Performanceoptimierte WMS-Dienste mit QGIS Server

Dr. Marco Hugentobler, Sourcepole AG
Twitter: @sourcepole

Inhalt

- Einleitung
- Werkzeuge zur Diagnose
- Tips für den Server Administrator
- Benchmark WMS der amtlichen Vermessung
- Benchmark WMS LK25
- Weitere Pläne

Einleitung

- FOSS4G WMS Benchmarks 2010 und 2011
- Schlechtes Abschneiden QGIS Server
- Viel Diskussion über Performance auf der QGIS Mailingliste
- “The software stack used by QGIS is much fatter than the one used by Mapserver for instance”
- “...it would be really surprised to see it keeping up with mapserver or mapnik performances.”

Einleitung (2)

- **Performance (Leistungsfähigkeit):**
  - Zeit zwischen Absetzen der Anfrage und Anzeige auf der Clientseite (Yang et al. 2011)
  - Durchsatz pro Zeiteinheit (z.B. FOSS4G Benchmarks)

- **Schritte eines WMS Requests:**
  - Daten von Datenquelle lesen
  - Render des Kartenbildes (im Speicher)
  - Umwandeln des Bildes in ein Webformat (z.B. Png8/24/32, jpeg...)
  - Transfer des Bildes vom Server zum Client

Werkzeuge zur Analyse

- Vmstat : Ist Prozessor oder Disk der Bremsfaktor
- Firebug: Anteil Übertragungszeit / Serverzeit
- Valgrind: Profiler. In welchen Funktionen wird wieviel Zeit verbraucht
- Jmeter: Simulation von parallelen Anfragen, um die Skalierbarkeit zu testen

Tips für den Serveradmin (1)

- Anzahl Objekte mit massstabsabhängiger Symbolisierung begrenzen
- Räumliche Indices für Vektoren / Pyramiden für Raster
- Datenbankzugriff über Local socket anstatt TCP, wenn die DB auf der gleichen Maschine läuft
- Gestrichelte Linien / Polygonumrandungen sowie Labelbuffers weglassen wenn möglich
- Bei Rasterkarten jpeg anstatt png verwenden

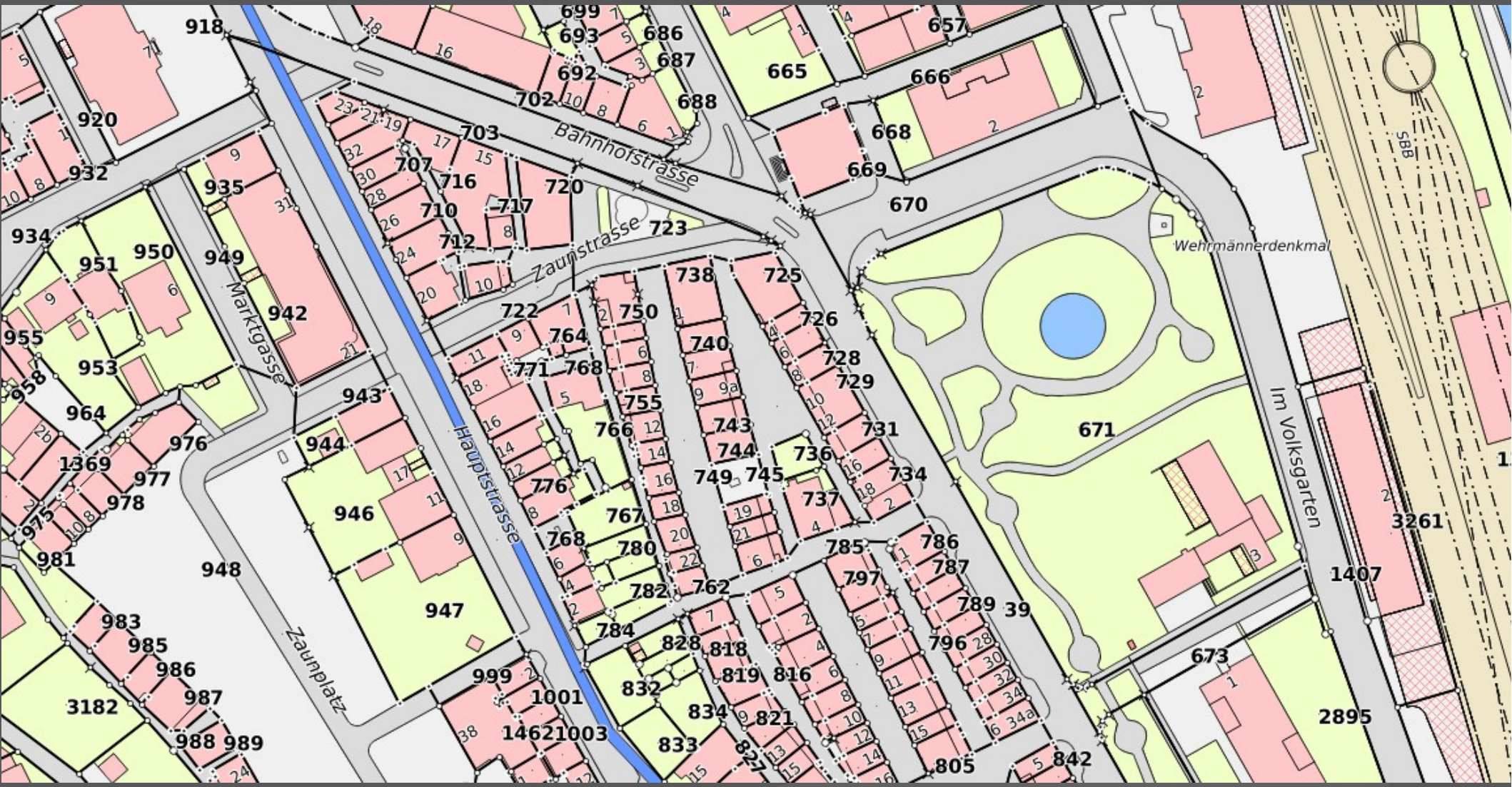
Tips für den Serveradmin (2)

