

# Introductie Verbindingen Bundelen (Connection Pooling)

[wim.bertels@ucll.be](mailto:wim.bertels@ucll.be)

Naamsvermelding-NietCommercieel-GelijkDelen 4.0  
Unported Licentie

# Verbindingen bundelen

Doel

=

De 'overhead' die optreed bij database connecties en lees/schrijf operaties beperken

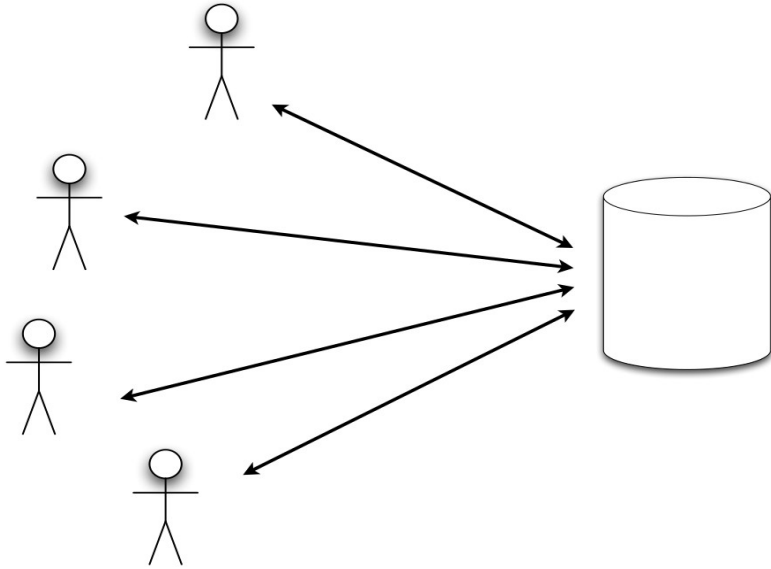
# Wat is connection pooling?

Een **connection pool** is een cache van database connecties die onderhouden worden door de database zodat ze hergebruikt kunnen worden voor request in de toekomst.

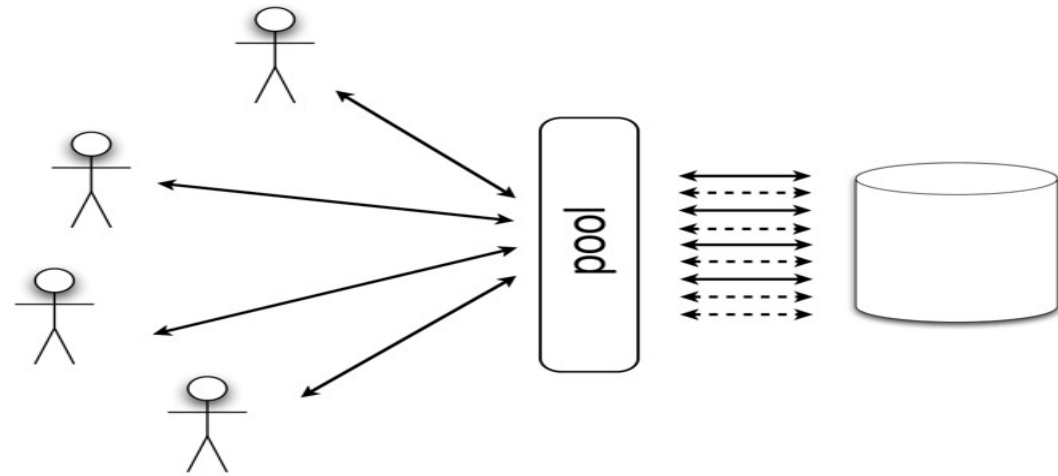
Opmerking:

Je kan verbindingen bundelen aan de client of server kant, of beide.

# Zonder Connection Pooling



# Met Connection Pooling



# Zonder pooling

Databank connecties zijn behoorlijk “dure” operaties. Elke connectie heeft een opstart kost en vraagt geheugen om te kunnen voldoen aan alle mogelijke eisen die van de connectie worden verwacht.

Bv

Er zijn 10 databank verbindingen

# Met Pooling

Verbindingen worden waar het kan gebundeld, gedeeld.

Bv

- Er zijn 10 verbindingen naar een pooling server
- De pooling server heeft maar 1 verbinding met de databank

# Verschillende vormen

Je kan op verschillende manieren gaan bundelen, vergelijk conceptueel een pooling server met een dynamisch wachtrij systeem bv

- Sessie:
  - Elke pooling connectie heeft zijn eigen databank connectie.
  - Deze databank connectie komt terug ter beschikking zo gauw de pooling connectie wordt afgesloten.
  - Alsof je een normale verbinding hebt, alleen is ze sneller ter beschikking
- Transactie:
  - Elke transactie verloopt door dezelfde databank connectie.
  - Deze databank connectie komt terug ter beschikking zo gauw de transactie beëindigd wordt.
  - Je kan geen veronderstellingen maken over zaken die buiten je transactie gebeuren!
- Instructie (statement):
  - Deze databank connectie komt terug ter beschikking na elke instructie.
  - Het is zinloos expliciet transacties te gebruiken!
  - Je kan hiermee wel het hoogst aantal verbindingen met de laagste kost bereiken voor applicaties