



ANALIZA PRIMENE AMI SISTEMA U CLOUD OKRUŽENJU

APPLICATION ANALYSIS OF AMI SYSTEM IN CLOUD ENVIRONMENT

Bratislav Batinić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – U ovom radu opisana je AMI arhitektura i predstavljen je problem rukovanja velikom količinom podataka. Nakon toga, prezentovano je softversko rešenje AMI sistema, gde se deo aplikacija nalazi na Cloud-u. Na kraju rada prikazani su i analizirani rezultati testiranja izabranog skupa funkcionalnosti implementiranog rešenja.

Ključne reči: AMI sistem, Cloud

Abstract – This paper describes the AMI architecture and presents the problem of handling a large amount of data. After that, the software solution of the AMI system is presented, where a part of the applications is on the Cloud. Also, the paper presents and analyzes the results of testing the selected set of functionalities of the implemented solution.

Keywords: AMI system, Cloud

1. UVOD

Osnovni zadatak tradicionalnih elektroenergetskih sistema bio je da se potrošačima obezbedi isporuka električne energije, a da to čine uz što niže troškove i uz odgovarajući kvalitet. Porast broja stanovnika, njihova potreba za električnom energijom, kao i promene u načinu na koji se električna energija proizvodi, prenosi, distribuira i troši, dovele su do nastanka pametnih elektroenergetskih mreža (*Smart Grid*). *Smart Grid* uvodi dvosmernu komunikaciju između preduzeća i potrošača i donosi promenu u načinu na koji se pristupa zahtevima potrošnje, sigurnosti i unapređenjima zaštite životne sredine.

U okviru *Smart Grid*-a, nalazi se i AMI (*Advanced Metering Infrastructure* - napredna merna infrastruktura). AMI sistem čine sistemi zaduženi za merenje, prikupljanje i analizu potrošnje električne energije, kao i sistemi zaduženi za komunikaciju sa pametnim mernim uređajima (*smart meters*).

Osnovni zadatak jednog AMI sistema jeste rad sa velikom količinom podataka, gde se podaci o potrošnji električne energije prikupljaju periodično. Često se u kontekstu AMI sistema spominje i *Cloud*. *Cloud* podrazumeva pružanje neograničenih usluga i deljenje istih putem interneta, gde korisnici plaćaju iznajmljivanje usluge kako bi koristili date resurse.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Čapko, vanr. prof.

Cloud čine infrastruktura (računarski resursi), centri podataka i virtualizacija, koja omogućava raspodelu fizičkog resursa na više virtualnih resursa koji se mogu nezavisno i paralelno koristiti prema potrebama korisnika. Upotreba *Cloud* usluga u AMI sistemu doprinosi visokoj pouzdanosti i dostupnosti podataka. Imajući u vidu da količina podataka u AMI sistemu varira u zavisnosti od doba dana, *Cloud* platforme koriste se upravo u cilju dobijanja skalabilnih i fleksibilnih resursa.

2. ARHITEKTURA AMI SISTEMA

AMI predstavlja integrisanu infrastrukturu koja podrazumeva dvosmernu komunikaciju između pametnih brojila i definisanih učesnika na tržištu. Pametna brojila šalju podatke o potrošnji električne energije, dok preduzeća imaju mogućnost upravljanja. AMI uključuje komunikacionu mrežu i pruža mogućnost automatskog čitanja potrošnje električne energije, bržeg i preciznijeg identifikovanja i izolovanja napada, brže restauracije i praćenja napona [1].

2.1. Infrastruktura AMI sistema

AMI se tipično sastoji od 3 komponente: pametna brojila, koja prikupljaju podatke u određenim vremenskim intervalima, komunikaciona mreža, koja prenosi podatke od pametnih brojila do MDMS-a (*Meter Data Management System*) i MDMS, koji skladišti i obrađuje podatke i povezuje ih sa ostalim sistemima [1]. AMI obuhvata i sistem upravljanja podacima kojim se vrši razmena podataka između svih učesnika u sistemu.