- Für Vektoren png8 verwenden falls Qualität ok
- Anzahl Koordinatensysteme einschränken (schnelleres GetCapabilities)
- Mit Anzahl paralleler Serverinstanzen experimentieren
- Antwortzeit mit progressivem Rendern verkürzen
- Einen Reverse Proxy Cache (z.B. Varnish) vorschalten und Karte in Kacheln abfragen
- Filebasierter Tilecache

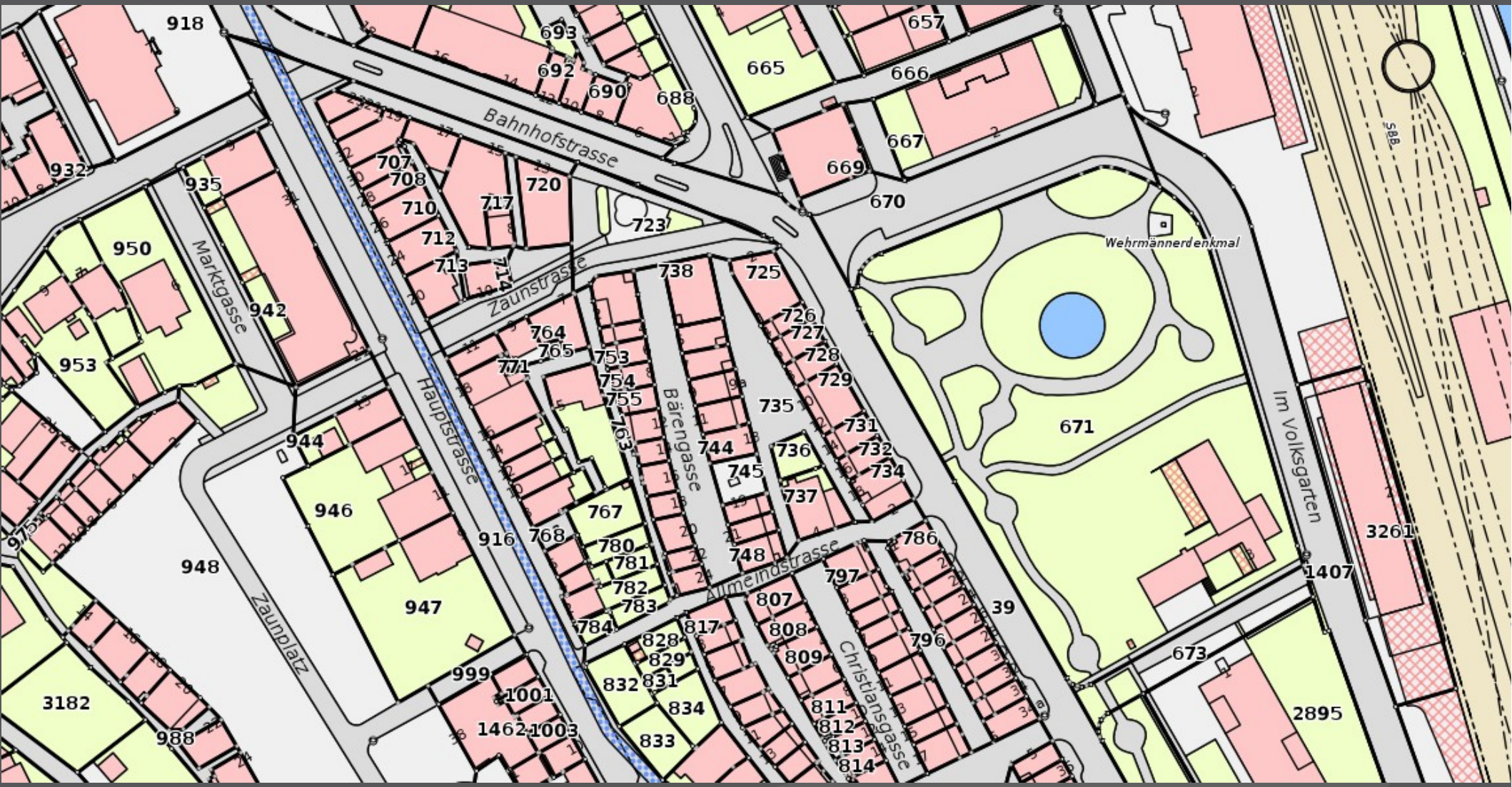
Benchmark AV WMS

- Jeder Kanton muss einen WMS der amtlichen Vermessung bereitstellen => der Benchmark ist höchst praxisrelevant
- Benchmark mit QGIS server (QGIS Enterprise) und mit UMN mapserver 6.2.1
- Beide Server / Konfigurationen im Praxiseinsatz
- Testdaten: Kanton Glarus, zufällige Ausschnitte 1:800 – 1:5'000
- Jmeter / WMS / DB auf privatem Laptop laufengelassen
- Ausgabeformat: png8

AV-WMS (QGIS Server)



AV WMS (UMN Mapserver)



Benchmark AV WMS (2)

UMN mapserver

N Requests	Count	Avg (ms)	Min (ms)	Max (ms)	Throughput (Req/s)
1	100	151	86	357	6.5
2	100	112	57	198	15.1
4	200	166	84	440	21.9
8	200	313	65	935	22.3
16	400	644	83	3232	22.8
32	384	965	60	12649	23.2
64	768	1901	100	28496	22.7

QGIS server

N Requests	Count	Avg (ms)	Min (ms)	Max (ms)	Throughput (Req/s)
1	100	128	52	712	7.7
2	100	101	34	457	16.4
4	200	130	41	744	27.4
8	200	246	44	2322	26.7
16	400	517	47	3960	25
32	384	761	46	9491	29.7
64	768	1474	63	25124	27.1

