Umstellung von Memcached zu Redis

== Client Bibliotheken ==

=== Performance ===
Wer auf Performance achtet, der sollte immer auf eine C Library zurückgreifen. Einen Benchmark findet man ihr [wiki] Memcached VS Redis - Vergleich der PHP Clients[/wiki]. Bei PHP biete phpredis die besten Ergebnisse.

=== GZIP ===
Es bietet sich an die GZIP Komprimierung größerer Inhalte zu aktivieren. Die meisten Memcached Libraries beherrschen dies.
Unter Redis bieten dies leider die wenigsten Libraries.

=== Consistent Hashing ===
Redis wird zukünftig eine Art serverseitiges Consistent Hashing implementieren, indem sich die Server untreinander austauschen. Aktuell ist dies nicht möglich und alle Clients implementieren andere Verfahren.
Nicht alle Libraries (z.B. phpredis) unterstützen Consistent Hashing und müssen daher selbst implementiert werden.

== Protokolländerungen ==

=== Increment ===
Während Memcached bei einem Increment auf einen nicht vorhandenen Schlüssel nichts unternimmt legt Redis automatisch einen neuen Schlüssel mit Increment Wert an.
Bei Redis Versionen unter 2.1.5 gibt es einen Bug. Wird das Increment auf eine Variable mit ttl angewandt, dann wird die Variable umgehend genullt.

=== Persistenz ===
Redis Werte können persistent gespeichert werden. Das ist nett, bringt aber auch Nachteile. Wenn der Speicher voll ist, und keine Platz mehr vorhanden ist, dann gibt es Exceptions.

=== Cleanup ===
Memcached hat einen ausgeklügelten Algorithmus der je nach Chunkgröße Slabs bildet und eine damit eine alternierende Anzahl an Items durchsucht um den ältesten zu löschen. Redis ist wesentlich simpler. Wenn Platz geschafft werden soll, dann werden zufällig drei Werte mit TTL gesucht und der Eintrag der als nächstes ausläuft wird gelöscht.

== Konfiguiration ==
=== Arbeitsspeicher ===
Memcached kann man abzüglich des vom System gebrauchten Speichern den vollen verfügbaren Arbeitsspeicher geben. Bei einem 8 GB Hauptspeicher kann man Memcached also etwa 7 GB zur Verfügung stellen.
Bei Redis könnte dies nicht reichen. In eigenen Erfahrungen hat der BackgroundSave Prozess teilweise den selben Speicher beansprucht und war laut "free" bei fast dem doppelten Speicher.