Pametna brojila omogućavaju merenje potrošnje transformatorskih stanica, domaćinstava, komercijalnih i industrijskih potrošača. **HAN** (*Home Area Network*) je bežična mreža u okviru manjeg prostora (kuća, kancelarija), gde je omogućeno povezivanje sa pametnim brojlilima putem kućnih aparata. Pametna brojila koriste PLC (*Power Line Communication*) i RF (*Radio Frequency*) sa koncentratorima/ruterima u okviru NAN (*Neighborhood Area Network*) mreže. NAN mreža je ograničena na određeno geografsko područje, obično na nivou transformatorske stanice. Koncentratori/ruteri su preko WAN (*Wide Area Network*) mreže povezani sa AMI aplikacionim centrima, kao što su NMS, MDMS i centrom podataka.

AMM (*Automated Meter Management*) podrazumeva daljinsko očitavanje, nadzor i upravljanje drugim komponentama AMI sistema, veliku brzinu obrade podataka, vezu sa drugim sistemima i transfer podataka u MDMS. Komunikacija između AMM sistema i ostalih

sistema vrši se putem WAN. **Koncentrator podataka** predstavlja uređaj koji je povezan sa pametnim brojilima i MDMS. Koncentrator mora da obezbedi neposrednu komunikaciju sa pojedinačnim brojilom. On automatski ili na zahtev izvršava funkcije očitavanja i memorisanja podataka sa brojila i predaje istih podataka na zahtev MDMS sistemu. Podaci sa pametnih brojila se prikupljaju putem NAN mreže, a dostavljaju MDMS-u putem WAN mreže. Komunikacija sa brojilima se odvija putem PLC komunikacije, a sa MDMS preko GPRS komunikacije [1]. Koncentrator ima i funkciju upravljanja/parametrizacije kojom se menjaju parametri brojila, a može se izmeniti i softver brojila. Obrada podataka na koncentratoru podrazumeva obradu alarma i sastavljanje izveštaja o kvalitetu isporuke električne energije koji se po zahtevu šalje MDMS sistemu. **MDMS** je deo AMI sistema. Uključuje procese prikupljanja podataka od koncentratora, obrađivanja i čuvanja istih. Obrada podataka podrazumeva detaljnu analizu gde se isti tipovi podataka grupišu, kako bi se olakšao postupak dobavljanja željenih podataka. Obrada podrazumeva i obračun utrošene električne energije, kako bi se obezbedili podaci sistemu za obračun i naplatu električne energije, kao i čuvanje obrađenih podataka.

2.2. Pametna brojila

Pametna brojila mogu da mere i snimaju stvarnu potrošnju energije tokom dana u određenom vremenskom intervalu. Ona šalju prikupljene podatke do MDMS-a. Pametna brojila omogućavaju bržu detekciju otkaza, reagovanje i restauraciju, kao i bolju informisanost klijenata o statusu elektroenergetske mreže.

Funkcionalnosti pametnih brojila su [2]:

- Dinamičko određivanje cena podrazumeva način na koji se zahteva ravnomerna maksimalna tražnja. U toku trajanja maksimalne potražnje, potrošači se podstiču da smanje potrošnju energije, pošto je cena tada najveća. Potrošači mogu smanjiti račun za utrošenu električnu energiju time što će smanjiti njeno korišćenje tokom perioda maksimalne potražnje.
- Dvosmerna komunikacija, koja podrazumeva slanje podataka od pametnih brojila do MDMS-a kada se zatraži i upravljanje pametnim brojilima od strane preduzeća.
- Daljinsko upravljanje od strane preduzeća za nekoliko funkcija, kao što su daljinsko isključivanje/povezivanje potrošača, detekcija prekida napajanja, slanje upozorenja potrošaču i slanje povratnih signala potrošaču.
- Detekcija kvara, kao i mogućnost obaveštavanja preduzeća kada je napajanje obnovljeno.
- Mogućnost nadogradnje softvera. Ovo omogućava mogućnost nadogradnje pametnog brojila, dodavanje novih funkcija i ispravljanje grešaka u ranijoj verziji softvera.