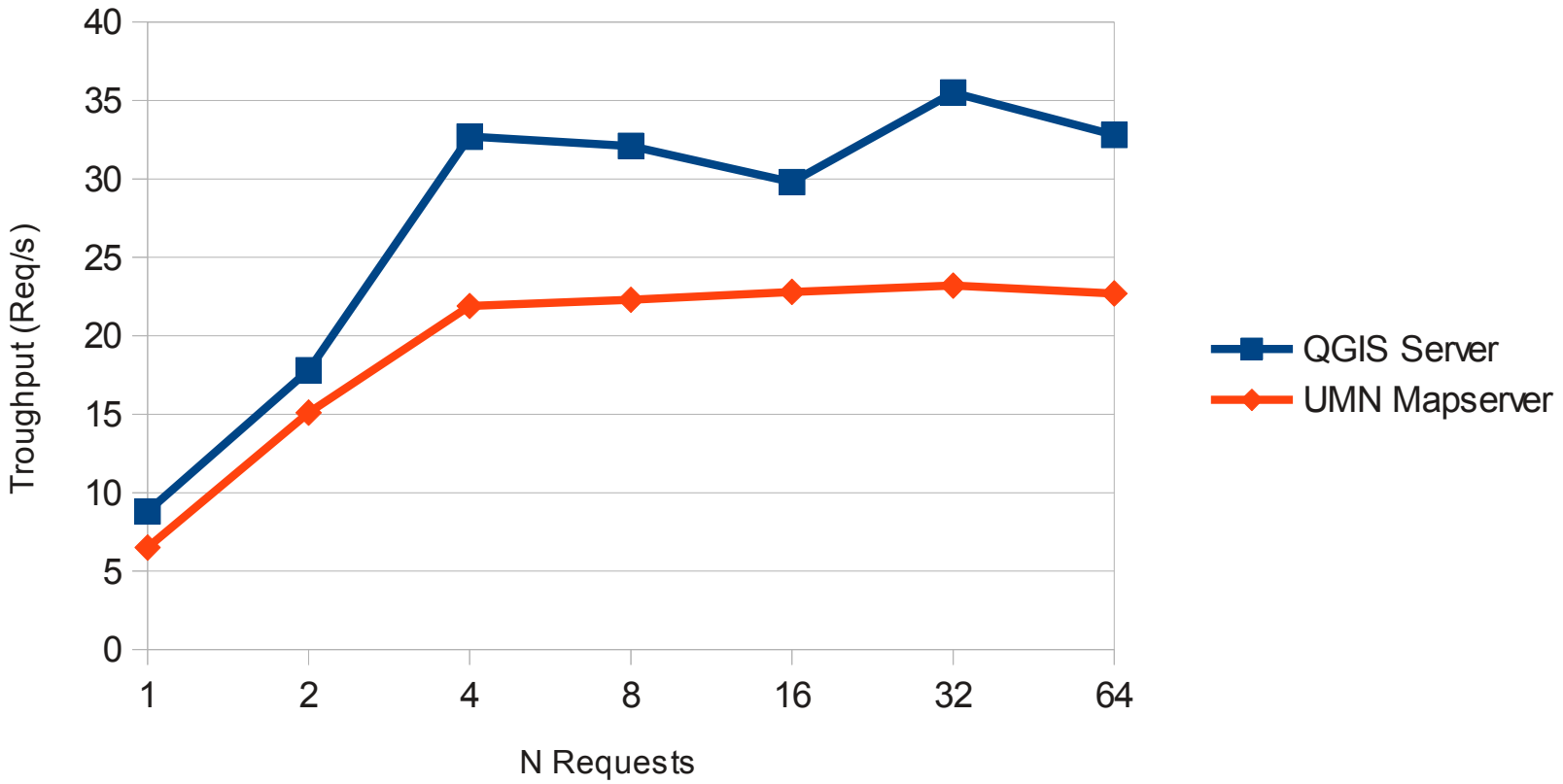
Benchmark AV WMS

- **Benchmark ist Prozessorlastig (Datenbankcache) => hohe Durchsatzzahlen**
- **QGIS server erreicht mehr Durchsatz, UMN kleinere maximale Antwortzeiten**
- **Variationen**
  - Ausgabeformat png24 anstatt png8: 27 / 23
  - Nur Bodenbedeckung (LCSFC): 54 / 55
  - Nur Punktsymbole (OSBP): 114 / 86
  - Nur Labels (SOOBJ,OSNR,LNNA,HADR,LOCPOS): 84 / 56

Benchmark AV WMS

- **Profilen eines QGIS Server Requests:**
  - 50% der Zeit für Umwandlung nach png8
  - Das Rendern braucht die anderen 50%
  - Rendern der Labels braucht 25%
  - Optimierung: Polygonkoordinate müssen ohne Umprojektion nicht durch proj4 geschleusst werden (4%)
  - Bei der Konvertierung nach png8 wird häufig QImage::pixel verwendet (5%). Besser: Zugriff über QImage::bits

Benchmark AV WMS: Resultate



Benchmark LK25

- Raster mit Palette (1 Band, 256 Farben)
- Zufällige Ausschnitte im Massstabsbereich 1:2'000 – 1:18'000
- Test mit Nearest neighbour resampling
- Test mit bilinear / average resampling
- Ausgabeformat: jpeg

Benchmark LK25 (nearest neighbour)

QGIS server

N req.	count	Avg	Min	Max	Throughput
1	100	45	8	93	21.2
2	100	39	6	91	40.5
4	200	51	7	112	59.2
8	200	82	9	193	70.4
16	400	180	6	1099	71.0
32	384	273	13	3254	72.4
64	768	590	21	9324	69.0

UMN mapserver

N req	count	Avg	Min	Max	Throughput
1	100	39	10	69	24.2
2	100	32	7	58	48.4
4	200	45	7	86	64.5
8	200	73	10	222	74.8
16	400	170	8	444	77.3
32	384	250	13	3331	77.5
64	768	525	11	8234	76.3

