda na idućem domaćem CIRED skupu.

2.4. Skrivena greška

Ovaj primjer govori kako je kopiranjem problematičnog odsječka koda u *prompt*, chat pronašao rješenje skrivene programske greške. Riječ je bilo o *iec104* klijentu koji koristi asinkrone metode. Uglavnom, nakon što klijent danima radi uredno, uruši se iz nekog nepoznatog razloga. Nakon prijedloga *chata* što je potrebno popraviti u kodu, klijent se više nije rušio, stabilno je radio tjednima.

U sva tri slučaja, da nije bilo chata, s naše strane bi se odustalo. Niti bi se izradio vizualni validator podataka te dodatne agregacije potrebne za analizu velike količine podataka, niti bi se izrađivala web aplikacija za održavanje IP opreme (ostalo bi u tekstualnim datotekama), a od *iec104* klijenta bi se odustalo jer je nepouzdan.

2.5. Još nekoliko primjera

Još neki primjeri koji idu u prilog učenja te usavršavanja postojećih vještina u *Pythonu* (*pandas*, *Django*, *Flask*): naručena je izrada programskog koda koji će predviđati vrijednost mjerenja koja su pohranjena u MJERinfo platformi s obzirom na meteo prilike. Predložen je program koji koristi linearnu regresiju koristeći biblioteku *scikit*, greška predviđanja je bila manja od 5% . Također, od modela je zatražena izrada klasteriranja 35.000 SN trafostanica s obzirom na GEO lokaciju (iz GIS-a): programski kod je načinio 21 skupa koji se iznimno dobro preklapaju s postojećim granicama distribucijskih područja. Dodatno, načinio je i koristan program koji s obzirom na koordinate SN stanica u nekom DP-u računa težište, dakle točku iz koje je teoretski najpraktičnije obići sve ostale točke. Softver može biti jako koristan i kada se rabi biblioteka *pandapower*. [2]

2.6. Problemi

Zašto to tako dobro funkcionira? Sustav je treniran na otvorenom znanju pohranjenom na forumima, repozitorijima softvera i sl. Međutim, kada ga idemo pitati nešto specifično za SCADA sustav kojeg koristimo zadnjih 20 godina i koji ima zatvoren sustav dokumentacije, *chat* (bar ovaj na kojeg imamo pretplatu te u trenutku pisanja) “nema pojma”, štoviše nikakvoga suvisloga savjeta ne može dati. Slično je i s GIS produktom kojeg koristimo u firmi.

3. CUDA

Nevjerojatan rast dionica firme *nVidije*, koja je donedavno bila najbogatija firma na svijetu (*DeepSeek*), ne zahvaljuje sigurno samo izradom gejmerskih grafičkih kartica. Rano su već, negdje tamo 2007. godine shvatili kako se grafičke kartice ne moraju koristiti samo za obradu grafike već i za poslovne i znanstvene proračune (početak CUDA-e).

Negdje u proljeće 2024. nam je pod ruku došla grafička kartica *nVidia p620* (arhitektura iz 2017. god.) koja ima 512 CUDA jezgri; CUDA je platforma za paralelno procesiranje te izradu aplikacija. Grafička kartica koju smo testirali ima moć računanja 5.1; najveći mogući u trenutku pisanja ovog teksta je 9.0. Za test je odabran CSV dokument veličine 370 MB koji je učitan koristeći *python pandas* te na odgovarajući način obrađen koristeći CPU. Takva obrada je trajala 21 sekundu. Međutim, kada je isti programski kod pokrenut koristeći pred-direktivu za korištenje GPU-a (mjesto CPU-a), isti taj zadatak izvršen je za 7 sekundi.

Za napomenuti je da testna grafička kartica nema puno radne memorije (tek 2 GB, primjena joj je više uredska, za pogonjenje više monitora) te nije mogla učitati veće datoteke. Također je važno napomenuti kako ovdje nije korišten *cuDF*, koji je posebno optimiziran za rad na grafičkoj kartici, ali traži izvjesne izmjene u postojećem kodu. Vjerujemo da bi tada izvršenje koda trajalo još kraće (barem kada se promatra promotivni materijal - u specifičnim primjenama *cuDF* nudi ubrzanja i do 150 puta).

Grafička kartica koja ima moć računanja 8.1 i 12 GB rama košta do 500 eura i ima preko 4000 CUDA jezgri; to nije visoka cijena, ukoliko će značajno skratiti trajanje neke zahtjevnije obrade podataka. [3]

4. CHAT BEZ INTERNETA

Učinilo nam se interesantnim jesenas pokušati izraditi vlastiti chat program koji će biti neovisan o internetu. Višestruka je korist posjedovanja takove „inteligencije na vlastitoj infrastkuturi“, primjerice: nema straha od gubitka privatnosti, analitika nad povjerljivim podacima ostaje unutar kuće. Podrška na internetu je jako dobra. Dovoljno je posjetiti stranicu https://huggingface.co/. Tamo se da pronaći nebrojena količina različitih LLM-ova (Large Language Model) koji se mogu besplatno preuzeti i koristiti. Izazov je samo odabrati model koji može optimalno riješiti zadani problem. Također, zakučasto je pronaći model koji dobro govori hrvatski jezik.