2.3. Karakteristike potrošača

Karakteristike potrošača mogu da se koriste za opisivanje krivih dnevnih opterećenja. Uočavaju se sličnosti između krive opterećenja korisnika istog tipa. Na primer, maksimalno opterećenje stambenih potrošača uvek se koncentriše u jutarnjim i večernjim časovima, a maksimalno opterećenje industrijskih potrošača u toku

dana 0. Razlike u potražnji za električnom energijom najčešće su primetne na sezonskoj osnovi, tokom sedmice i tokom dana. Na samu potražnju može uticati više faktora, među kojima i događaji poput popularnih emisija na televizijskim programima. Tipično, potražnja za električnom energijom je veća zimi nego leti, a takođe i niža u noćnim satima.

Što se tiče vremenskih intervala prikupljanja podataka, za domaćinstva je tipičan interval 1 sat, što je dovoljno za potrebe naplate. Za komercijalne i industrijske potrošače, preduzeća uglavnom koriste interval od 15 minuta za čitanje podataka sa pametnih brojila [2]. Preduzeća koriste podatke o potrošnji električne energije, kako bi dobili informacije o potrošačima koji su relevantni za optimizaciju njihovih programa energetske efikasnosti.

3. ANALIZA RADOVA AMI ARHITEKTURE

U radu [3], istraživana je dizajn optimalnih komunikacionih infrastruktura za AMI sistem. Tradicionalna AMI komunikaciona arhitektura podrazumeva MDMS koji je centralizovan i okružen glavnim operacijama i upravljačkim funkcijama. Potom je predložena arhitektura sa distribuiranim MDMS-om, gde se podaci šalju od pametnih brojila do koncentratora, koji dalje prosleđuju svom MDMS-u, a koji je povezan sa centralnim MDMS-om. U okviru istog rada predložena je i potpuno distribuirana AMI komunikaciona arhitektura, koja podrazumeva distribuiranje svih komponenti sistema.

U radu [7] analizira se upotreba *Cloud*-a u AMI sistemu, gde preduzeća razvijaju i održavaju servise na *Cloud*-u, kojima pristupaju pametna brojila i tako kontrolišu uređaje u vidu odgovora koji stiže sa *Cloud*-a. Ukoliko provajder servisa na *Cloud*-u ažurira ovu uslugu, pametno brojilo nije svesno ove promene, već će u porukama sa servisa dobijati drugačije odgovore.

4. OPIS I ARHITEKTURA IMPLEMENTIRANOG SISTEMA

Model podataka implementiranog rešenja je modelovan upotrebom CIM (*Common Information Model*) apstraktnog UML (*Unified Modelling Language*) modela kojim su predstavljeni elementi elektroenergetskog sistema, gde je za svakog **potrošača** vezano više analognih signala, tj. merenja koja se odnose na **aktivnu snagu (P)**, **reaktivnu snagu (Q)** i **napon (V)**, kao i **transformator** koji sadrži signal za merenje napona. Oni su grupisani na nivou **transformatorske stanice**, odnosno **geografskog područja**. Takođe, za svakog potrošača i transformator definiše se i **naponski nivo**, koji ima podatak o vrednosti nominalnog napona. Merenja imaju podatak i o adresi RTU uređaja za koji su vezani, kao i o jedinici merenja (P, Q i V). Potrošači i transformatori sadrže podatak o validnom opsegu napona, dok potrošači sadrže podatak i o tipu potrošača (domaćinstvo, tržišni centar i firma) i maksimalnoj vrednosti aktivne i reaktivne snage.

Prethodno opisani objekti u sistemu predstavljaju statičke podatke sistema i nisu podložni izmeni. S obzirom na prirodu sistema, gde se merenja (P, Q i V) prikupljaju periodično, korišćena je posebna klasa koja od atributa ima referencu na potrošača, attribute koji predstavljaju

izmerene vrednosti P, Q i V, vreme kada su izmerene vrednosti, podatak da li je potrošač u alarmu i tipu alarma (previsok ili prenizak napon).