Benchmark LK25: ältere QGIS Versionen

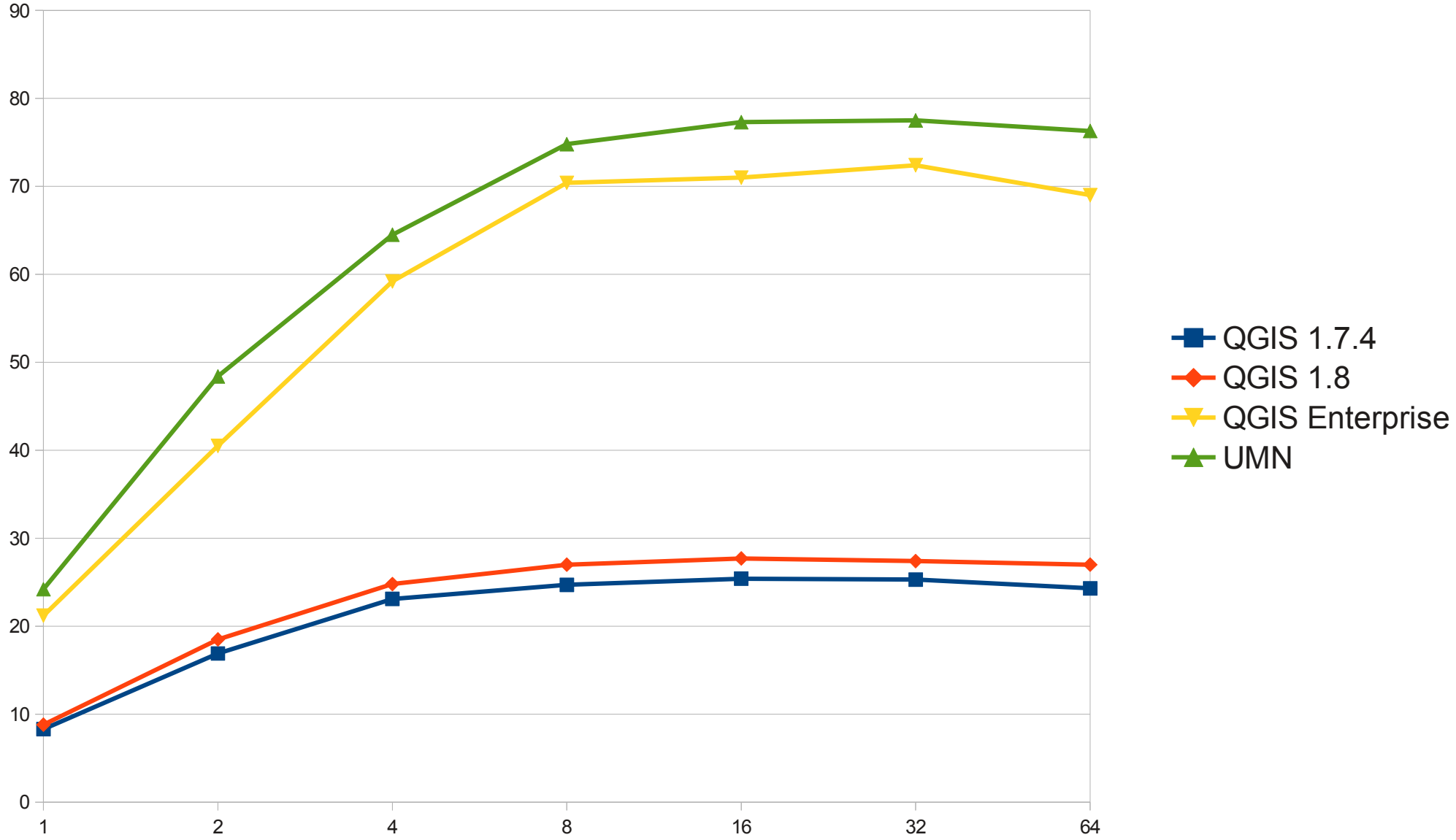
QGIS server 1.7.4

N req	count	avg	min	max	throughput
1	100	118	16	190	8.3
2	100	109	14	219	16.9
4	200	160	22	267	23.1
8	200	283	29	594	24.7
16	400	586	24	1661	25.4
32	384	879	25	10780	25.3
64	768	1689	51	28690	24.3

QGIS server 1.8

N req	count	avg	min	max	throughput
1	100	111	10	174	8.8
2	100	100	9	189	18.5
4	200	143	15	249	24.8
8	200	253	23	504	27.0
16	400	528	15	1810	27.7
32	384	792	19	8600	27.4
64	768	1532	17	24470	27.0

