



[3] 2015 3[1]

АГГ+ часопис за архитектуру, грађевинарство, геодезију и сродне научне области  
ACEG+ Journal for Architecture, Civil Engineering, Geodesy and other related scientific fields

068-077 **Оригинални научни рад** | Original scientific paper

UDK | UDC 72.013:004.4

DOI 10.7251/AGGPLUS1503066A

Рад примљен | Paper received 17/12/2015

Рад прихваћен | Paper accepted 25/12/2015

**Младен Амовић**, Архитектонско-грађевинско-геодетски факултет Универзитета у Бањој Луци,  
Војводе Степе Степановића 77/3, 78 000 Бања Лука, Република Српска, [mamovic@agfbl.org](mailto:mamovic@agfbl.org)

**Миро Говедарица**, Факултет техничких наука Универзитета у Новом Саду, Др Зорана Ђинђића  
1, 21000 Нови Сад, Србија, [miro@uns.ac.rs](mailto:miro@uns.ac.rs)

**Владимир Пајић**, Факултет техничких наука Универзитета у Новом Саду, Др Зорана Ђинђића 1,  
21000 Нови Сад, Србија, [rajicv@uns.ac.rs](mailto:rajicv@uns.ac.rs)

**Славко Васиљевић**, Архитектонско-грађевинско-геодетски факултет Универзитета у Бањој  
Луци, Војводе Степе Степановића 77/3, 78 000 Бања Лука, Република Српска, [svasiljevic@agfbl.org](mailto:svasiljevic@agfbl.org)

ПРОСТОРНО - ВРЕМЕНСКИ  
ТИПОВИ И АНАЛИЗЕ ПОДАТАКА  
У BIG DATA ПАРАДИГМИ

SPATIO - TEMPORAL TYPES  
AND DATA ANALYSIS IN BIG  
DATA PARADIGM

Оригинални научни рад  
Original scientific paper  
Рад прихваћен | Paper accepted  
17/12/2015  
UDK | UDC  
72.013:004.4  
DOI  
10.7251/AGGPLUS1503066A

### Младен Амовић

Архитектонско-грађевинско-геодетски факултет Универзитета у Бањој Луци, Војводе Степе Степановића 77/3, 78 000 Бања Лука, Република Српска, [mamovic@agfbl.org](mailto:mamovic@agfbl.org)

### Миро Говедарица

Факултет техничких наука Универзитета у Новом Саду, Др Зорана Ђинђића 1, 21000 Нови Сад, Србија, [miro@uns.ac.rs](mailto:miro@uns.ac.rs)

### Владимир Пајић

Факултет техничких наука Универзитета у Новом Саду, Др Зорана Ђинђића 1, 21000 Нови Сад, Србија, [rajicv@uns.ac.rs](mailto:rajicv@uns.ac.rs)

### Славко Васиљевић

Архитектонско-грађевинско-геодетски факултет Универзитета у Бањој Луци, Војводе Степе Степановића 77/3, 78 000 Бања Лука, Република Српска, [svasiljevic@agfbl.org](mailto:svasiljevic@agfbl.org)

## ПРОСТОРНО-ВРЕМЕНСКИ ТИПОВИ И АНАЛИЗЕ ПОДАТАКА У *BIG DATA* ПАРАДИГМИ

Модел за управљање великим серијама просторно-временских података имплементиран је на Apache Spark open-source платформи за складиштење и обраду великих серија података на дистрибуираним рачунарским системима формираним од комерцијално доступних радних станица. Алгоритми за обраду просторно-временских података су дефинисани према правилима Spark SQL програмског модела, а релационе операције на DataFrame-овима (специјализованим системом оквира података) коришћењем специфичног језика домена (domain – specific – language → DSL). Увођењем просторно-временских типова података омогућава се стандардизован приступ у *Big Data* парадигми.