Za razvoj aplikacija u *Cloud* okruženju korišćena je *Service Fabric* platforma, koja omogućava razvoj distribuiranih aplikacija. *Service Fabric* je zasnovan na mikroservisima koji podrazumevaju softversku arhitekturu u kojoj se aplikacija sastoji od manjih, nezavisnih procesa, koji komuniciraju međusobno preko standardnih protokola. Mikroservisi se smeštaju na klasterima. **Klaster** (engl. *Cluster*) predstavlja skup čvorova, odnosno grupu fizičkih ili virtualnih mašina koji su povezani kako bi obezbedili pouzdano i dostupno okruženje za pokretanje aplikacija. Klasteri se mogu napraviti u **lokalnom okruženju**, gde je jedna fizička mašina logički podeljena na nekoliko čvorova, u okviru **Azure okruženja**, gde se može koristiti veći broj mašina, ili na nekom drugom *Cloud* servisu. **Čvor** (engl. *Node*) predstavlja fizičku ili virtualnu mašinu koja pripada određenom klasteru.

Implementirano softversko rešenje sadrži infrastrukturu za daljinsko očitavanje električne energije, daljinsko očitavanje statusa napajanja individualnog potrošača, kao i očitavanje vrednosti napona na mestu potrošača. Fokus rada bio je na arhitekturi koja omogućava akviziciju i obradu podataka. Aplikacije koje rade u lokalnom okruženju su: Klijent, SCADA (*Supervisory Control And Data Acquisition*) i simulator. Funkcionalnosti datih aplikacija, kao i ostalih komponenti sistema koje se nalaze na *Cloud*-u slede u nastavku.

Korisnik je u mogućnosti da preko **klijentske aplikacije** šalje statičke podatke sistema u vidu CIM/XML dokumenta. Ukoliko je validacija poslatih podataka prošla uspešno, korisnik u toku rada aplikacije ima uvid u trenutne vrednosti podataka koji se menjaju, uključujući potrošnju električne energije i vrednosti napona. Dati podaci odgovaraju realnim situacijama. Korisnik ima mogućnost pregleda krive potrošnje i napona po određenim kriterijumima koji se odnose na odabir tipa sezone (leto i zima), tipa dana (radni dan i vikend), tipa potrošača (domaćinstvo, tržišni centar i firma) i specifičnog dana. U slučaju alarmnih situacija, korisnik će dobiti obaveštenje o vremenu generisanja alarma, nazivu potrošača i tipu alarma (previsok ili prenizak napon).

Implementirano rešenje pruža podršku za rad sa više simulatora koji u ovom slučaju predstavljaju RTU uređaje. Za komunikaciju između SCADA-e i simulatora korišćen je dnp3 protokol (*Distributed Network Protocol*). Merenja koja simulator šalje su **aktivna snaga**, **reaktivna snaga** i **napon**. Prilikom prijema merenja od simulatora, SCADA vrši proveru vrednosti napona i ukoliko je izvan validnog opsega označava alarm za dato merenje. Takođe, SCADA šalje sva merenja do *Calculation Engine*-a.

U procesu distribuirane transakcije omogućeno je samo dodavanje novih entiteta. Ideja **koordinatora transakcije** jeste da započne transakciju, proveriti da li sva 3 servisa (NMS, SCADA i *Calculation Engine*) mogu da pređu iz jednog konzistentnog stanja u drugo i da na osnovu rezultata završi transakciju (*Commit* ili *Rollback*).

NMS opisuje statičke elemente sistema. NMS je svestan samo statičkih podataka sistema i ima zadatak da dostavlja podatke drugim servisima.

U okviru *Calculation Engine*-a podržan je sledeći set funkcionalnosti: primanje izmerenih podataka od strane SCADA-e i slanje do klijentske aplikacije, komandovanje u slučaju detekcije alarma i agregacija mernih podataka u više nivoa. Primljeni podaci od SCADA-e su agregirani u 3 nivoa i upisuju se u minutnu, satnu i dnevnu tabelu. Agregirani podaci služe klijentu da u zavisnosti od tražene rezolucije dobije grafički prikaz potrošnje električne energije uz statističke podatke. Servis za agregaciju u unapred definisano vreme pokreće procese agregacije na pomenutim nivoima. Merenja u odnosu na koje se vrši agregacija se dobijaju od SCADA-e na svakih 30 sekundi.

