

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra softwarového inženýrství
Obor: Aplikace softwarového inženýrství



Porovnání účinnosti komprese dat ve formátech XML a JSON

Comparison of the effectiveness of data compression in XML and JSON format

DIPLOMOVÁ PRÁCE

Vypracoval: Bc. Tomáš Smola
Vedoucí práce: Ing. Tomáš Liška, Ph.D.
Rok: 2015

Před svázáním místo téhle stránky

vložte zadání práce

 s podpisem děkana (bude to jediný oboustranný list ve Vaší práci) !!!!

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

.....
Bc. Tomáš Smola

Poděkování

Děkuji Ing. Tomáši Liškovi, Ph.D. za vedení mé diplomové práce a za podnětné návrhy, které ho obohatily.

Bc. Tomáš Smola

Název práce:

Porovnání účinnosti komprese dat ve formátech XML a JSON

Autor: Bc. Tomáš Smola

Obor: Aplikace softwarového inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Ing. Tomáš Liška, Ph.D.

Katedra softwarového inženýrství, Fakulta jaderná a fyzikálně
inženýrská, České vysoké učení technické v Praze

Konzultant: —

Abstrakt: Abstrakt

Klíčová slova: Klíčová slova

Title:

Comparison of the effectiveness of data compression in XML and JSON format

Author: Bc. Tomáš Smola

Abstract: Abstract

Key words: Key words

Obsah

Úvod	1
1 Obecné seznámení s formáty XML a JSON	2
1.1 XML	2
1.1.1 Charakteristika	2
1.1.2 Syntaktická analýza	2
1.1.3 Parsování	3
1.1.4 Výhody a nevýhody	3
1.2 JSON	3
1.2.1 Charakteristika	4
1.2.2 Syntaktická analýza	4
1.2.3 Parsování	5
1.2.4 Výhody a nevýhody	5
2 Komprese dat	7
2.1 Princip komprese dat	7
2.2 Typy kompresních metod	7
2.3 Charakteristika komprese	8
2.4 Kompresní algoritmy pro data v obecném formátu	8
3 Popis existujících kompresních algoritmů	9
4 Přehled existujících implementací kompresních algoritmů pro efektivní uchovávání dat ve formátu XML a JSON	10
5 Vlastní implementace vybraných kompresních algoritmů	11
6 Porovnání účinnosti komprese dat ve formátu XML a JSON	12

Závěr	13
Seznam použitých zdrojů	14
Přílohy	15

Úvod

Kapitola 1

Obečné seznámení s formáty XML a JSON

V této kapitole seznámím čtenáře se značkovacím jazykem XML a následně s JSONem, formátem pro výměnu dat. Mým cílem je popsat základní charakteristiky a syntaxi obou formátů tak, abych byl já, a následně i čtenář, schopen pochopit v kapitole 4 principy a výhody vybraných algoritmů využívajících znalosti struktury datových souborů.

1.1 XML

Na základě značkovacího jazyka SGML (Standard Generalized Markup Language), jehož obecnost činí úplnou implementaci velmi náročnou, vznikl vybráním nejpoužívanějších možností nový značkovací jazyk XML (eXtensible Markup Language), je tedy podmnožinou jazyka SGML. XML je obecný a otevřený, jeho vývoj a standardizaci realizovalo konsorciem W3C (World Wide Web Consortium) [2]. XML umožňuje snadné vytváření konkrétních značkovacích jazyků pro popis dokumentů a dat ve standardizované, textově orientované podobě.

1.1.1 Charakteristika

1.1.2 Syntaktická analýza

V podstatě jde o textový dokument, je tvořen posloupností Unicode¹ znaků, ve kterém se rozlišují dva základní prvky: elementy neboli značky a obsah. Při práci s XML je nutné mít na paměti, že je na dodržení syntaxe kladen velmi velký důraz. Při dodržení správného způsobu zápisu a pravidel, která budou popsána níže, lze dokument považovat za tzv. *well-formed XML* [2].