**Кључне ријечи:** Геопросторни подаци, *Big Data*, *Apache Spark SQL*, *Data Frames*, *OGC*

## SPATIO - TEMPORAL TYPES AND DATA ANALYSIS IN BIG DATA PARADIGM

### ABSTRACT

The model for managing large volumes of spatio-temporal data is implemented in Apache Spark platform for storing and processing large sets of data. The algorithms for processing spatio-temporal data are defined according to the rules of Spark SQL programming model and relational operations on dataframes (specialized system of data frames) using domain specific language (domain - specific - language → DSL). With the introduction of spatio-temporal data types, a standardized approach to a Big Data paradigm is enabled.

**Key words:** *Geospatial data*, *Big Data*, *Apache Spark SQL*, *Data Frames*, *OGC*

## 1. УВОД

Развој технологија омогућио је аквизицију, складиштење, управљање и обраду великих количина геопросторних података знатно већим брзинама. Стандардни приступи за њихово складиштење и обраду не могу да на задовољавајући начин одговоре на захтјеве које постављају толике количине података. Моделом управљања великим количинама геопросторних података унаприједиће се постојећи приступи у складиштењу великих количина геопросторних података прикупљени различитим геоинформационим технологијама и унаприједити њихова аналитичка обрада и комбиновање различитих типова података у аналитичкој обради. Модел архитектуре нових типова геопросторних података је заснован на Spark просторном упитном језику програмског модела који представља широко прихваћен приступ за управљање и генерисање великих серија података са паралелно дистрибуираним алгоритмом на cluster-у. Уз помоћ Spark SQL модула је интегрисано релационо процесуирање са Spark функционалним програмирањем апликације. Постојеће библиотеке у којима су дефинисане методе и геопросторни типови су проширене новим типовима података.

## 2. ДЕФИНИСАЊЕ ПРОБЛЕМА И ЦИЉ СТУДИЈЕ

Сателитска и радарска даљинска детекција, глобални навигациони сателитски системи, аерофотограметријске камере, сензорске мреже, ласерско скенирање допринијели су експоненцијалном порасту количине прикупљених просторно-временских података [1]. Паралелна обрада великих серија геопросторних података може допринијети да њихова анализа траје свега неколико секунди умјесто неколико сати. Опције за оптимално складиштење и обраду података су прошириле технологије, као што су MapReduce и in-memory computing, а које обезбеђују високо оптимизоване могућности. Модул Spark SQL нуди генерални оквир за трансформисање структуре, која је коришћена за извођење анализа, планирање и генерисање кодова у реалном времену [2] и [3].

### 2.1. ДОСАДАШЊИ РАДОВИ У ОВОЈ ОБЛАСТИ

Досадашња истраживања из области управљања просторно-временским типовима података у Big Data парадигми су релативно малобројна. Van Oosterom et al. су извршили детаљну анализу четири приступа за управљање великим скуповима тачака, првенствено над типом облака тачака: ORACLE RDBMS, PostgreSQL RDBMS, column store MonetDB i file system plus Rapidlasso LAStools за индексирање и обраду. Spark SQL представља модул у Apache Spark-у који интегрише референтне обраде Spark функционалног програмирања Апликације [4], [5] и [6]. Изграђена на Spark модулу, Spark SQL омогућава Spark програмерима велике предности приликом релационог процесуирања (нпр. декларативне упите и оптимизовано складиштење) и омогућава SQL корисницима да позивају комплексне аналитичке библиотеке у Spark (нпр. машинско учење). Посебан акценат се ставља у дефинисању нових типова података којима ће бити проширене постојеће библиотеке просторно-временских података којима су дефинисани основни геометријски примитиви. Њихова дефиниција као класе мора бити устројена тако да задовољи све елементе геометрије и топологије у складу са одговарајућим ISO стандардима у овој области. Библиотека GeoSpark проширује Apache Spark са просторним адаптивним дистрибуираним сетом података (Spatial Resilient Distributed Datasets – SRDD) који ефикасно учитавају, процесуирају и анализирају велике количине просторних података кроз машине. Она проширује RDD (Resilient Distributed Dataset) у форму

просторног RDD-а (SpatialRDDs) и ефикасно раздваја SRDD просторне елементе кроз машине и представља паралелизоване просторне (геометријске операције које прате Open Geospatial Consortium (OGC) трансформације и акције (за SRDD) да обезбиједи интерфејсе за кориснике приликом писања аналитичких програма за просторне податке [7], [8], [9] и [10].