Benchmark LK25 nearest neighbour



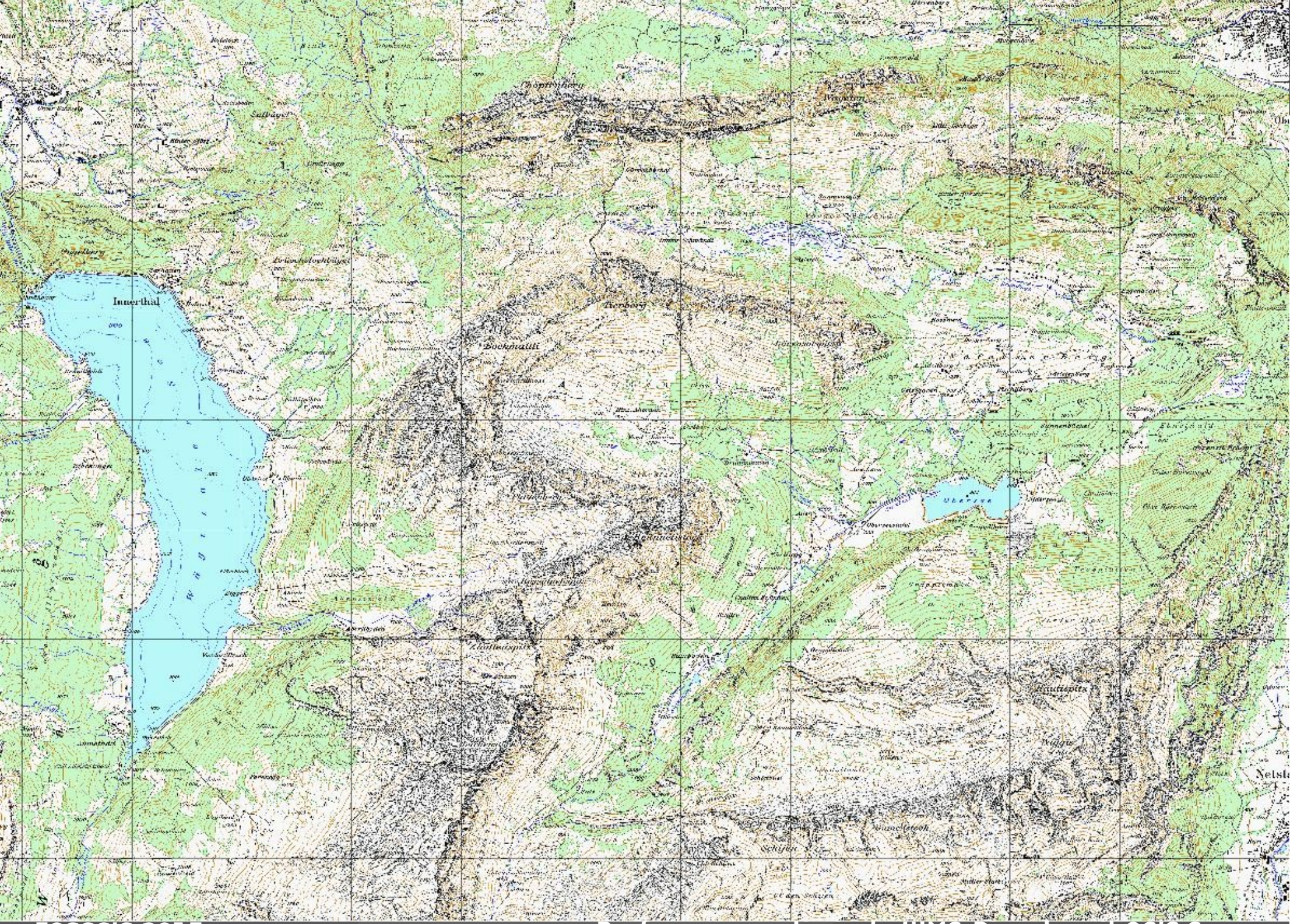
Benchmark LK25 Average resampling (mit Faktor 4)