¹Unicode je standard pro konzistentní kódování, reprezentaci a manipulaci znaků většiny světových abeced.

Element

Základním prvkem každého XML dokumentu je element, který je vyznačen pomocí takzvaných tagů², mezi které může být vložen obsah. Počáteční i ukončující tag je dle definice [2] složen z dvojice znamének < (menší než) a > (větší než), mezi kterými je zapsán název tagu a volitelně i atributy. Ukončovací tag má navíc před svým názvem znak / (lomeno). Při správné aplikaci pravidel může vypadat element například následujícím způsobem:

```
<název_elementu název_atributu="hodnota atributu"></název_elementu>.
```

V případě, že element neobsahuje žádný obsah, lze ho zkráceně zapsat jako tzv. prázdný element:

```
<název_prázdného_elementu />.
```

V případě nedodržení správné syntaxe může nastat problém při rozpoznávání zapsaných dat, což může mít za následek nekompatibilitu mezi různými systémy při výměně dat.

Atribut

Počáteční tag elementu může obsahovat atributy upřesňující jeho význam. Atribut je vždy složen ze svého názvu a hodnoty, které jsou odděleny znakem = (rovná se). Hodnota je navíc zapsána mezi dvojicí znaků " (uvozovky) nebo ' (apostrof), přičemž hodnota může obsahovat jeden z těchto znaků tak, že se syntakticky nekříží. Následuje příklad atributu, jehož hodnota obsahuje znak ':

```
název_atributu="hodnota atributu obsahující znak ' (apostrof)".
```

Obsah

Vše, co není tagem, je v dokumentu považováno za obsah. Kromě obvyčejného textu mohou být obsahem další vnořené elementy, komentáře, instrukce pro zpracování, reference a další. Vzhledem k tomu, že určité znaky mají v syntaxi XML speciální význam (např. <, >), využívají se pro jejich zápis znakové entity³. Úplný výčet toho, co může XML dokument obsahovat, je včetně pravidel definován v [2].

1.1.3 Parsování

1.1.4 Výhody a nevýhody

1.2 JSON

JSON neboli JavaScript Object Notation je odlehčený způsob zápisu (formátování) dat. Tento textový formát je nezávislý na počítačové platformě a je čitelný pro člověka. JSON je založen na dvou univerzálních datových strukturách: kolekce dvojic klíč/hodnota a seřazený seznam hodnot, které podporují v nějaké formě asi všechny známé moderní programovací jazyky. Díky těmto vlastnostem se JSON stal velmi oblíbeným formátem pro vzájemnou výměnu dat.

²Tag definuje formu části textu.

³Pomocí znakových entit (sekvence znaků) lze zapsat znaky, které neobsahuje zvolené znaková sada, nebo mají v použitém kontextu speciální význam.