5. REZULTATI IMPLEMENTIRANOG SISTEMA

Testiranje je rađeno u lokalnom i Azure okruženju. Testiranje je vršeno redom sa 1.000, 10.000, 100.000 i 1.000.000 objekata. U 1.000 objekata nalazi se 250 potrošača i 750 merenja (po 250 merenja aktivne snage, reaktivne snage i napona), i tako proporcionalno za veći broj objekata. Svako pojedinačno testiranje je vršeno sa 20 ponavljanja, pa dobijeni rezultati predstavljaju prosečnu vrednost istih.

Trajanje distribuirane transakcije obuhvata konverziju podataka u odgovarajući format i dodavanje istih u bazu podataka svake od komponenti. Rezultati testiranja prikazani su u tabeli 1.

Tabela 1: Trajanje distribuirane transakcije

Broj objekata	Lokalno okruženje	Azure okruženje
1,000	27.800 s	8.384 s
10,000	4 min 27 s	1 min 34 s
100,000	45 min 11 s	15 min 32 s
1,000,000	7 h 40 min	2 h 33 min

Rezultati testiranja detekcije alarma na *Calculation Engine*-u i slanje komande ka SCADA-i prikazani su u tabeli 2. Svako slanje merenja je sadržalo 10% potrošača koji su u alarmu. Za 1.000 objekata, sledi da je 25 potrošača u alarmu.

Tabela 2: Detekcija alarma i komandovanje

Broj objekata	Lokalno okruženje	Azure okruženje
1,000	0.302 s	1.314 s
10,000	0.922 s	3.888 s
100,000	9.848 s	31.532 s
1,000,000	1 min 41 s	5 min 19 s

Calculation Engine vrši upis novodobijenih merenja u bazu podataka, čiji rezultati su prikazani u tabeli 3.

Tabela 3: Trajanje upisa merenja u bazu podataka

Broj objekata	Lokalno okruženje	Azure okruženje
1,000	0.110 s	0.515 s
10,000	0.971 s	6.015 s
100,000	12.425 s	1 min 11 s
1,000,000	1 min 56 s	12 min 44 s

Testiranje je vršeno i za proces slanja merenja od strane SCADA-e, pa sve do prikaza izmenjenih vrednosti u vidu tabele na klijentskoj aplikaciji, što je prikazano u tabeli 4. Dobijena vremena uključuju i vremena dobijena testiranjem iz tabela 2 i 3. Takođe, vremena koja su dobijena ovim testiranjem uključuju i sumiranje merenja na nivou pripadnosti transformatorskoj stanici i geografskom području, kao i upisivanje rešenih alarma u bazu podataka od strane *Calculation Engine-a*.

Tabela 4: *Trajanje slanja merenja*

Broj objekata	Lokalno okruženje	Azure okruženje
1,000	0.584 s	3.319 s
10,000	2.851 s	13.057 s
100,000	28.789 s	1 min 50 s
1,000,000	4 min 47 s	17 min 57 s

Vremensko trajanje distribuirane transakcije pokazalo je da se povećavanjem broja redova koji se upisuje u bazu podataka svake od komponenti koja učestvuje u distribuiranoj transakciji povećava približno linearno. Sam proces distribuirane transakcije u okviru Azure okruženja brži za oko 3 puta u odnosu na isti proces u lokalnom okruženju.

Proces detekcije alarma na *CalculationEngine-u* i slanje komande na SCADA-i je za oko 4 puta brže u lokalnom okruženju, u odnosu na Azure okruženje.

Perzistencija pristiglih merenja na *CalculationEngine-u* na oba testirana okruženja ima približno linearni rast sa povećavanjem broja merenja. Samo upisivanje merenja u bazu podataka je za oko 6 puta brže u lokalnom okruženju. Imajući u vidu da sam upis u Azure okruženju vrši jedna mašina, koja ima slabije performanse od lokalnog računara, kao i činjenica da se baza podataka u lokalnom okruženju nalazi na istom računaru kao i sama aplikacija, pa iz navedenih razloga sledi zaključak zašto je ovaj proces brži u lokalnom okruženju.

