

Branimir Gabrić
HEP-ODS d.o.o.
Branimir.gabric@hep.hr

Kristijan Frano Ćavar
HEP-ODS d.o.o.
KristijanFrano.Cavar@hep.hr

Ivan Periša
HEP-ODS d.o.o.
Ivan.perisa@hep.hr

MJERINFO, PLATFORMA ZA RAD S VREMENSKIM SERIJAMA U PROCESNIM SUSTAVIMA

SAŽETAK

Na prošloj CIRED smotri je predstavljen rad InfluxDB, baza za pohranu vremenskih serija i njena usporedba s Oraclom. U međuvremenu je InfluxDB iskorišten kao srce sustava te je izgrađen čitav okoliš temeljen na softveru otvorenih tehnologija koje su trenutno popularne u svijetu IT industrije: MongoDB za pohranu meta podataka o mjerenjima, Postgre za događaje, .NET Core kao API Gateway, Go za preko 20 adaptera. U sustav se slijevaju mjerenja iz SCADA-e, AMI-ja, brojila sa sučelja HOPS-ODS, događaji i indikacije. Bilo da nad platformom razvija web aplikaciju, raspisuje izračun u Python-u (Pandas) ili se pak povezuje s postojećim informacijskim sustavom, korisniku su podaci dostupni preko REST poziva u lako čitljivom JSON formatu.

Ključne riječi: InfluxDB, Open source, vremenske serije, REST, SCADA mjerenja i događaji

MJERINFO, TIME SERIES DATA PLATFORM IN OT SYSTEM

SUMMARY

Past CIRED featured the work of InfluxDB, a time series storage base and its comparison with Oracle. Meanwhile, InfluxDB has leveraged the heart of the system and surrounded by variety of open-source software: MongoDB for metadata storage, Postgre for Events and Logs, .NET Core as an API Gateway, Golang is used for over 20 adapters. Measurements from operational technology systems such as SCADA, AMI, HOPS-ODS interface counters, events and indications flow are gathered in the system. Whether is developing a web application over the platform, publishing a calculation in Python (pandas), or connecting to an existing information system, the user invokes the information via REST in an easy to read JSON format.

Key words: InfluxDB, Open source, Time Series Database, REST, SCADA Measurements and Events

1. UVOD

Ideja za izradu sustava koji će sadržavati sva dostupna mjerenja s vremenskom oznakom polazi od prezentacije firme OSI Soft, čiji proizvod PI System ima upravo takovu namjenu još od ranih 80-tih godina -nedostatak mu je visoka cijena.