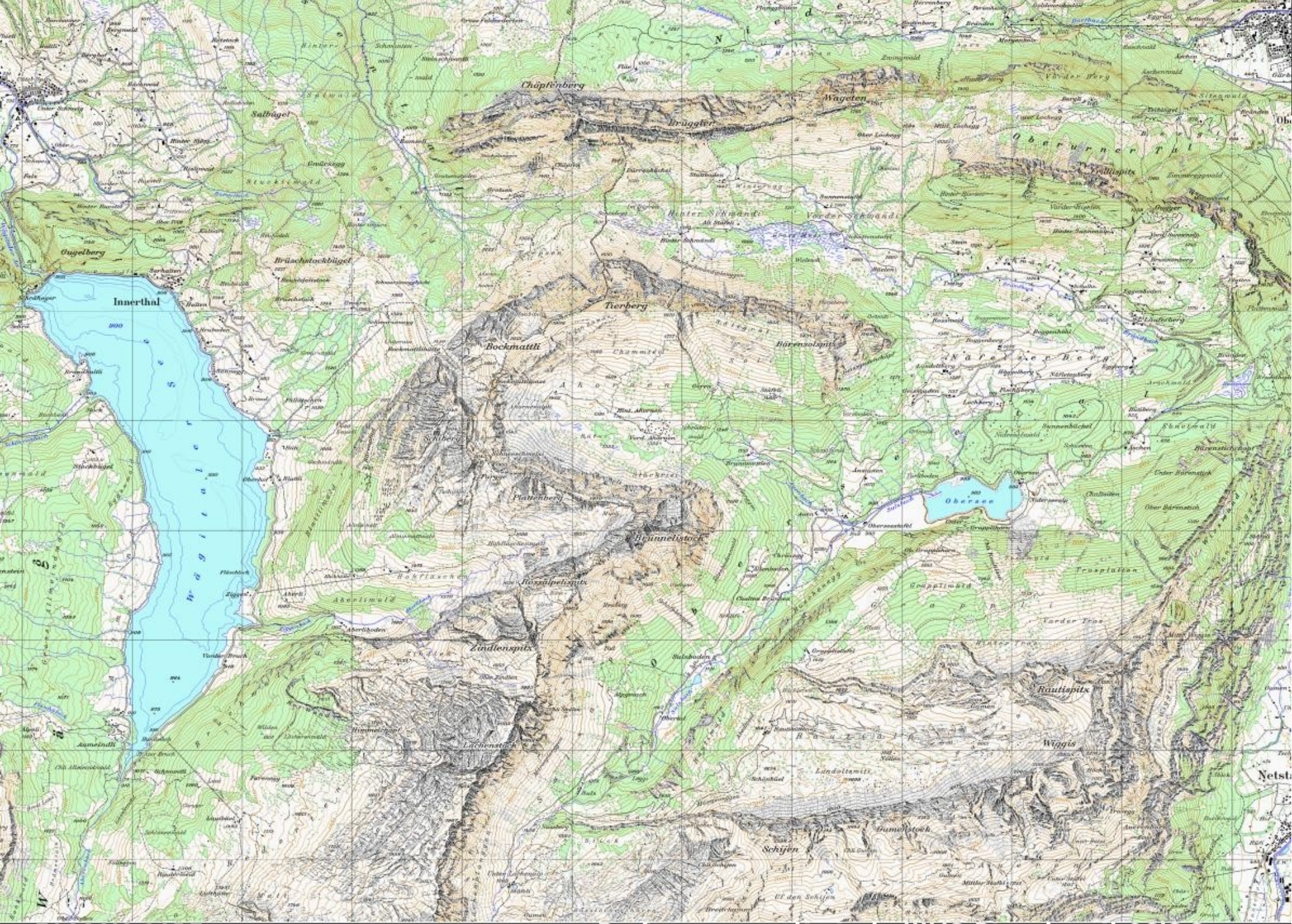
QGIS server

N req	count	avg	min	max	throughput
1	100	118	18	548	8.3
2	100	83	7	541	22.1
4	200	146	7	946	22.6
8	200	216	11	1550	28.5
16	400	453	13	3167	28.9
32	384	743	11	9354	29.3
64	768	1736	8	28293	23.0