## 2.2. МАТЕРИЈАЛИ И МЕТОДЕ

Модел за управљање великим серијама геопросторних података имплементиран је на Apache Hadoop open-source платформи за складиштење и обраду великих серија података на дистрибуираним рачунарским системима формираним од комерцијално доступних радних станица. Алгоритми за обраду геопросторних података дефинисани су према правилима Spark SQL програмског модела, а релационе операције на DataFrame (специјализованим системом оквира података) коришћењем специфичног језика домена (domain – specific – language → DSL). Подаци су складиштени на екстерним складишним системима који подржавају нове типове података. Имплементација алгоритама је изведена у програмском језику Scala. Испитивање модела за управљање великим серијама просторно-временских података извршена је над узорком који обухватити различите типове геопросторних података. Пошто је предмет истраживања управљање великим серијама просторно-временских типова података, узорак је дефинисан тако да количина података по испитиваном скупу буде реда гигабајта.

## 3. АРХИТЕКТУРА СИСТЕМА

Дефинисањем нових просторно-временских типова података, на одговарајући начин уклопљена у концепте Big data, архитектура система треба да постави у своју структуру стандарде за све системе податка. Интеграција података је обезбијеђена у оквиру постојећег система података дефинисане библиотеке као и захтјева структуре од стране одговарајућег апликативног софтвера. Архитектура ових података адресира податке у складиштима и податке у процесуирању гдје описује складишта података, групе података и њихове атрибуте, картирање карактеристика података, апликације које их користе и локације на које се смјештају подаци. То обезбјеђује критеријум за операције процесуирања података да би се омогућило креирање токова података и контрола тока података у систему.

Платформа која је коришћена за рад Apache Spark SQL модула је виртуелна машина (VMware Workstation) верзија 6.0 која је радила у Stand-alone VM моду. Модул који је коришћен за развој новог просторно-временског типа података је био Apache Spark SQL, верзија Apache Spark-а је 1.4.1 у Scala програмском језику, верзије 10.5.0. Креирање нових типова података вршено је у складу са стандардима ISO 19107, ISO 19111 и ISO 19115. Процесуирање у Apache Spark SQL вршено је кроз DataFrames с одговарајућим форматима података. Као улазни и излазни формат коришћен је GeoJSON због његових предности за процесуирање векторских података, у овом случају података К.О. Бања Лука 6 који су послужили као тест примјер за развијени модел и библиотеку са новим типом података, као и методама које подржава. За визуелизацију излазног формата послужила је open source платформа софтвера QGIS гдје је извршена крајња верификација функционалности креираног модела.