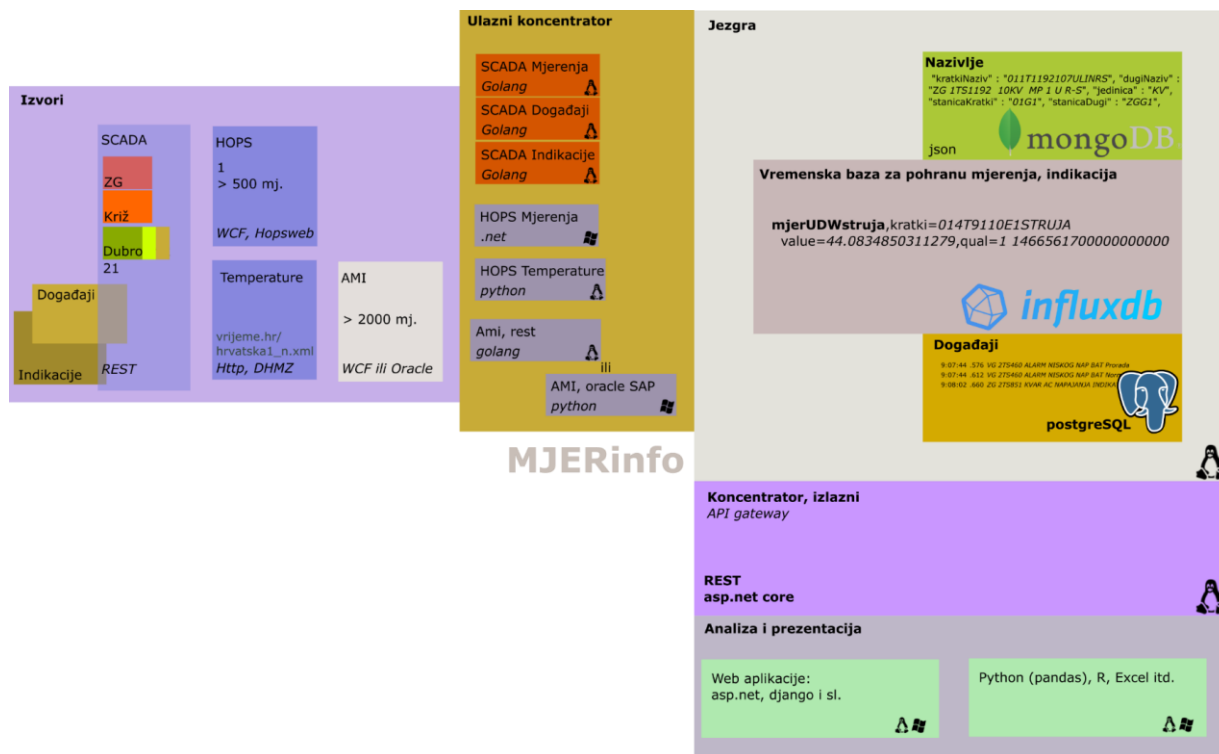
Zaneseni konceptom, pokušali smo ga realizirati koristeći minimalna financijska sredstava. Situacija koja je svakako pogodovala mogućnosti realizacije jest proplamsaj projekata otvorenog koda (*Open-source*) u zadnjih 15-20 godina. Skoro svaki renomirani softver ima svoju reprezentativnu alternativu otvorenog koda npr.: Windows-Linux, Photoshop-Gimp, Illustrator-Inkscape itd. Brojnost IoT uređaja zahtjeva sustav koji može primiti velike količine mjerenja u jedinici vremena; razlog tome je i nastanak baze za pohranu vremenskih serija – InfluxDB firme InfluxData. Riječ je o bazi otvorenog koda koja iznimno brzo pohranjuje vremenske serije (100k mjerenja u sekundi) te je mnogo štedljivija u pohrani u odnosu na konvencionalne relacijske baze podataka. Usporedba InfluxDB i Oracle baze opisana je u prethodnom CIRED radu [1].

InfluxDB je iskorišten kao središte probnog (*Proof of Concept*) sustava koji unutar firme ima radni naziv MJERinfo. InfluxDB je srce sustava, a poduprijet je s brojnim i raznolikim softverom otvorenog koda. U kontekstu Službe za procesne sustave i telekomunikacije HEP ODS-a, riječ je o zapravo o prikupljanju mjerenja iz SCADA-e, meteorološkim podacima te mjerenjima iz susjednih sustava koji su potrebni za pravilno vođenje pogona: mjerenja operatora prijenosnog sustava te mjerenja sa sučelja pametnih brojlara.

U konačnici, ideja platforme je podržati razvoj novih aplikacija u procesnom sustavu te u ostalim sektorima unutar firme. Ideja je poticanje inženjera za korištenjem Python programskog jezika kako bi obrađivali mjerenja pohranjena u sustavu te potom donosili kvalitetnije poslovne odluke.

2. OPIS SUSTAVA

2.1. Arhitektura



Slika 1. Grafički prikaz arhitekture sustava MJERinfo platforme

Na grafici iznad prikazana je logička arhitektura sustava. Promatrajući u smjeru kazaljke na satu, na početku se nalaze *Izvori*. Taj kvadrat predstavlja vanjske izvore podataka; riječ je brojnim adapterima koji dohvaćaju podatke iz vanjskih sustava, najčešće se nalaze na web poslužiteljima samih vanjskih sustava i najčešće su to web servisi, u nekim slučajevima i skripte. Potom slijedi *Ulazni koncentrator* koji proziva web servise iz prethodnog kvadrata. Podaci se obrađuju, normaliziraju i prosljeđuju *Jezgri* sustava. Vrijednosti s vremenskom oznakom ulaze u InfluxDB bazu, dodatne informacije o mjerenjima u MongoDB bazu, događaji u Postgre.

U ljubičastom kvadratu prikazan je API Gateway, koji korisniku izlaže programske metode (API-ji) (više u poglavlju 3.1.) i kojima se pristupa podacima: doznaju informacije o kontekstu mjerenja, njegovu vrijednost i sl. Konačno, u zadnjem, donjem-desnom kvadratu prikazane su same primjene: primjerice web aplikacije s bogatim prikazom ili tek Python bilježnice s odgovarajućim proračunima korisnika.