Problematika se testirala na dva načina: koristeći gotovi produkt otvorenog koda - LLM studio (može se pokušati kod kuće) te koristeći *python*, vlastiti kod (naravno, preporučeni skelet koda je preuzet s interneta). Računalo koje je bilo na raspolaganju za testiranje je radna stanica iz 2014. godine, tada vrhunac tehnologije. Tom uređaju je bilo potrebno u prosjeku nekoliko minuta odgovoriti na ponuđeno pitanje, dok primjerice na isto takovo pitanje na privatnom, kućnom, računalu (s grafičkom karticom *nvidia rtx4060 8GB*) sustav odgovara u stvarnom vremenu. Dakle, kao i u prethodnom poglavlju, kod korištenja LLM modela je od iznimne važnosti korištenje moderne grafičke kartice. Također, ukoliko grafička kartica ima više RAM-a, veći model stane u memoriju i razgovor protječe bez zatezanja.

Ono što se dodatno testiralo je kako jedan takav sustav može shvatiti dokumentaciju koja je izrađena unutar firme i može li biti od ikakve pomoći. Ova rabota je još zahtjevnija, primjerice, testnom računalu je potrebno oko 8 minuta za odgovoriti na pitanje (8 jezgri na 100%). Riječ je o RAG postupku gdje se ulazni, privatni dokumenti, nad kojim model nije testiran, pokušava obraditi na način da bude shvatljiv tom modelu. Korišten je rudimentarni RAG pristup gdje je pdf pretvoren u tekst i razdijeljen u mnogo manjih cjelina. Nije korišten RAG princip gdje se od pojmova u ulaznim dokumentima gradi graf (povezanost pojmova).

Načinili smo više testova. Prvo je injektiran tekst knjige *Wizard of Oz* (s projekta Guttenberg.) Program smo pitali može li sažeti što se dogodilo na početku knjige. Odgovor je bio zadovoljavajući. Potom je pitano može li model pronaći neke likove u knjizi i ispisati ih u obliku *python* polja:

[Ozma, Princess, Wizard, Mombi, Glinda, Gwig, Sorcerer, Prince, Zeb]

Drugi test je bio tekst Staroga i Novog zavjeta s web stranica Kršćanske sadašnjosti . Pitali smo ga tko je bio Isus Krist, a ovaj je odgovorio na solidnom hrvatskom na način da bi se takvim odgovorom mogao pohvaliti prosječan krizmanik. Kad smo ga pokušali prevariti i pitati tko je Dorothy, ovaj je rekao: *Ovaj tekst nije povezan s ličnošću Dorothy, nego se radi o dva različna teksta iz Starog zavjeta i Novog zavjeta. Dorothy nije spomenuta u ovim tekstovima.*

Konačno, u program su učitani različiti bilteni naše firme. Pitali smo program raznovrsna administrativna pitanja poput potpore za rođenje djeteta, kakva prava imaju radnici iznad 60 godina, te pitanja u svezi otkaznog roka. Chat je uredno odgovarao npr. : *Radnik stariji od 60 godina ima pravo na najmanje plaću na temelju koeficijenta radnog mjesta za koje je imao sklopljen ugovor o radu u mjesecu...* Za pomoć djetetu je dogovorio na engleskom jeziku, ali je naveo točnu novčanu svotu potpore. Također prava za otkazni rok je naveo u točkama s obzirom koliko je zaposlenik bio dugo zaposlen. Odgovori nisu savršeno nijansirani, ali su pristojni, s obzirom na kratkoću vremena koju smo proveli na odabir modela.

Korišteni modeli su bili *Zephyr-7b-beta.q4_0* te *Solar-10.7b-slerp.q6_k. Embedding* modeli su pritom bili: *bge-small-en-v1.5* te *bge-multilingual-gemma2 (mogu se pronaći na stranici* https://huggingface.co/) . Koristila se biblioteka *llama.cpp* koja omogućuje razvoj i korištenje glomaznih jezičnih modela na običnim računalima.

Uporabom privatnog *chatbota* unutar MJERinfo platforme omogućuje se napredniji pristup korištenju platforme. Naime, MJERinfo oduvijek ima eksponiranu *Jupyter* radnu bilježnicu gdje svaki zaposlenik ima mogućnost razvijati svoj vlastiti kod koristeći resurse MJERinfo platforme. Na ovaj način korisnik može dobivati upute i izgenerirani programski kod koji je specifičan samo za MJERinfo.

Također, moguće je priložiti privatne i strateške dokumente unutar procesnog „svijeta“ te ih propitivati, sažimati, istraživati bez straha od gubitka privatnosti. [4], [5]

5. ZAKLJUČAK

Pokušali smo pokazati kako moderne računalne tehnologije koje bitno utječu i na našu svakodnevicu, mogu biti od pomoći u firmi koja se bavi elektrodistribucijom. Konkretno, primjeri koji se spominju u ovom radu zapravo svjedoče da se zasigurno ne bi ni pokušalo pristupiti rješavanju pojedinih zadataka jer problem, ili nadrasta znanje korisnika, ili bi njegovo rješavanje predugo trajalo. Bilo da je riječ o izradi aplikacije, web servisa, ispravci postojeće aplikacije, bilo da je riječ o znatnom ubrzanju postojećeg logičkog izvršavanja procesa ili pokušaju preslikavanja privatnih chat modula u okružje poslovno-procesne mreže, moderne informatičke tehnologije svakako nije za odbaciti već maksimalno pokušati iskoristiti njihov potencijal kako bi se unaprijedilo IT i OT poslovanje i ujedno olakšao svakodnevni rad. Ovaj rad nije napisala umjetna inteligencija.

1. LITERATURA

[1] K .F. Ćavar, B. Gabrić, I. Periša, "MJERinfo – time series platform“, 4th International Conference on Smart Grid Metrology, Cavtat, Croatia, April 2023.

[2] https://scikit-learn.org/stable/

[3] https://developer.nvidia.com/cuda-zone

[4] https://lmstudio.ai/

[5] https://huggingface.co/