### 3.1. ЛОГИЧКИ МОДЕЛ ПОДАТАКА

Већина постојећих система за рад са геопросторним подацима заснива се на сличном логичком моделу података. DataFrame може бити конструисан од табела у системском каталогу (базиран на екстерним изворима података) или од постојећих RDD-ова изворних Java/Python објеката. Једном конструисани могу се подешавати са различитим релационим операторима, као што су Where и GroupBy, који преузимају изразе (expressions) из DSL-а, слично оквирима података у R-у и Python-у. Сваки DataFrame може бити прегледан и као RDD објеката редова. Омогућавају корисницима да позову процедурални Spark API као што је *map*. Супротно стандардним оквирима података апликације, Spark DataFrames су спори у смислу да сваки DataFrame објекат представља логички план да процесуира сет података, али не извршава се док корисник не позове специфичну „output операцију“, као што је *сачувај* или да позове процедуралну Spark API као што је *map*. На овај начин пружена је богата оптимизација кроз све операције гдје је коришћен Spark да изгради DataFrame. У коду апликације бр. парцела и парцеле су DataFrame-ови. Дио парцела је израз у DSL оквиру података, који је заробљен као абстракт синтаксне структуре ради репрензетовања Scala функције као стандардне Spark апликације. Сваки DataFrame једноставно представља логички план (као нпр. прочитај посједнике табеле и филтрирај за површине мање од 4000 јединица). Када корисник позове опцију *count*, која је излазна операција, Spark SQL гради физички план да израчуна коначни резултат. Ово може укључивати оптимизације као што су само *скенирај* „површина“ колону података ако је њихов складишни формат у форми колоне или чак користити индексирање у извору података да се израчунају исти редови. Модел Spark SQL користи угнијеждени модел података и подржава све важније SQL типове података, укључујући: *boolean*, *integer*, *double*, *decimal*, *string*, *date* и *timestamp*, као и комплексне (нпр. неатомске) типове податка: *structs*, *arrays*, *maps* и *union*. Насупрот многим стандардним DBMS-овима, Spark SQL обезбјеђује подршку за комплексне типове података у упитном језику и апликацијама. Важно је напоменути да Spark SQL, такође подржава кориснички дефинисане типове. Користећи овај тип система, постоји могућност да се прецизно моделују подаци из различитог скупа извора и формата, укључујући Hive, релационе базе података, JSON итд. Модел Spark SQL извора података је погодна карактеристика која омогућава корисницима да једноставно повежу њихове податке складиштене у различитим форматима и системима са Spark SQL-ом. Опремљен са изворима података апликације и SQL-ом, корисници могу да почну манипулисање подацима са минималним подешавањима и конфигурацијом. Извори података апликације и унифицирани чувај/учитај интерфејси изграђени на њима, поједностављују процес чувања/учитавања података у и из различитих извора.

### 3.2. ФИЗИЧКИ МОДЕЛ ПОДАТАКА

Архитектура нових типова комплексних података, се реализује кроз стандардне процесе физичке архитектуре система реализацијом механизма података за специфичне типове функционалности. За те потребе као могућа апликација Apache Accumulo садржи значајна побољшања на BigTable дизајну у форми ћелија-базиране приступне контроле и механизма серверског програмирања који може модификовати кључ/вриједности различитих тачака у процесу управљања просторно-временским подацима. На овај начин врши се имплементација стандардних интерфејса геодата да обезбиједи геопросторну функционалност на великом сету података развојних апликација. Промјенљива база података за смјештање великих количина геопросторних података је Hadoop



### 3.3. УЛАЗ/ИЗЛАЗ

Дефинисани модел Apache Spark SQL има могућност процесуирања различитих типова и формата података. Као тестни примјер коришћено је подручје КО Бања Лука 6, површине 5564171 m<sup>2</sup>. Као формат који служи за приказивање геометрије и топологије у датом моделу, коришћен је GeoJSON, у коме су подешени сви одговарајући параметри Гаус-Кругерове пројекције. За потребе превођења из векторског .shp фајла у одговарајући GeoJSON фајл коришћена је OGR/GDAL библиотека у којој је и дефинисана функција трансформације у овај формат. За извршавање ове трансформације неопходна је компатибилна библиотека ogr2ogr која је интегрисана у оквиру QGIS 2.10.1. На овај начин су одговарајуће припремљени подаци о парцелама уčitани у DataFrame-ове дефинисаних типова и Magellan библиотеку. Као основни оквир који се користи за проток податка је DataFrame, дефинисањем Context-а имплицитне функције за учитавање података (Слика 1.). На овај начин је обезбјеђено да Apache Spark омогући процесуирање и вршење операција над учитаним подацима. У излазу креираног модела поново је коришћен GeoJSON за формат излазних података над којима је извршен упит и који је касније коришћен за визуелизацију података. Током компајлирања је праћено кроз код у оквиру DataFrame-а да ли се у току процесуирања јављају одређени прекиди у моделу.

```
import sqlCtx.implicits._
var path = "parc"
val dfParc = sqlCtx.read.format("magellan").option("type",
    "geojson").load(path)
```