2.2. Korištena računalna oprema i operacijski sustavi

Sustav trenutno pogone dva računala. Jedno je serversko računalo iz 2014. godine na kojem se nalaze baze podataka. Riječ je o 4-jezgrenom Xeon (E3-1270) računalu sa 16 GB RAM-a i 300 GB diskovnog prostora. Koristi Ubuntu server 18.04 LTS. Drugo računalo je nešto starije i slabije (6GB RAM-a i Xeon W3520 iz 2011.), na njemu se trenutno nalazi testni sustav i pogoni API Gateway, 6GB rama i 4-jezgreno računalo pogonjeno Mint-om 21 (Ubuntu 18.) Oba računala su spojena na neprekidno napajanje.

2.3. Korištene softverske tehnologije

Za napomenuti je kako se projektu doprinosilo u brojnim kraćim vremenskim porcijama kroz više godina, tako da se u njemu zrcale raznolike tehnologije koje su u odgovarajućem trenutku bile interesantne. Na samom početku, proljeće 2017., za pohranu mjerenja s vremenskom oznakom odabrana je InfluxDB baza – čini srce sustava. Potom je odabrana baza MongoDB za pohranu meta podataka o mjerenjima; NoSQL baza je odabrana jer je ideja bila da se poslije s lakoćom može mijenjati struktura podataka o mjerenjima koja je kroz duže vrijeme razvoja bila podložna promjenama. PostgreSQL je odabran za pohranu događaja. Jedna instanca Postgrea (nije prikazano na grafici iznad) je i sustav za pohranu logova (nalazi se u *Docker* kontejneru, a kojem se pristupa koristeći REST – PostgREST dodatak). Za API Gateway koji brine o REST sučelju prema korisniku izabrana je web API varijanta Microsoftove ASP.NET Core 3 tehnologije (odnedavno OSS). Za napomenuti je da korisnik nema direktan pristup bazama koje čine sustav, već samo API Gatewayu, koji pak, koristeći prikladne *pluginove*, pristupa mjerenjima i meta podacima, nadalje, zaslužan je i za konačni JSON izlaz. Inače, izravan pristup Influxu i pokretanje nekih najobičnijih SQL naredba može dovesti do nestabilnosti sustava (*overkill*).

Početni rad sa sustavom je krenuo koristeći programski jezik R, ali je uskoro zamijenjen s Pythonom. Veliki dio kontrolnih skripti, skripti za automatizirani unos *bulk* podataka izveden je koristeći Python 3. Adapteri koji podatke direktno šalju u InfluxDB izrađeni su u Golangu (InfluxDB je također izrađen u Golangu (dizajniran u Googlu) . Svi web servisi koji se nalaze na web poslužiteljima SCADA sustava su napisani koristeći C# pri Microsoft WCF tehnologiji. Skripte za dohvat AMI podataka su napisane koristeći .Net Core 3 u klijent varijanti. Također, Api Gateway je izrađen koristeći .Net Core web API koristeći jezik C# (koriste se iznimno moćni LINQ upiti za strukturiranje i filtriranje podataka, te C# 6 specifikacija koja omogućuje *null type reference* (npr. *int?*) što je korisno u formatiranju JSON izlaza gdje mjesto vrijednost, ukoliko je nema, piše *null*.)

Za pravodobno pokretanje skripti koristi se *Linuxov crontab*, odnosno *Task scheduler* za neke skripte koje se nalaze na Windows računalu. JSON se koristi za razmjenu podataka, a za postavke vrlo praktični YAML.

2.3. Interna validacija

Kako je izvor podataka iz kojih se dohvaćaju mjerenja mnogobrojan i raznovrstan, podložan je svakojakim greškama: greške mogu biti na izvorištu (manjak podataka u arhivskoj bazi, pad servera i sl.), na razini komunikacije, problemi s mrežom, na odredištu (neotkrivene greške u skriptama) itd.