Rezultati procesa slanja merenja od SCADA-e do klijentske aplikacije su pokazali da je sa manjim brojem merenja navedeni proces brži u lokalnom okruženju za oko 5 puta u odnosu na Azure okruženje, dok je sa većim brojem merenja proces u lokalnom okruženju brži za oko 3.5 puta. Treba imati u vidu da iako klijentska aplikacija nije deo *Cloud* okruženja, ona se u lokalnom okruženju nalazi na istom računaru kao i aplikacije koje su deo *Cloud-a*.

Rezultati koji su dobijeni pokazuju da na brzinu testiranih funkcionalnosti bitno utiče i karakteristika mašine. Na primeru distribuirane transakcije pokazano je da bitnu ulogu igra broj mašina u okviru klastera, odnosno raspodela procesa upisivanja u bazu podataka na svakoj od mašina. Zbog ograničenja u vidu performansi računara, testiranje je obuhvatilo slanje redova do veličine 1.000.000. Očekivano je da u realnim sistemima broj merenja bude i veći.

6. ZAKLJUČAK

U ovom radu predstavljen je AMI sistem uz poseban osvrt na komunikacionu infrastrukturu, pametna brojila i tipove potrošača. Analizirani su radovi iz oblasti arhitekture AMI sistema i korišćenja *Cloud* usluga u takvim sistemima, koji su ukazali na benefite i probleme svakog

od predloženih rešenja. Implementirano rešenje je obuhvatilo izabrani set funkcionalnosti, uz poseban osvrt na alarme, slanje podataka prema tipu potrošača i agregaciju podataka u istorijsku bazu podataka.

Vršeno je poređenje vremenskih rezultata pojedinačnih funkcionalnosti, kao i rezultata dobijenih testiranjem u okviru izabrana dva *Cloud* okruženja, odnosno lokalnog i Azure okruženja.

Razmatrani pravci daljeg razvoja su testiranje implementiranog sistema u okviru *Cloud* okruženja gde bi mašine u okviru klastera imale bolje karakteristike. Uz to, moguć je i odabir drugog *Cloud* provajdera, pa bi se dobijeni rezultati mogli porediti između svih izabranih *Cloud* okruženja.

7. LITERATURA

- [1] JP Elektroprivreda Srbije, "*Funkcionalni zahtevi i tehničke specifikacije AMI/MDM sistema*", 2014.
- [2] G. R. Barai, S. Krishnan, and B. Venkatesh, "*Smart metering and functionalities of smart meters in smart grid - A review*", Electrical Power and Energy Conference (EPEC), 2015.
- [3] J. Feng, "*An analytics of electricity consumption characteristics based on principal component analysis*", IOP Conference Series: Earth and Environmental Science, pp. 0–6, 2018.
- [4] U.S. Department of Energy, "*Advanced Metering Infrastructure and Customer Systems: Results from the Smart Grid Investment Grant Program*", Off. Electr. Deliv. Energy Reliab., pp. 1–98, 2016.
- [5] H. Bai, "*Programming Microsoft Azure Service Fabric*", Microsoft Press, 2016.
- [6] R. Q. Hu, J. Zhou, R. Q. Hu, and S. Member, "*Scalable Distributed Communication Architectures to Support Advanced Metering Infrastructure in Smart Grid*", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 9, pp. 1632–1642, 2012.
- [7] M. Yigit, V. C. Gungor, and S. Baktir, "*Cloud Computing for Smart Grid applications*", Comput. Networks, vol. 70, pp. 312–329, 2014.

Kratka biografija:



Bratislav Batinić je rođen 18.10.1994. godine u Bačkoj Topoli. Završio je srednju ekonomsku školu "Dositej Obradović" u Bačkoj Topoli 2013. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 2017. godine. Master studije na istom fakultetu je upisao školske 2017/2018. godine. Ispunio je sve obaveze i položio je sve ispite predviđene studentskim programom.