Слика 1. Имплицитно дефинисана функција за учитавање података

### 3.4. КОРИСНИЧКИ ДЕФИНИСАНЕ ФУНКЦИЈЕ

Једна од карактеристика која је омогућена у напредним аналитичким обрадама у Spark SQL су кориснички дефинисани типови. Кориснички дефинисани типови омогућавају кориснику да креира сопствене класе које су интероперабилније са SparkSQL-ом. Креирањем кориснички дефинисаног типа за класу X пружа се *могућност* креирања DataFrame-а који има класу X у шеми.

```
class Point(val x: Double, val y: Double, val z: Double, val t: Double)
extends Shape {
  def this() {this(0.0, 0.0, 0.0, 0.0)}
  override private[magellan] val delegate = {
    val p = new ESRIPoint()
    p.setX(x)
    p.setY(y)
    p.setZ(z)
    p.setT(t)
    p}
}
```

Слика 2. Дефинисање функције Point

Да би SparkSQL препознао кориснички дефинисане типове, ти кориснички дефинисани типови морају бити убиљежени са SQL кориснички дефинисаним типовима. Као примјер који је разрађен у оквиру развојног дијела можемо узети начин регистравања тродимензионалне тачке проширене са временском димензијом (x,y,z,t) као кориснички дефинисан тип. Такве векторе представљају тип DOUBLE values (Слика 2.). После регистрације овог типа, тачка ће бити препозната као изворни објекат који Spark SQL конвертује у DataFrame и биће усвојен у кориснички дефинисаним функцијама на Тачкама.