Uglavnom, svaki dohvaćeni podatak zapisan je u mapu kao datoteka u JSON formatu. Izrađene su Python skripte koje provjeravaju mapu i analiziraju veličinu datoteke s obzirom na očekivanu (ne zalaze u dublju analizu). Ukoliko je veličina bitno manja od prosječne veličine 15 minutnog seta podataka, ovaj ulazi u popis za ponovni dohvat. Iduća skripta ponovno okida dohvat podataka. Unutar ekosustava, o kojem će više riječi biti u nastavku rada, napravljena je i kontrolna aplikacija koja šalje email obavijest ukoliko je korpus podataka za odgovarajući sustav necjelovit (npr. za SCADA sustav, tako da često može upućivati i na probleme s arhiviranjem podataka na matičnom sustavu). Važno je napomenuti da se sustav vremenski sinkronizira na NTP servere unutar poslovne mreže HEP-a. Vremenski intervali su po UTC-u i početni su (za vremenski raspon od 13:00-13:15, ovaj zapisuje vremensku oznaku 13:00), nadalje zapis vremena koji se provlači kroz cijelu platformu poštuje normu RFC 3339.

3. KORIŠTENJE SUSTAVA

3.1. Primjeri web API-ja

Primjeri web API-ija i odgovarajući rezultat će biti prikazan u sljedećoj tablici:

Tablica 1. Neki primjeri web API-ija

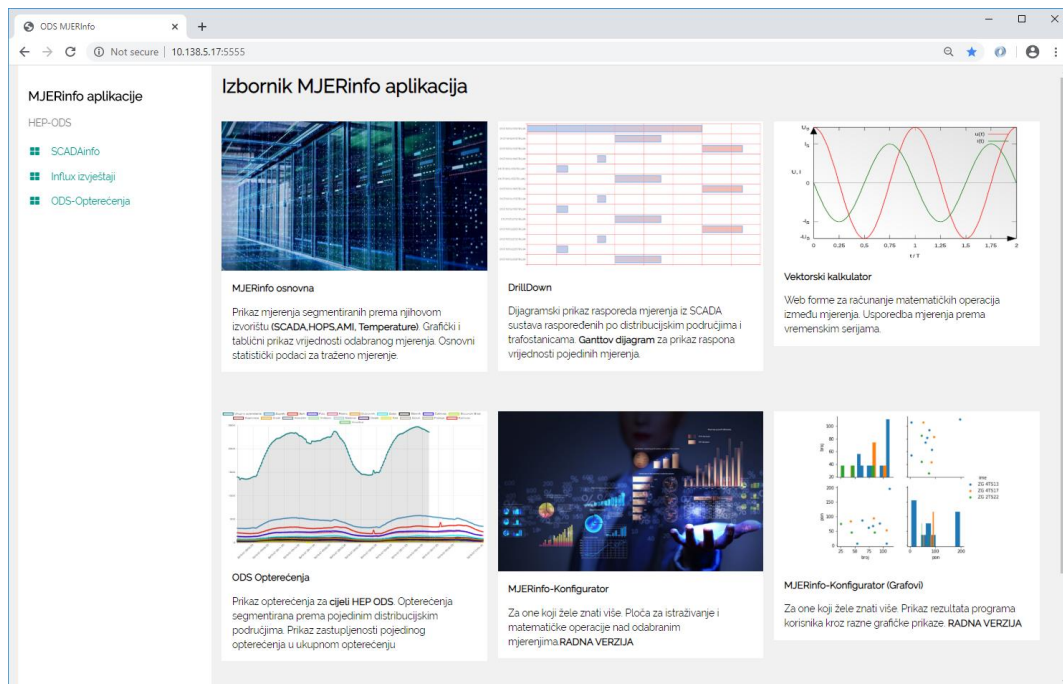
Opis A:	Dohvat informacija o DP-ovima.
Metoda A:	http://localhost:8083/api/mjerinfo/sifraDp
Rezultat A:	<pre>{ sifra: "4001", naziv: "Elektra Zagreb", brojStanica: 0 }, { sifra: "4002", naziv: "Elektra Zabok", brojStanica: 0 }, ...</pre>
Opis B:	Dohvat informacija o trafostanicama.
Metoda B:	http://localhost:8083/api/mjerinfo/stanice/4001
Rezultat B:	<pre>{ kratkiNaziv: "013T1", dugiNaziv: "ZG3TS1", opis: "Kršnjavoga", brojMjerenja: 44 }, { kratkiNaziv: "013T10", dugiNaziv: "ZG3TS10", opis: "", brojMjerenja: 31 }, ...</pre>
Opis C:	Osnovne informacije o mjerenju:

Metoda C:	.../api/mjerinfo/mjerenja/4001?stanicaKratki=013T1
Rezultat C:	<pre> { kratkiNaziv: "013T110J13STRUJA", dugiNaziv: "ZG 3TS1 10KV 1KV1904 STRUJA" }, { kratkiNaziv: "013T110J17STRUJA", dugiNaziv: "ZG 3TS1 10KV 1KV221 STRUJA" },... </pre>
Opis D:	Osnovna stat. obrada za odabrano mjerenje i vremenski raspon.
Metoda D:	.../api/mjerinfo/scada/statmjer/4001/20190919T10000Z/20190922T110000Z?kratkiNaziv=013T110J13STRUJA
Rezultat D:	<pre> { kratkiNaziv: "013T110J13STRUJA", min: 37.22047424316406, max: 91.71243286132812, spread: 54.49195861816406, count: 292, distinct: 276, mean: 56.92088033075202, median: 47.78158760070801, stddev: 16.264179826966632, mode: 44.545536041259766 } </pre>

3.2. MJERinfo ekosustav aplikacija

MJERinfo platforma omogućuje razmjenu podataka s drugim informacijskim sustavima koristeći REST arhitekturu. Web API metode su dokumentirane i dodatno opimjerene na Jupyterhubu (podignut unutar intraneta firme).

Na Slici 2. je samo početna slika, dok su aplikacije MJERinfo ekosustava detaljno opisane u posebnoj ovogodišnjoj CIRED radu [2]. su u spomenutom radu opisane sve aplikacije ponaosob . Pojedine aplikacije prikazuju podatke tablično, neke u obliku grafa, posebno je interesantna aplikacija kojom se omogućuje propadanje kroz podatke (*drill down*): započinje se s odabirom izvora podataka, a završava na pojedinom mjerenju. Neke aplikacije su koncipirane na način da ciklički rade statističke obradu, a ukoliko su podaci alarmantni, korisnik biva obaviješten e-porukom.



Slika 2. Početna stranica s popisom MJERinfo aplikacija

3.4. Zajednica korisnika

Ideja platforme pored same pohrane svih mjerenja s vremenskom serijom na jednom mjestu jest i potaknuti samostalne inženjere na razvoj vlastitih programskih rješenja. Korisnicima je omogućen dokumentirani popis web API poziva koji funkcioniraju unutar intraneta firme. Dovoljno je instalirati Python programski jezi; s osnovnom instalacijom Pytona se dobije početni rudimentarni IDE (razvojno okruženje). Preporuka je koristiti neko naprednije poput Sypder, Visual Studio Code-a, Anaconde. Korisnicima se omogućuje i spajanje na Jupyterhub (također unutar intraneta) gdje korisnik u Jupyter bilježnici može razvijati svoj program. Ukoliko se pokaže da je izračun koristan i ima perspektivu češćeg korištenja, vrlo ga je lako, uz neznatnu izmjenu, generalizirati te postaviti kao web aplikaciju unutar MJERinfo ekosustava aplikacija (Python Django).

3.5. Neke brojke

U ovom potpoglavlju ćemo koristiti brojke kako bi čitatelj stekao kvalitetniju predodžbu sustava. U sustavu se trenutno nalazi preko 70k mjerenja. Podatci su od 1.1.2018. godine, 15 minutne vrijednosti, a gdje nema 15 minutnih, satne su. Odgovarajuća količina podataka je i iz 2014. jer je napravljen test importa povijesnih podataka s trake. Tjedno u sustav ulazi preko 20 milijuna SCADA mjerenja. Za dvije godine SCADA+HOPS podataka na disku je zauzeto 6,7 GB. Za zaključiti je iz postojećih izračuna da će za SCADA i HOPS podatke svakih 5 godina biti zauzeto dodatnih 16,7 GB diskovnog prostora.

3.6. Još za implementirati

Ono što još nije implementirano, a spomenuto je u sažetku, odnosi se na pohranu indikacija. U trenutku pisanja sažetka rada se mislilo da će i ova stavka kroz pola godine biti sastavnica sustava, ali za sada još nije. Nešto o čemu se razmišlja je podrška za CIM. Svako mjerenje u bazi ima odgovarajući

UUID koji je dobar temelj za mRID kojeg CIM zahtjeva. Koristile bi se klase navezane na *MeterReading* klasu. Izlaz bi bio konvencionalno u XML-u ili u nečemu što bi se moglo nazivati CIMJSON.