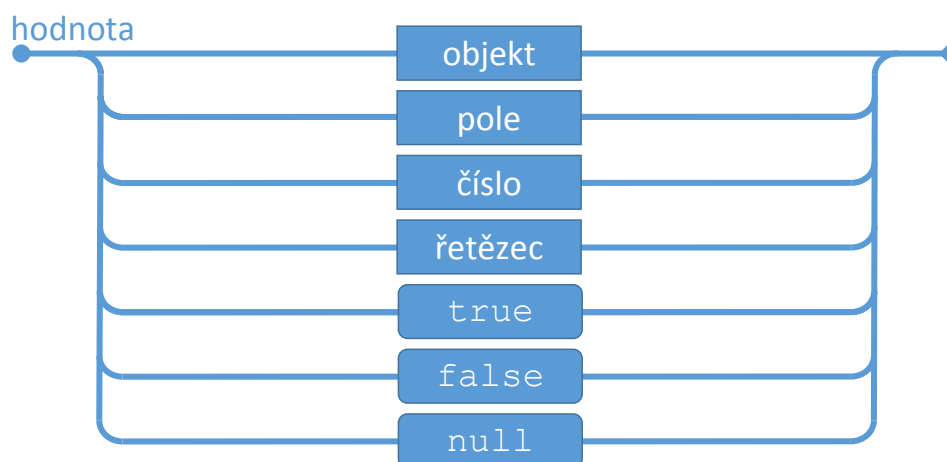
1.2.1 Charakteristika

1.2.2 Syntaktická analýza

Jak již bylo zmíněno, je JSON textový formát a je tedy posloupností tokenů tvořených z Unicode znaků. Sada tokenů obsahuje šest strukturálních tokenů: [(levá hranatá závorka), { (levá složená závorka),] (pravá hranatá závorka), } (pravá složená závorka), : (dvojtečka) a , (čárka); dále obsahuje znakové řetězce, čísla a tři doslovné tokeny: `true`, `false` a `null`.

Hodnoty

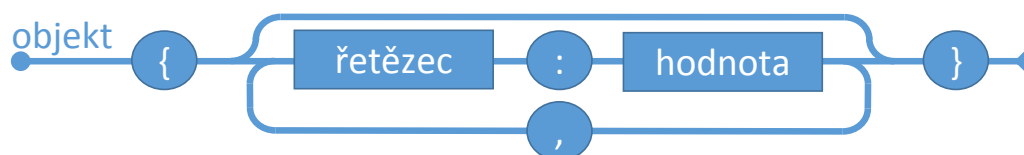
Za hodnotu je v JSONu považován objekt, pole, číslo, řetězec, `true`, `false`, nebo `null`.



Obrázek 1.1: Struktura hodnoty

Objekty

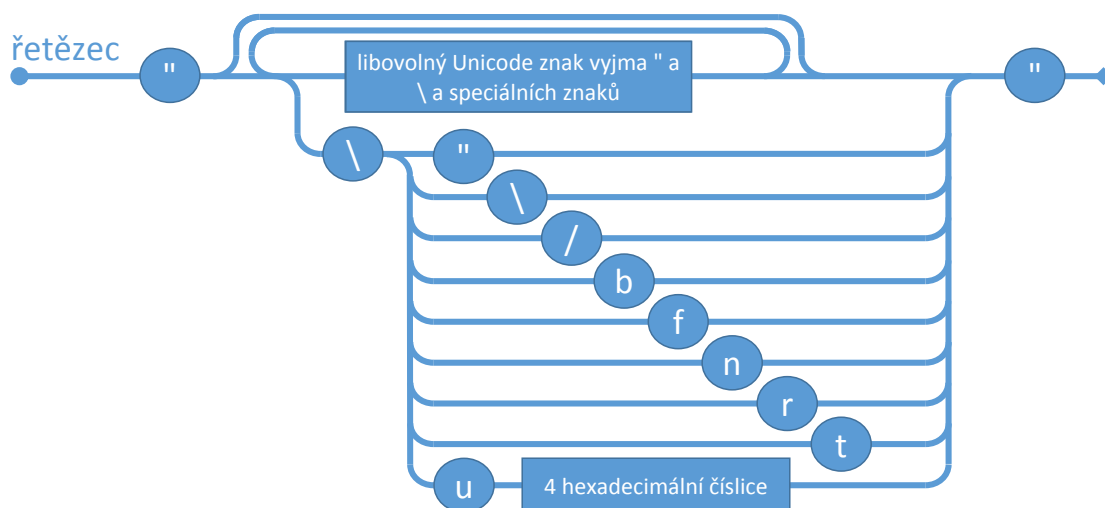
Objekt je reprezentován dvojicí složených závorek, uvnitř kterých je žádná nebo více dvojic klíč/hodnota, přičemž klíč je řetězec. Klíč a hodnota jsou odděleny dvojtečkou a jednotlivé dvojice odděluje čárka.



Obrázek 1.2: Struktura objektu

Pole

Pole je složeno z dvojice hranatých závorek, mezi kterými může být nula nebo více seřazených hodnot, které jsou odděleny čárkou.