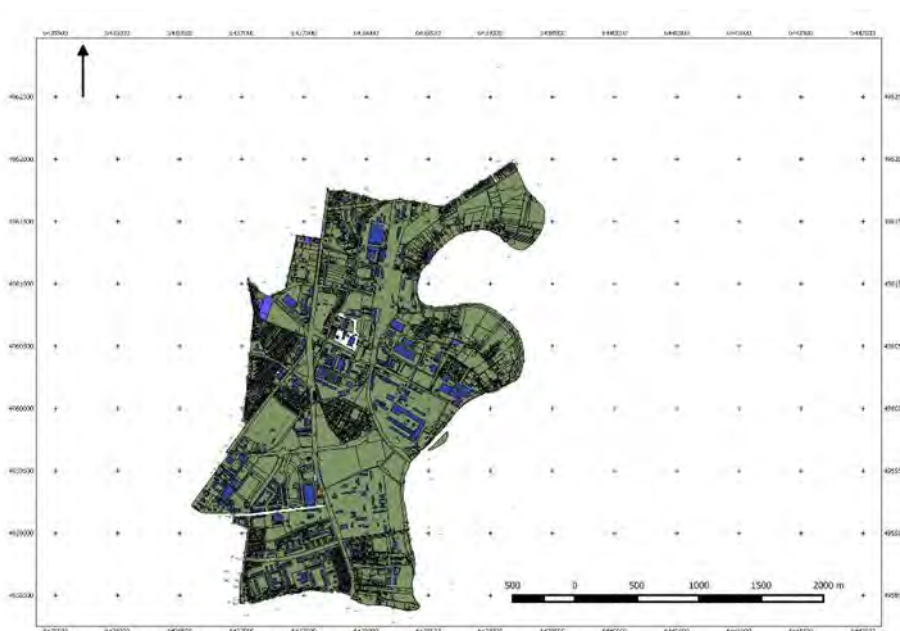
### 3.5. ОПЕРАЦИЈЕ

Оператори над просторно-временским типовима података који служе за извршавање упита, дефинисани су посредством специфичног језика домена и на тај начин је омогућено недвосмислено и поједностављено дефинисање објеката који су искоришћени у писању апликације. Примјењене су просторне операције Within и Intersects и формиран је нови DataFrame гдје су парцелама придружени одговарајући бројеви. Рађен је просторни спој између парцела које су тачке и парцела које су полигони (spatial join). Коришћена је операција која комбинује два или више сетова података за просторне исказе. У дијелу гдје је вршено спајање непосторних типова података, спајајући атрибути су истог типа и обично је сваки просторни атрибут описан минималним граничним правоугаоником (МГП). Просторно спајање одвија се кроз два корака: кораком филтрирања и кораком пречишћавања (Слика 3.).

```
var parcBrParc = dfBrParc.select($"point", $"metadata"("TxtMemo")).
join(dfParc.select
($"polygon")).where(dfBrParc.col("point") intersects
dfParc.col("polygon"))
var polygonJson = "{\"type\": \"FeatureCollection\", \"features\": [ ";
parcBrParc.collect.foreach(row -> {
    polygonJson += "{ \"type\": \"Feature\", \"properties\": { \"Id\":
    \"%s
    \", \"format(row.getString(1))
    polygonJson += \" \"geometry\": { \"type\": \"Polygon\",
    \"coordinates\":
    [ [ \"
        val polygon = row.getAs[Polygon]("polygon")
        polygon.points.foreach { point -> {
            polygonJson += "[ %f, %f, %f,
            %f], \"format(point.x, point.y, point.z,
            point.t)}}
        polygonJson = polygonJson.substring(0, polygonJson.length() - 1)
        polygonJson += " ] ] } \n},";
```

Слика 3. Просторни упит над просторно-временским подацима





Слика 4. Визуелизовани приказ К.О. Бања Лука 6 након извршеног упита

У кораку филтрирања, низови МГП-а преклапају се са подручјем упита које је установљено. У њему приликом извршавања се установљава да ли се два правоугаоника сијеку. Они правоугаоници који прођу корак филтрирања иду до корака пречишћавања, гдје је коришћено тачно просторно представљање. Просторни исказ је обиљежен на овим просторним представама. У овом кораку број процесуираних редова је мањи него у претходном. У алгоритму угниједених функција сваки ред је скениран, сваки пар редова и они који задовољавају исказ просторног спајања се додају крајњем резултату у QGIS апликацији (Слика 4.).

### 3.6. СПЕЦИФИЧНИ ЈЕЗИК ДОМЕНА

Разлика између језика генералне намјене и специфичног језика домена није увијек огромна, јер програмски језици имају специјализоване карактеристике за партикуларне домене који су потпуно примјенљиви. Такође, могу бити способни да прошире апликацију, гдје у пракси примарно користе специфични домен. Систем који се користи је модуларан и флексибилан и подржава вишеструке алгоритме дизајниране парадигме као што је MapReduce, in-memory computing или agent-based programming. Садржи web базиран кориснички интерфејс гдје корисници домена (нпр. ГИС аналитичари) могу дефинисати висок ниво процесуирања радног тока коришћењем специфичног језика домена (DSL) [13] и [14]. Креирањем DSL-а омогућено је рјешавање дијелимичних типова проблема много јасније него што је изворни језик пружао. На овај начин је декларисана синтакса и граматика за специфичне циљеве у дизајну и имплементацији модела и метода над просторно-временским подацима гдје се дефинише специфична синтакса која проналази и мијења стандардне изразе (Слика 1.). Ови мали језици (скриптни језици) су искоришћени у оквиру ћелије да обаве комплексне задатке код којих су имплицитно

дефинисане функције и оператори над геометријским примитивима који су коришћени приликом дефинисања сваког појединачног примитива и метода над њима (Слика 5.). Коришћењем овог приступа дефинисане су функције `Within` и `Intersects` у колонама израза и облика. Коришћењем ове методе не компајлира се `byte` код или извршни код, већ низ разних објеката података облака тачака. У овом случају, дефинисан је начин приступања и управљања `HBase` табелама уместо стандардних релационих база података.

```
package object dsl {
  trait ImplicitOperators {
    def intersection(other: Expression): Expression =
      Intersection(expr, other)
    def intersection(other: Shape): Column = Column(Intersection(expr,
      ShapeLiteral(other)))
    def transform(fn: Point -> Point) = Transformer(expr, fn)}
}
```

Слика 5. Дефинисање функције `Intersects` у оквиру модела Домен специфичног језика

#### 4. ЗАКЉУЧАК

У раду је описано иницијално истраживање о могућностима дефинисања и обраде просторно-временских типова података у `Big Data` парадигми коришћењем `Apache Spark` платформе. Претходне студије, као и разни случајеви коришћења су показали да `Apache Spark` представља добру основу за ефикасно управљање апликацијама које обрађују велике количине података на кластеру рачунара.