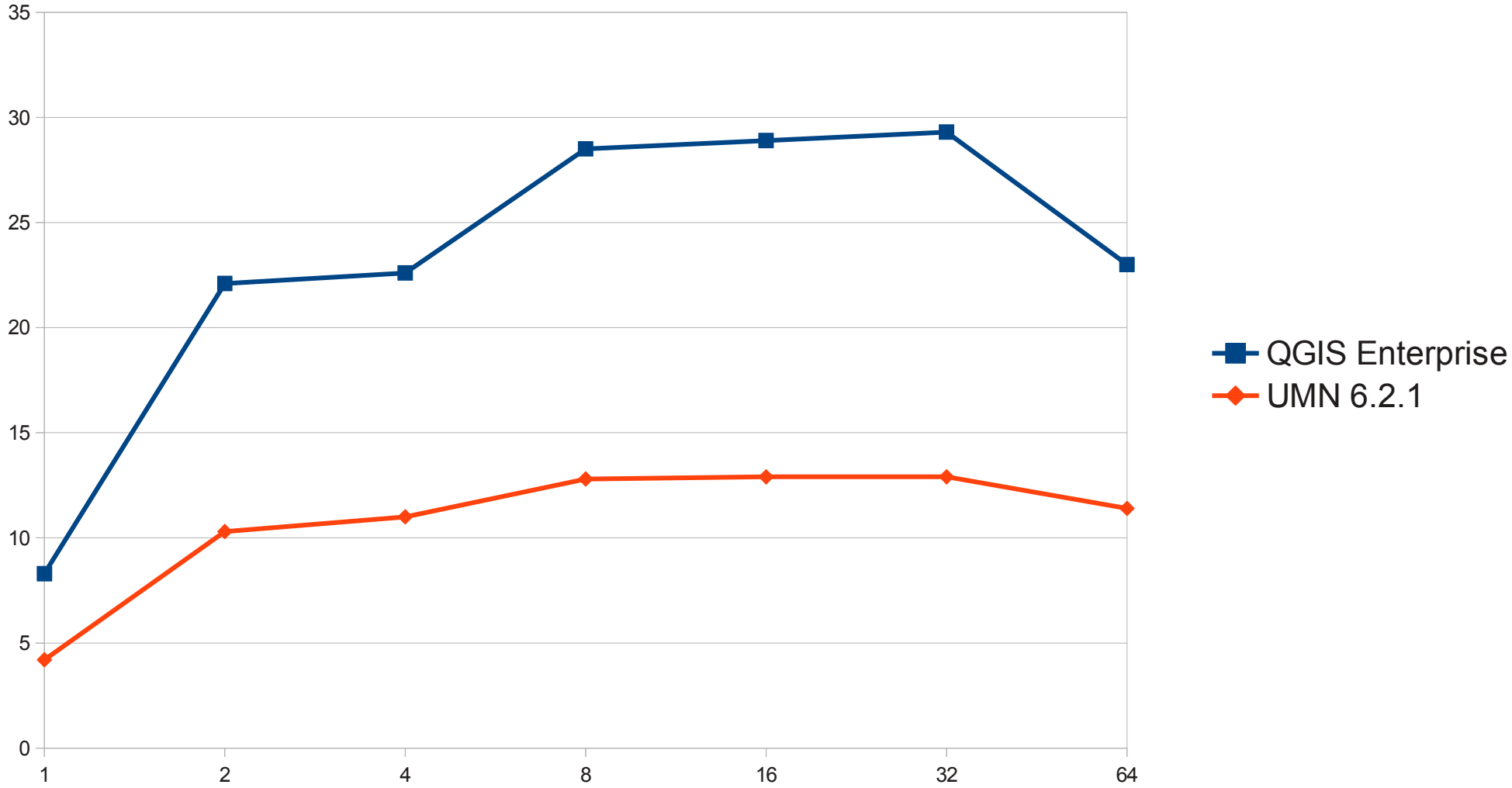
UMN mapserver

N req	count	avg	min	max	throughput
1	100	233	13	936	4.2
2	100	183	7	863	10.3
4	200	326	10	1671	11.0
8	200	559	10	3208	12.8
16	400	1112	10	5507	12.9
32	384	1758	13	22122	12.9
64	768	3642	12	58632	11.4


Inserthal
Bockmattli
Ueberse


Chöpfenberg
Wageten
Brüggler
Salbügel
Stockimwald
Oberurner Tal
Hüllispitz
Hinter Schmandi
Vorder Schmandi
Ospelberg
Brüschatockbüel
Innerthal
Tierberg
Bockmattli
Bärensolspitz
Schiberg
Plattenberg
Obersee
Brunnelistock
Rossalpelispitz
Zindlenspitz
Rautispitz
Wiggis
Lachenstock
Schijen
Urnersee

Benchmark LK25 Average resampling (Faktor 4.0)



Weitere Pläne

- Benchmark mit weiteren Karten (z.B. Orthofoto)
- Weitere Performanceverbesserungen QGIS Server (z.B. Labelcache)
- Vermeiden von Performance regressions durch Benchmark

Danke!

Marco Hugentobler
marco @ s o u r c e p o l e . c h