Obrázek 1.5: Struktura řetězce

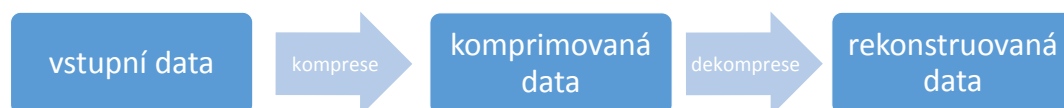
Kapitola 2

Komprese dat

Komprese nebo také komprimace dat je taková transformace dat, která má za cíl úsporu zdrojů při ukládání nebo archivaci a nebo snížení datového toku při přenosu, to vše při současném zachování informace obsažené v datech. Jinými slovy jde o redukci velikosti datových souborů, jehož následkem je úspora paměťových či přenosových kapacit. Postup, při kterém z komprimovaných dat rekonstruujeme data originální, se nazývá dekomprese.

2.1 Princip komprese dat

Data velmi často obsahují tzv. redundantní¹ informaci, toho právě využívá komprese – data jsou zpracována tak, aby byla redundance minimalizována. Jak lze vidět na obrázku 2.1, je na vstupní data použita operace komprese. Operací dekomprese dostaneme poté data rekonstruovaná – v závislosti na použité kompresní metodě, respektive na požadavcích získáme buď data přesně odpovídající původním a nebo pouze částečná. Z tohoto hlediska rozlišujeme dva typy kompresních metod: ztrátové a bezztrátové.



Obrázek 2.1: Princip komprese

2.2 Typy kompresních metod

Jak název napovídá, při ztrátové kompresi ztratíme část informace obsaženou v původních datech, respektive jsou původní data pouze aproximována. Toto nám nemusí vadit například u obrázků, zvuku a videí, kde je využito nedokonalosti lidských smyslů. Lidské ucho nedokáže například slyšet velmi vysoké frekvence. Má smysl v datech určených k poslechu zachovávat informaci, kterou nemůže člověk slyšet? Častá odpověď je „ne“. Tohoto principu využívá mnoho kompresních metod, například známý zvukový formát MP3. Odstraněním „nepotřebné“ informace z dat je dosaženo ještě větší redukce objemu.

¹Redundance znamená informační nadbytek, např. vícenásobný výskyt slov v textu.

Naopak v případě bezeztrátových metod je při kompresi zachována veškerá informace a při dekompresi jsou rekonstruována původní data. Těchto metod se využívá převážně tam, kde není možné původní data jakkoliv pozměnit. Například data ve formátech XML a JSON, kterým se věnuji v této práci, si nemůžeme dovolit pozměnit (přestanou mít původní význam), nebo dokonce ztratit.

2.3 Charakteristika komprese

Nejdůležitější charakteristikou je tzv. kompresní poměr, který vyjadřuje velikost komprimovaných dat vůči původním, lze ho zapsat následujícím vztahem:

$$\text{kompresní poměr} = \frac{\text{délka komprimovaných dat}}{\text{délka původních dat}}. \quad (2.1)$$

Další sledovanou charakteristikou je tzv. úspora místa, která je vyjádřena jako:

$$\text{úspora místa} = 1 - \text{kompresní poměr}^{-1}. \quad (2.2)$$

Při volbě vhodné metody nás také zajímají časy potřebné pro kompresi i dekompresi.

2.4 Kompresní algoritmy pro data v obecném formátu

Kapitola 3

Popis existujících kompresních algoritmů

Kapitola 4

Přehled existujících implementací kompresních algoritmů pro efektivní uchovávání dat ve formátu XML a JSON

Kapitola 5

Vlastní implementace vybraných kompresních algoritmů

Kapitola 6

Porovnání účinnosti komprese dat ve formátu XML a JSON

Závěr

Seznam použitých zdrojů

- [1] Standard ECMA-404. *The JSON Data Interchange Format*. Geneva: Ecma International, 2013, [online], [cit. 28. října 2014]. Dostupné na: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
- [2] W3C. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 26. listopadu 2008, [online], [cit. 26. listopadu 2014]. Dostupné na: <http://www.w3.org/TR/2008/REC-xml-20081126/>.

Přílohy