`Spark` Модел са декларативним `SQL DataFrame` API омогућава релационо процесуирање геопросторних података. Нови типови података су засновани на кориснички дефинисаним функцијама и домен специфичном језику који је обезбиједио могућност повезивања релационих класа и метода. На основу предложеног модела и тестног примјера, приказана је егзактна веза у примјени концепата пословне интелигенције и управљања просторно-временским типовима података у `Big Data` парадигми. Доказани услови су искористиви као нова и савремена метода и платформа на чијим основним принципима може бити развијен софистициран софтвер за управљање великим количинама не само геопросторних података већ и `streaming` података, а над којима се могу извршавати дјелотворни, брзи и сложени аналитички упити.

Ово истраживање у коме су развијени основни вишедимензионални типови представља модел податка који комбиновано са `building information model`-ом даје платформу за креирање модела простора. Такав модел може послужити као одговарајућа стандардизована подлога за израду комплетног система вишедимензионалног катастра.

## 5. БИБЛИОГРАФИЈА

- [1] Ambrust, M., Xin, R. S., Lian, C., Huai, Y., Davies, L., Bradley, J. K., Mneg, X., Kaftan, T., Franklin, M. J., Ghodsi, A., Zaharia, M., (31.5.-04.06.2015) „SPARK SQL: Relational Data Processing in Spark“, SIGMOD'15, Australia, Victoria, Melbourne, DOI: 10.1145/2723372.2742797.
- [2] Dean, J., Ghemawat, S., (2004) “MapReduce: Simplified Data Processing on Large Clusters”, OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco.
- [3] Zikopoulos, P. C., de Roos, D., Parasuraman, K., Deutsch, T., Corrigan, D., Giles, J., (2012) “Harness the Power of Big data”, McGraw Hill, USA.
- [4] INSPIRE Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007: Establishing an Infrastructure for Spatial Information in the European Community, Official Journal of the European Union.
- [5] Karau, H., Konwinski, A., Wendell, P., Zaharia, M., (2015) “Learning SPARK”, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA, ISBN: 9781449358624.
- [6] Russell, M. A., (2014) “Mining the Social Web, Data mining Facebook, Twitter, LinkedIn, Google+, Github, and more”, USA, O'Reilly.
- [7] Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J., Welton., C., (2009) “MAD skills: new analysis practices for Big data”, VLDB.
- [8] Xin, R.S., Rosen, J., Zaharia, M., Franklin, M.J., Shenker, S., Stoica, I., (2013) “Shark: SQL and rich analytics at scale”, In SIGMOD.
- [9] Ward, J. S., Barker, A., (2013) “Undefined By Data: A Survey of Big data Definitions”, School of Computer Science University of St Andrews, UK.
- [10] Winslett, M., A. Cary, Z. Sun, V. Hristidis, and N. Rishe (2009), Experiences on Processing Spatial Data with MapReduce, in Scientific and Statistical Database Management, edited, pp. 302-319, Springer Berlin / Heidelberg.
- [11] Pajić, V., Govedarica, M., Galić, Z., Alargić, I. (2015), Point cloud processing on Apache Spark, YU INFO 2015, Beograd, ISBN: 978-86-85525-15-5.
- [12] Krämer, M., Senner, I., (2015) “A modular software architecture for processing of big geospatial data in the cloud”, Computers & Graphics, Vol 49, pp 69–81.
- [13] Giachetta, R. (2015) “A framework for processing large scale geospatial and remote sensing data in MapReduce environment”, Computers & Graphics, Vol 49, pp 37–46.
- [14] Liu, K., Boehm, J. (2015) “CLASSIFICATION OF BIG POINT CLOUD DATA USING CLOUD COMPUTING”, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-3/W3, ISPRS Geospatial Week 2015, La Grande Motte, France, doi:10.5194/isprsarchives-XL-3-W3-553-2015.