Za implementirati ostaje da se svi elementi sustava podignu u *Docker* kontejneru (za sada je u *Dockeru* samo čvor za pohranu logova). Na taj način platforma može biti neovisna o operacijskom sustavu koji je pogoni, je li instaliran lokalno, na oblaku i sl.

3.7. Prednosti i nedostaci

Za početak prednosti: u projekt nisu uložena izravna novčana sredstva, koristi se postojeća infrastruktura, svi alati su besplatni, radno vrijeme uloženo u projekt moglo bi se kategorizirati kao vrijeme za edukaciju i istraživanje. MJERinfo je satkan od moćnih, dobro ispitanih rješenja iz svijeta otvorenog koda. Sustav može održavati jedna osoba i za sada je riječ o prekapacitiranom sustavu s obzirom na potrebe. Kod je otvoren i lako čitljiv, web API je jednostavno nadograđivati/održavati. Brojne su mogućnosti povezivanja s popularnim rješenjima za vizualizaciju podataka: Grafana, Kapacitor i sl. Pad sustava nije krucijalan jer se naslanja na stabilne baze podataka matičnih sustava. Projekt se lako može postaviti u kontejnere i podići na oblak.

Nedostatci: sustav je nedovoljno testiran. Nije uloženo vrijeme u računalnu sigurnost. Za neke dodatne naprednije stvari bi trebalo uključiti dodatna znanja i vještine, dodatne ljude. Primjerice, pohrana topologije, (početna ideja bila je pohraniti je na Neo4j platformi), potom CIM izvoz i sl., tražili bi svakako dodatni razvoj i resurse i nadilazili bi okvir projekta kojeg trenutno bez većeg zamora može održavati/razvijati samo jedna do dvije osobe.

4. ZAKLJUČAK

Kao što se da zaključiti iz prethodno napisanoga, riječ je o konceptu koji je nastao unutar kuće, potaknut prezentacijom firme OSI-soft. Prepoznala se potreba centralizacije svih mjernih podataka i događaja, a kako bi se isti mogli lakše obrađivati, uspoređivati, kako bi se nad njima mogle raditi osnovne i napredne kalkulacije (npr. strojno učenje).

Poseban naglasak je na širenju unutar zajednice korisnika; pod time se misli na samostalne inženjere koji bi procesne podatke mogli u par linija koda dohvatiti te analizirati koristeći vrlo moćnu *Pandas* biblioteku. Tako izrađene programske rutine vrlo lako i skoro neizmijenjene mogu postati web aplikacije unutar MJERinfo ekosustava.

Sustav je otvoren prema popularnim *Open-source* rješenjima za vizualizaciju podataka. Dodavanje nove metode i/ili izmjena postojeće web API metode ne iziskuje mnogo vremena. Mada su sve stavke prikazane na Slici 1. u potpunosti automatizirane, ne može se govoriti o sustavu koji se podesi i zaboravi; neumoljivo je ovisan o preko 20 raznovrsnih drugih sustava koji su mu izvorište podataka.

Kao zaključak bi se moglo reći da je ovaj dokaz koncepta (PoC) opravdao izradu i postojanje. Naslijedit ga slobodno može sustav s većom mogućnosti podrške, satkan od drugih tehnologija otvorenog ili zatvorenog koda. Najvažniji plod ove platforme su izrađene aplikacije krojene iskustvima iz procesa: aplikacije iz MJERinfo ekosustava su opisane u komplementarnom radu na ovogodišnjem simpoziju [2].

5. LITERATURA

[1] B. Gabrić, M. Penzar „Primjena baze za pohranu vremenskih serija u procesnim sustavima“, 6. savjetovanje hrvatskog ogranka međunarodne elektrodistribucijske konferencije CIRED, Opatija, 13. - 16. svibnja 2018.

[2] K. F. Čavar, B. Gabrić „MJERinfo web aplikacije“, 7. savjetovanje hrvatskog ogranka međunarodne elektrodistribucijske konferencije CIRED, Šibenik, 17. - 20. svibnja 2020.